A Multi-Agent Reinforcement Learning (MARL) Decision-Support tool for Energy-Efficient Incremental Residential Development.

Sergio E. M. Poco-Aguilar

A thesis presented in fulfilment of the requirements for the degree of Doctor of Philosophy

Supervisors:

Prof Darren Robinson, Dr. Parag Wate

School of Architecture

The University of Sheffield

Sheffield, United Kingdom

Year 2023

Abstract

Building operations are responsible for about 30% of global greenhouse emissions. Thus, to attain emissions reduction objectives, action on buildings' operational energy use is vital. In this effort, the residential sector plays a key role. Nevertheless, the priorities in this sector are diverse around the world. Discussion in the global south tends to focus on informality and adequate housing provision rather than thermal comfort or energy savings. While it is true that emissions per capita in developed countries are considerably higher and the lack of quality housing characterises most urban areas in the developing world, these facts should not rule out the inclusion of environmental sustainability policies within low-income housing programmes in the global south. Many such programmes nowadays act on the incremental strategy, which is the most common form of building development in these regions. Incremental houses are unfinished but in conditions of habitability and are upgraded at a pace based on the financial capacities of their dwellers.

The current research responds to the envisaged need for tools and methods to promote passive building design in aided incremental residential development. This design strategy is used to minimise mechanical conditioning dependence while maintaining indoor thermal comfort. It currently relies on computational optimisation methods to achieve its goal. While high-profile buildings can afford these methods, they remain out of reach of help agencies and dwellers in incremental housing settings. As such, the present thesis defends the position that to unlock the capacity of millions of incremental dwellers to contribute to emissions reduction and propel a positive cycle of increased physical well-being with economic and environmental benefits, these methods should evolve to cater for the needs of these settings. As such, it presents an agent-based simulation workflow demonstrating the possibility of applying these advanced methods in incremental residential scenarios.

Acknowledgements

First of all, I will always be grateful to my supervisory team. Thanks for trusting in me and my research proposal, and thanks for always pushing us beyond the boundaries of what I thought I could achieve with my previous experience and education. Thank you as well to the White Rose Doctoral Training Partnership for giving me the opportunity to study in the UK; without their economic support, I could not have been able to carry on with this challenging journey. I am also very grateful to the University of Sheffield and its personnel for providing the tools needed for my research and for expediting all bureaucratic processes.

I would also like to thank the teachers and colleagues at the School of Architecture, I take invaluable lessons from this place of knowledge and friendship. Thanks to my family, who, despite not understanding my obstination to pursue a PhD so far away from home, has always been there for me. Last but not least, thanks to Stef for cheering me up and taking care of me, particularly during these challenging last few months; I told you it will soon be over.

Para la galleta en el horno

Table of Contents

List of Figures

List of Tables

Nomenclature

Most used acronyms

AVN Adquisicion de Vivienda Nueva, A sub-programme within TP that subsidises the acquisition of a new dwelling.

FMV Fondo MiVivienda S.A., a public company within private law that manages the funds for social housing in Peru.

NCMV Nuevo Credito MiVivienda, A social housing programme in Peru aimed at the middle class.

RNE Reglamento Nacional de edificaciones, Peru's main Building code.

TP Techo Propio Programme, A social housing programme in Peru aimed at lower income sectors.

Currencies

MU Monetary unit

PEN Peruvian sol

S/ Peruvian sol

USD United states dollar

Mathematical symbols

$\cup$ Union

$\forall$ For all

$\sum$ Sum

Non-Standard Units

GLB Global

m.a.s.l. Meters above the sea level

Pce One Piece

Pt One point

Other symbols

Inf Infinite

k Thousands

S South

W West

Reinforcement Learning Variables

α Learning Rate

ϵ Exploration factor

γ Discount factor

λ Eligibility traces

A Set of Actions

a Action

r Reward

S Set of states

s State

Non-SI Units

$''$ Inches

MMBTU One million British thermal units

Chapter 1

Introduction

Buildings' operations contribute significantly to Greenhouse Gas emissions, accounting for approximately 30% of global emissions. If Net Zero targets are to be achieved, intervention in improving existing and future building stock is urgent. Although this is widely acknowledged, the role of the residential sector in the global south is usually minimised in these efforts. In the developing world, the emissions per capita are certainly just a fraction of those generated in the global north, but keeping emissions low as housing quality is improved and consumption patterns change is an argument to consider. In this sense, profiting from aided incremental development schemes, not only to improve living conditions as it is currently done but also to provide quality and thermally comfortable living solutions with minimal environmental effects, is the main topic of the current research. As such, this chapter introduces the identified problem that justifies the development of the current research. It continues by posing research objectives and describing the methodology designed to tackle the research problem. This methodology uses an Agent-based approach to execute a workflow of simulation tools and methods that allow compliance with the set objectives. Finally, and based on the workflow components, this chapter presents a structure for the rest of the document.

1.1 Problem statement

The following section will present the problem statement that inspired and derived in the development of this thesis. As such, it is composed of two parts. The first one will introduce the concept and relevance of incremental residential development. This section concludes by providing a general model of residential incremental development in the global south. A second section introduces the concept of passive building design, remarking on its usefulness as an efficient strategy to reduce emissions from the built environment. This second section defends the idea that to meet the increasing demand for housing in the global south with comfortable and sustainable dwellings in the short term, a passive design approach to aided incrementalism is needed. This "passive incremental design" strategy requires developing tools and methods that demonstrate that both approaches can be combined while bringing increased benefits for dwellers and supporting agencies.

1.1.1 Incremental residential development in the global south

Unemployment, labour informality and low wages all limit access to adequate housing for millions in the global south. Meanwhile, states cannot supply enough low-cost quality public housing for this ever-increasing demand. In this context, people find alternative solutions to cope with their needs. Incremental housing, a system where a house is incomplete but in conditions of habitability (Van Noorloos et al., 2020) and is later upgraded "at a pace based on the financing capacities of the families" (Greene & Rojas, 2008), has been consistently employed as a strategy for those who face this need. So much has this been that studies assert that between 50% to 90% of residential development in the global south is incremental (Habitat for Humanity, 2014).

These figures reflect the high relevance of informal urban development in the global south. Still, it would be wrong to judge incrementalism as a strategy solely used by the urban poor in informal slums. As the size and quality of housing depend on the financing capacities of families, and in the current day and age, credit plays a big role in this capacity, the factor that pushes people to practice incrementalism

is not solely their income category but their capability to qualify for a loan large enough to pay for a finished residence. As achieving that status depends on proving a stable and permanent flow of money, most people who rely on the informal sector are pushed to incrementalism despite their income. Considering that in low- and middle-income countries, 67.1% of the total employment is informal (OECD, 2023), it is no wonder why so many people will rely on the incremental strategy for housing provision. Additionally, as property ownership gives a sense of reassurance during economic or political uncertainty (a permanent state in many global-south countries), many households prefer to expand a property they own rather than buy a bigger one, even if they qualify for a loan. Forced multi-generational co-habitation, resulting from high property costs in relation to incomes and the possibility of profit from a property's additional spaces, might also contribute to incremental development's popularity. As such, we could say that the incremental process happens in many contexts and due to diverse causes, but in general terms, we can identify certain factors that are always present.

First, we have an owner-occupier, an individual or collective household with ownership rights over a certain property they currently inhabit. This property is the second factor, a dwelling with certain characteristics that allow it to be expanded if needed. These characteristics include having enough space at the ground level to build an attached or detached annexe or a rooftop prepared to receive loads of one or multiple stories above. Thirdly, there is a need to be satisfied with the potential extension, whether real or perceived, current or future. This could be the intention to capitalise on the property to generate an income, or it could be a desire to enjoy living in a "decent and adequate" dwelling, one with enough spaces in size and number to develop all daily activities with comfort and adequate levels of individual and collective privacy. This factor, of course, is very subjective and has many cultural connotations, such as those implied in what privacy and comfort are and how they are measured. Finally, the factor differentiating the sort of incrementalism we will be referring to in this thesis from others is the owner's financial capacity. In the global south, most people practice incremental development because they cannot access the means that would allow them to get a fully "decent and adequate" dwelling at once. It would be mostly the case that not even the incremental extensions can be

paid in full. As a result, the "decent and adequate" dwelling comes, if ever, in pieces. Instead of paying a mortgage in parts, the parts and components of the house are paid in full, but as repayments, they come at different points in time. This latter factor usually implies that the need is ahead of infrastructure and is real rather than perceived.

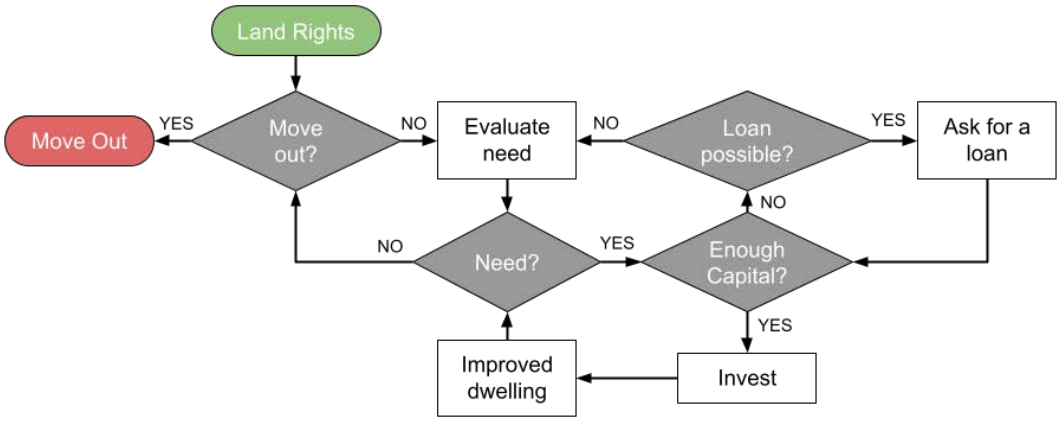


Figure 1.1.1: *Need-based loop in incremental residential development*

As shown in Figure 1.1.1, incremental development in this context can be seen as a cycle in which dwellers always chase their objective (a decent dwelling, one with real need) with the limited resources they can access. As time passes, new needs and opportunities emerge, and others disappear so that the "need" needs to be re-evaluated. Eventually, the dwellers might achieve their goal or move out of the property. In summary, two variables drive the transition between development stages: on one hand, the need, a push factor, and on the other, the financial resources, which bring an extension project to reality. A question of this schema left for future chapters is an objective definition of "need" to be used in this research. Meanwhile, financial capacity remains relative to material and labour costs and the technology used to extend a dwelling.

This schema also entails an opportunistic approach towards building development. As the decision to expand is taken once enough resources are gathered, rational builders will prefer to minimise their resource use to anticipate the expansion, expand the dwelling as soon as the resources are collected, and mind only immediate need satisfaction when locating the expansion. This implies that, on the one hand, there is not much space for planning and envisioning, and on the

other, the construction quality might be impacted. Because of this, when uncontrolled, incremental development might be a dangerous force that brings unsafe and unsustainable living spaces. Due to this, governments and international agencies have engineered programmes that support the incremental strategy (Wakely & Riley, 2011), so the risks associated with its weaknesses can be minimised, but the opportunities remain. This support may come in the form of financial, technical and legal help so that dwellers feel more confident to invest, thinking in long-term returns while minimising their initial cost and improving the overall quality and safety of housing.

While the discussion has so far focused on incremental housing at the individual household scale, this thesis is interested in the aggregate consequences of these practices. In this sense, incremental housing development constitutes an example of a complex adaptive system, one deeply rooted in the changing socio-economic conditions of the households that practice it. These conditions ultimately lead to investment, which in turn impacts the physical condition of the built environment and its associated variables. While some models exist to simulate incremental residential development at the individual (Alvarado, Donath, & Böhme, 2009) and collective scales (Augustijn-Beckers, Flacke, & Retsios, 2011; Peña-Guillen, 2019), these are focused on reproducing the phenomena to investigate the spatial outcomes of these interactions, paying little attention to other variables implied in the process, such as operational energy-use or emissions.

1.1.2 Passive incremental building design

One of the most pressing challenges of this century is to provide comfortable living spaces while curbing their use-related emissions. This challenge is particularly acute in the residential sector of the global south, where a significant portion of the population still lacks quality housing. The urgency is further compounded by the need to reduce emissions from a rapidly expanding built environment. The current climate emergency and the subsequent commitments to reduce energy-related emissions necessitate innovative solutions that can achieve both goals in a relatively short time. This section posits that supporting the incremental strategy could be one such solution. By enabling access to micro-credits to enhance the quality and

comfort of housing and reduce energy-related emissions, we could potentially unleash the potential of millions of self-builders in the global south to contribute to the fight against climate change. A positive side effect could be the conversion of energy savings into financial gains and comfort into productivity, thereby initiating a virtuous cycle of improved economic and physical well-being with environmental benefits.

In this context, passive building design emerges as a beacon of hope, offering a strategy that can significantly reduce the long-term energy costs for dwellers without compromising their indoor thermal comfort. This design approach, which uses layout, fabric and form to reduce or remove mechanical cooling, heating, ventilation and lighting demand (BRE, 2018), is not only fundamental but also highly effective in achieving building energy efficiency (Sadineni, Madala, & Boehm, 2011, Rodriguez-Ubinas et al., 2014). While the path to achieving a fully passive building design is not without its challenges, it's reassuring to know that modern Passive building design relies on Building Energy Simulation and Optimization (BESO) methods (Stevanović, 2013, Tian et al., 2018). These methods, which involve energy modelling and optimisation, can help ensure that design options translate into energy savings and thermal comfort, further bolstering the case for passive building design.

Even when BESO allows finding high-performing solutions with relative ease, a highly trained and competent workforce able to understand, interpret and transform data into designs is still required. Additionally, passive design usually implies additional financial costs on materials and technologies that ensure adequate levels of solar protection, insulation, and others. As a result, passive building design has an additional initial investment in talent, technology and materials compensated in the long term with lower energy bills. This increased initial investment can usually be afforded only by high-profile commercial buildings, keeping passive design out of reach of most people practising incremental development in the global south. The opportunistic nature of incremental development also doesn't match the careful planning needed to practice adequately passive building design.

As such, new approaches are needed if passive building design is ever combined with the incremental strategy. First, we will need tools that can show that both

approaches can be mixed and that, together, they can achieve increased energy savings and thermal comfort, both to individual households and as interventions at the neighbourhood scale. Initially, these tools would help convince dwellers and help agencies about the feasibility and profitability of applying passive building design strategies as programmes for incremental environments. Second, we will need to make the required technical support affordable so that BESO becomes a participatory tool rather than a high-technology gadget reserved for a specialised public. Third, we will need the commitment of the help agencies so that funds and promotion measures are put in place to ensure that the passive building design strategy is applied in the real world.

The current thesis presents the very first-of-its-kind approximation to these goals. This comes in the form of a workflow that informs interested parties on the applicability and profitability of implementing the passive building design strategy in residential incremental environments. Bearing in mind the various and complex variables implicated in this design strategy, such workflow is limited to building envelope geometries and their effect on individual and aggregate energy use and indoor thermal comfort, leaving for future developments the implementation of other strategies (such as insulation, openings, shading devices, ventilation, etc.) that would allow achieving fully passive incremental dwellings and neighbourhoods.

1.2 Objectives

This thesis aims to inform the interested parties in implementing an aided form of incremental residential development in the global south on the possible application and benefits of passive building design strategies at individual and aggregate levels. Bearing this in mind, this thesis has the following primary objective:

Primary objective: Demonstrate that passive building design, in general, and the optimisation of building envelope geometries, in particular, can be used successfully in the context of incremental residential development in the global south to achieve increased thermal comfort with energy efficiency at individual and aggregate levels. From this primary objective, three secondary objectives are derived:

- Build a computational workflow able to simulate the incremental development

of dwellings based on plausible assumptions and data.

- Develop a BESO methodology for optimised building envelope geometries able to deal with the particularities of the residential incremental strategy in the global south.

- Develop a workflow linking the outputs of the previously stated objectives to obtain plausible results that demonstrate the possibility of applying passive building design strategies in an incremental residential development context and showcase its comparative benefits.

These secondary objectives are presented in more detail as follows:

Secondary Objective I: Build a computational workflow able to simulate the incremental development of dwellings based on plausible assumptions and data. To achieve this, we need to:

- Devise a model of incremental dwellers' investment behaviour in the global south, based on plausible assumptions.

- Feed this model with plausible data on the socio-economic conditions that trigger incremental development.

- Implement this model in a platform able to feed the BESO methodology.

Secondary Objective II: Develop a BESO methodology for optimised building envelope geometries able to deal with the particularities of the residential incremental strategy in the global south. This tool should be able to:

- Evaluate the energy use and indoor thermal comfort of building envelope geometries.

- Select the best-performing options.

- Repeat the optimisation process every time the incremental process takes place.

Secondary Objective III: Develop a workflow linking the outputs of the previously stated objectives to obtain plausible results that demonstrate the possibility of applying passive building design strategies in an incremental residential development context and showcase its comparative benefits. This workflow should:

- Allow the interaction between the incremental development model and the BESO.

- Test plausible scenarios that derive interesting and informative data outputs

- Present data at individual and aggregate levels clearly and convincingly for the use of interested parties

1.3 Methodology

The proposed methodology uses an Agent-based modelling (ABM) approach. This approach, which has been consistently used to study complex systems in urban and architectural research (Chen, 2012), allows for getting data at the individual and aggregate levels and testing different interaction scenarios. As such, the methodology is expressed as an ABM workflow for social simulation able to facilitate a greater understanding of incremental housing development as an emergent socio-spatial phenomenon. This workflow follows the exploratory modelling paradigm (Bankes, 1993), and a post-rationalist approach to planning theory. The earlier implies that the model is used to define salient characteristics and "inform" debates rather than aiming at accurate or focused predictions, while the latter considers design as an argumentative process (Rittel & Webber, 1973) in which "negotiation and compromise between parties" (Bazjanac, 1974 cited by Batty, 2013) are essential for the decision-making process.

This methodological setup should allow for studying the aggregate impact of individual choices on operational energy use at the neighbourhood scale. Its components respond to the research objectives. As such, the methodological proposal (Figure 1.3.1) is divided into two parts (A and B), with an intermediary component between them. This intermediary is central to the posed methodology and, along with Part A, responds to Secondary Objective I: While Part A feeds with plausible data the sought model, the intermediary is in itself the model described on the said objective. Part B, meanwhile, responds to Secondary Objective II, while the overall arrangement (C) responds to Secondary Objective III. Its execution leads to the results expected by the Primary Objective of the thesis. Due to the ABM approach,

these results come initially at the individual level and are subsequently aggregated to understand the macro trends.

To achieve its objective, the intermediary component reproduces the generalised incremental residential development model presented in the first section of this chapter (Figure 1.1.1). Part A, meanwhile, achieves its objectives by generating data in terms of "need" and "financial capacity" for all the periods requested. As presented before, these two variables are the determinants of the incremental process and feed the intermediary component. With the interaction between these two components, data on the timing of dwellings' expansion is produced, and thus, the timing of buildings' geometries change is delivered to Part B.

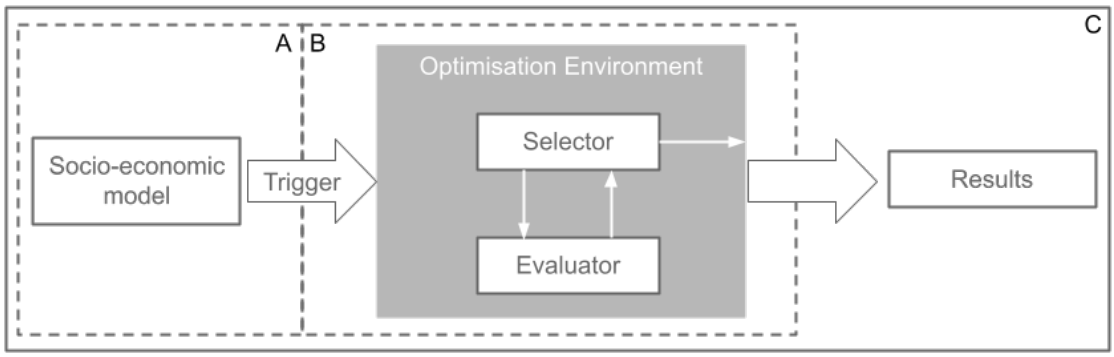


Figure 1.3.1: *Thesis Methodology*

Part B, meanwhile, follows the typical set-up of a contemporary building design optimisation loop: an Evaluator that provides information on the performance of available options and a Selector which searches for new candidates and selects the best-performing ones. Part B does not only use optimisation as a way to implement BESO in the workflow, but also to simulate agents' cognition in an ABM setting under a bounded rationality premise. As such, agents make satisfactory rather than optimal decisions, a problem tackled by optimisation methods if these are bounded by computation and available information (Brownlee, 2007). Our optimisation loop will, therefore, converge in "good enough" rather than optimal solutions. This optimisation process's objective function aligns with what is presented in Secondary Objective II: the energy use and indoor thermal comfort resulting from varying building envelope geometries.

To use BESO as an agent cognition tool, the workflow assumes that agents can access a piece of information usually hidden from them: the potential operational

energy use of their building expansion alternatives. Rationally bounded individuals will, therefore, aim to minimise the running costs of their residences when deciding where to incrementally expand their dwellings. The emergent phenomena depicted will, therefore, not reproduce what exists in the real world but will be the consequence of a universal policy choice, sharing the said information with the participating agents.

This implies the evaluation of a "socio-technical" system in which agents' interactions are directed by their bounded rationality complemented by a piece of technical information provided by a third party (an "aid" agency). This technical information is provided by the "Evaluator" component in the workflow. Solutions are satisficing as the problem remains bounded by computation (time and resources) and the time horizon of the available information (it only includes extension options possible with the currently available financial means). This additional piece of information also serves to keep track of the aggregate values and thus evaluate the consequences, in environmental terms, of an emergent phenomenon.

To obtain this information, the workflow uses Building Energy Modelling (BEM) as an "evaluator" component. Contemporary BEM software can accurately predict energy use and thermal comfort at individual building scales, which becomes convenient when setting up buildings as agents in an ABM setup. EnergyPlus (*EnergyPlus™, Version 00*, 2017), a whole building white box BEM software widely used and validated in academia and industry, became the selected tool among available BEM software. As a white-box model, EnergyPlus "uses detailed physics-based equations to model building components, sub-systems and systems to predict whole buildings and their sub-systems behaviours" (Li & Wen, 2014). Its popularity has also meant that it has become integrated into several wider platforms, such as the Urban Modelling Interface (Reinhart, Dogan, Jakubiec, Rakha, & Sang, 2013) and the Housing Energy Hub (Sousa, Jones, Mirzaei, & Robinson, 2018), which broadens the development alternatives for the rest of the workflow components.

Agents solve their satisficing problems using Multi-Agent Reinforcement Learning (MARL), which also serves as the "selector" component in the optimisation loop. This method is used as it allows the creation of a decentralised system that enables communication and learning. The experiments shown in the last chapter of this

thesis demonstrate how these characteristics lead to negotiation and compromise via tolerance values on the objective function. Other advantages of MARL are:

- It allows scalability, as with minimal changes, the same algorithm can jump from simulating the cognitive process of one agent to tens interacting of agents.

- MARL and, more specifically, Q-learning has been in circulation for many decades now. As a result, there are abundant learning resources to develop, test, and debug a model that uses them. Given the researcher's background, this was vital to complete the project.

To get plausible and persuasive results, the workflow needed real-world data. For this, a case study was needed. As a result, basic module developments in Peru were selected. This typology of aided self-help developments serves as an ideal case study as, unlike other such cases, they have trackable initial states at building and neighbourhood scales. This allows taking out multiple variables that would otherwise over-complicate the modelling task. As the chosen location, Peru responds to its long-standing tradition with this type of development, the availability of statistical data to create plausible models, and its variety of climates that allow the simulation of diverse thermal comfort scenarios.

To sum up, the proposed workflow will be able to explore the possible outcomes of a single policy action and multiple agent interaction scenarios in the development of incremental housing projects in Peru by using BESO as an agent cognition tool in an ABM setting. This is to inform and help social scientists, policymakers, and urban designers have a better understanding of this emergent socio-spatial phenomenon and thus act accordingly to maximise their social benefits and minimise their impacts (particularly those related to their operational energy use). A complementary function (yet to be tested) is to serve as a "discussion facilitator" between communities and planners. All this should ultimately serve to achieve the main objective of this thesis, which is to demonstrate that passive building design can be used in the context of incremental residential development.

1.4 Contribution to Knowledge

The contribution to knowledge of the current research can be summarised in three key aspects.

- **First**, it demonstrates that the integration between microsimulation and MARL for agent cognition in an ABM setting is possible and functional. Although optimisation methods in architecture using social-like behaviour exist (such as particle swarm), and MARL has been previously used for agent cognition in social simulation models (Sert, Bar-Yam, & Morales, 2020; Martinez-Gil, Lozano, & Fernández, 2014), no record has been found of the integration between a micro-simulation model and MARL mechanisms. This is probably because, up to now, no usefulness has been found in integrating socio-economic data projections and dynamic optimisation techniques applied to the built environment. As such, this thesis opens research avenues so far unexplored.

- **Second**, it simulates incremental residential development using learning agents: Incremental residential development is not an uncommon research topic in the simulation community; nevertheless, so far, we have read only about constraint programming (Alvarado et al., 2009), rule-based ABM (Augustijn-Beckers et al., 2011) and diffusion dynamics (Peña-Guillen, 2019) as simulation methods. The current thesis presents the first attempt to use learning agents for residential incremental development modelling and simulation. It also significantly increases the number of simulated incremental dwellings in comparison with previous experiences, while it employs 3D CAD software as a simulation platform and data visualisation tool.

- **Third**, it implements a multi-agent and dynamic optimisation method in a building's envelope geometry optimisation task: Although nowadays, building geometry optimisation is a common task in BESO, studies use parametric optimisation as standard. This approach allows the use of robust methods, such as Evolutionary algorithms, that can promptly produce highly relevant results with relatively modest computational resources. Nevertheless, the post-rationalist approach to planning theory and the ABM nature of the workflow requires a method that goes beyond simply optimising single states in a static

environment. We need a tool that generates agent interactions that open negotiation spaces in multi-state building development. Dynamic optimisation in general, and MARL in particular, did not offer so far any comparative benefit in building envelope geometry optimisation against the parametric approach. This thesis tests these methods in a very specific setting and with very specific requirements, which demonstrate that there is space for dynamic optimisation in the BESO field of study.

1.5 Thesis structure

This thesis is structured so that enough space is given to describe the tools and methods used in each workflow component. As such, Chapter 2 is reserved to present the chosen case study. Chapter 3 presents Component A, while Component B uses Chapters 4 and 5. The first of these two is used to describe its execution environment, and the second one discusses the selected optimisation method and tests its capabilities (partially presented in Poco-Aguilar, Wate, and Robinson (2022)). Chapter 6, meanwhile, presents the outcomes of the workflow execution and the implementation of the intermediary component, with the thesis's primary objective being attained here. Finally, Chapter 7 presents the thesis conclusions and provides an outlook of possible future developments for this research.

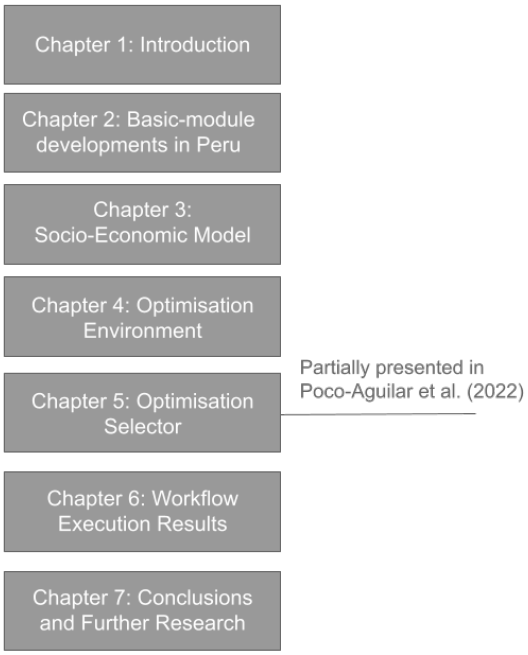


Figure 1.5.1: *Thesis Structure*

References

Alvarado, R. G., Donath, D., & Böhme, L. F. G. (2009). Growth patterns in incremental self-build housing in chile. *open house international*, *34*, 18.

Augustijn-Beckers, E. W., Flacke, J., & Retsios, B. (2011, March). Simulating informal settlement growth in dar es salaam, tanzania: An agent-based housing model. *Computers, Environment and Urban Systems*, *35*, 93-103. doi: 10 .1016/j.compenvurbsys.2011.01.001

Bankes, S. (1993). Exploratory modeling for policy analysis. *Operations research*, *41*(3), 435–449.

Batty, M. (2013, 11). A Theory for Collective Action. In *The New Science of Cities.* The MIT Press. Retrieved from https://doi.org/10.7551/mitpress/9399.003.0020 doi: 10.7551/mitpress/9399.003.0020

Bazjanac, V. (1974). Architectural design theory: Models of the design process. *Basic questions of design theory*, *3*, 20.

BRE. (2018, August). *The home quality mark one, technical manual sd239, england, scotland and wales.* BRE Group.

Brownlee, J. (2007). Satisficing, optimization, and adaptive systems. *Complex Intelligent Systems Laboratory, Centre for Information Technology Research, Technical Report A*, *70305*.

Chen, L. (2012, June). Agent-based modeling in urban and architectural research: A brief literature review. *Frontiers of Architectural Research*, *1*, 166-177. doi: 10.1016/j.foar.2012.03.003

Energyplus™, version 00. (2017, 9). Retrieved from https://www.osti.gov//servlets/purl/1395882

Greene, M., & Rojas, E. (2008, April). Incremental construction: a strategy to facilitate access to housing. *Environment and Urbanization*, *20*, 89-108. doi: 10.1177/0956247808089150

Habitat for Humanity. (2014). *Shelter report 2014. step by step: Supporting incremental building through housing microfinance.* Retrieved from https://reliefweb.int/report/world/shelter-report-2014-step-step-supporting-incremental-building-through-housing

Li, X., & Wen, J. (2014, September). Review of building energy modeling for control

and operation. *Renewable and Sustainable Energy Reviews*, *37*, 517-537. doi: 10.1016/j.rser.2014.05.056

Martinez-Gil, F., Lozano, M., & Fernández, F. (2014). Marl-ped: A multi-agent reinforcement learning based framework to simulate pedestrian groups. *Simulation Modelling Practice and Theory*, *47*, 259–275.

OECD. (2023). *Informality and globalisation. in search of a new social contract.* Author. doi: 10.1787/c945c24f-en

Peña-Guillen, V. (2019, September). Simulation of house consolidation process in lima using an epidemic diffusion mechanism. *Computers, Environment and Urban Systems*, *77*, 101347. doi: 10.1016/j.compenvurbsys.2019.101347

Poco-Aguilar, S. E. M., Wate, P. S., & Robinson, D. (2022, December). Multi-agent learning of incremental housing development strategies for solar utilisation in peru. In *Proceedings of bso conference 2022: 6th conference of ibpsa-england* (Vol. 6). Bath, UK: IBPSA-England. Retrieved from https://publications.ibpsa.org/conference/paper/?id=bso2022_28

Reinhart, C. F., Dogan, T., Jakubiec, J. A., Rakha, T., & Sang, A. (2013). Umi-an urban simulation environment for building energy use, daylighting and walkability.

Rittel, H. W., & Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy sciences*, *4*(2), 155–169.

Rodriguez-Ubinas, E., Montero, C., Porteros, M., Vega, S., Navarro, I., Castillo-Cagigal, M., ... Gutiérrez, A. (2014, November). Passive design strategies and performance of net energy plus houses. *Energy and Buildings*, *83*, 10-22. doi: 10.1016/j.enbuild.2014.03.074

Sadineni, S. B., Madala, S., & Boehm, R. F. (2011, October). Passive building energy savings: A review of building envelope components. *Renewable and Sustainable Energy Reviews*, *15*, 3617-3631. doi: 10.1016/j.rser.2011.07.014

Sert, E., Bar-Yam, Y., & Morales, A. J. (2020). Segregation dynamics with reinforcement learning and agent based modeling. *Scientific reports*, *10*(1), 11771.

Sousa, G., Jones, B. M., Mirzaei, P. A., & Robinson, D. (2018). An open-source simulation platform to support the formulation of housing stock decarbonisation strategies. *Energy and Buildings*, *172*, 459–477.

Stevanović, S. (2013, September). Optimization of passive solar design strategies: A review. *Renewable and Sustainable Energy Reviews*, *25*, 177-196. doi: 10.1016/j.rser.2013.04.028

Tian, Z., Zhang, X., Jin, X., Zhou, X., Si, B., & Shi, X. (2018, January). Towards adoption of building energy simulation and optimization for passive building design: A survey and a review. *Energy and Buildings*, *158*, 1306-1316. doi: 10.1016/j.enbuild.2017.11.022

Van Noorloos, F., Cirolia, L. R., Friendly, A., Jukur, S., Schramm, S., Steel, G., & Valenzuela, L. (2020, April). Incremental housing as a node for intersecting flows of city-making: rethinking the housing shortage in the global south. *Environment and Urbanization*, *32*, 37-54. doi: 10.1177/0956247819887679

Wakely, P., & Riley, E. (2011). *Cities without slums. the case for incremental housing.* Retrieved from www.citiesalliance.org

Chapter 2

Basic-module developments in Peru

2.1 Introduction

The current chapter aims to present basic module developments in Peru as suitable study cases to test the proposed workflow. To achieve this, the chapter starts by describing Peru's country profile, focusing on the country's relevant physio-geographic and socio-economic characteristics. A second section dives into the history of housing provision policies in the country to introduce the concept of basic module developments and characterise them in present-day Peru. A third section presents current policies in Peru that deal with thermal comfort and sustainability in the country's residential sector. A final section concludes by reckoning that basic module developments offer an ideal testing platform for the workflow due to their inherent top-down characteristics, which imply trackable initial states. Meanwhile, Peru's location is confirmed due to its rooted tradition of top-down support for incremental development and its diverse geography that allows testing the model with different climatic scenarios whilst keeping control over socioeconomic variables. This is complemented by the fact that the workflow might serve as a tool to support the country's approach to the comfortable and environmentally conscious design of low-cost housing fitted for its context.

2.2 Relevant facts and figures of Peru

2.2.1 Physio-geographical characteristics of the country

The Republic of Peru is in western South America, facing the southern Pacific Ocean. It is wholly positioned below the equatorial line and within the torrid zone, with its northernmost point located at 0° 2' 21.42" and its southernmost point at 18° 21' 0.42" of southern latitude. With a total land area of 1,285,216 km2 (about 5.3 times that of the UK), it is the third largest country of the subcontinent, only after Brazil and Argentina (Central Intelligence Agency, 2023a). The Peruvian climate is very diverse, derived from its location and geographical features. The country is divided into three regions due to the Andes mountains, which traverse the country from north to south, reaching altitudes up to 6,746 metres (22,133 ft) above sea level (Central Intelligence Agency, 2023b). On the eastern side of the

Andes lays the tropical rainforest; on the western side, protected from the easterlies by the peaks, drier and stable conditions emerge. These latter conditions are aggravated by the predominance of the northward-flowing Humboldt current, which brings cold waters from Antarctica, creating a semi-permanent temperature inversion phenomenon when it crosses tropical latitudes. As a consequence, ocean humidity becomes trapped in the lower layers of the atmosphere, and vertical cloud formations cannot emerge. Thus, heavy rains in the central and southern coastal regions are sporadic, but as the influence of the Humboldt Current weakens further north, a different climatic pattern emerges, developing a savannah-like ecosystem.

Figure 2.2.1: *Eco-regions in Peru.*
Source: ProjectPeru (2023)

Despite the desert-like climate in the western part of the country, several rivers flow from the top of the Andes to the Pacific Ocean, feeding the soil and creating highly productive valleys on their way. Eastwards, wider meandering rivers fed by higher amounts of rain run freely on the plains and eventually join to form the Amazon River, thus creating an intricate communication network that connects towns and cities. The Andes are, therefore sculpted by water on all their sides, forming deep valleys in which urban areas and productive activities have emerged.

In some sectors, broad plateaus emerge, where livestock, mining and commercial activities sustain towns and cities that are among the highest inhabited places in the world (NASA Earth Observatory, 2023).

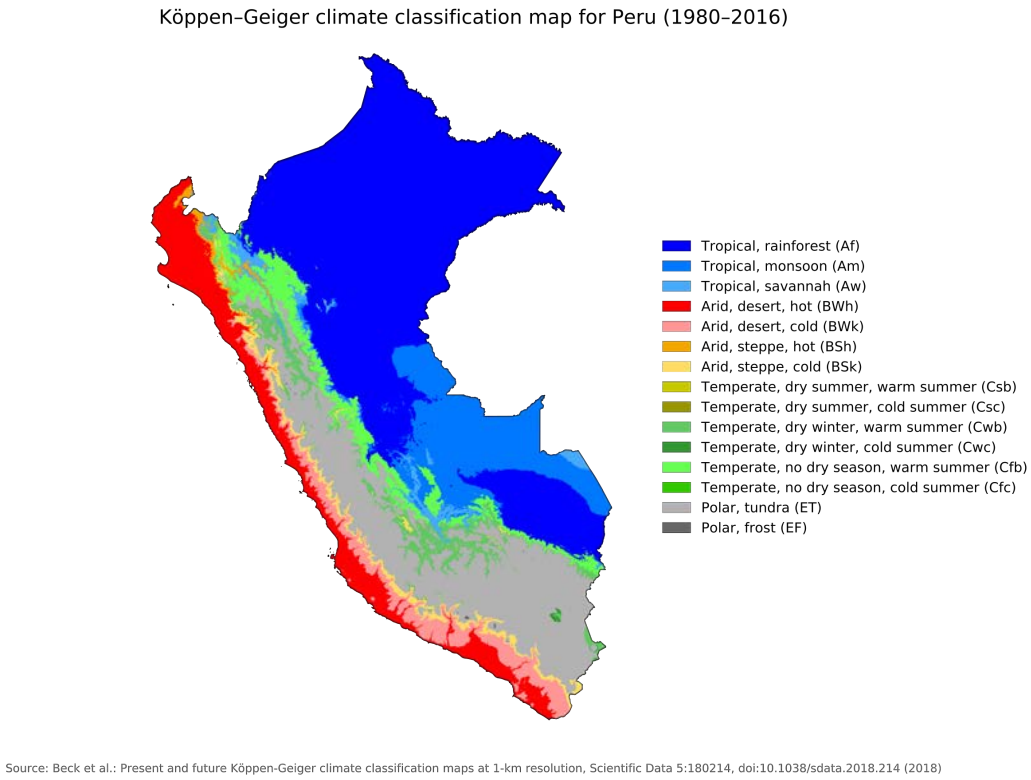


Figure 2.2.2: *Koppen-Geiger map of Peru.*
Source: Beck et al. (2018)

Climatic and geological conditions determine the most common natural disasters possible in the Peruvian territory. On the one hand, the same forces that formed the Andes mountains are still present. The Nazca plate, which lies below the Pacific, sub-ducts below the South American continental plate in front of the Peruvian coastline, thus making possible the occurrence of volcanic eruptions and destructive earthquakes, more frequent and dangerous the closer one gets to the sub-duction zone. On the other hand, the relative climatic stability that southern cold waters bring to the west of the country can be suddenly interrupted when the global El Niño Southern Oscillation (ENSO) or the local El Niño Costero phenomena warm the waters, thus bringing heavy rains where unaccustomed, and drought to rain-dependant areas. Populations in deep valleys are particularly vulnerable to landslides, as dry slopes might receive atypical rainfall during these events.

2.2.2 Socio-economic characteristics of the country

Peru's population is projected to reach 33.7 million inhabitants for 2023 (INEI, 2023). This figure is unevenly distributed in the territory, with almost a third of the country's inhabitants living in the metropolitan area of Lima-Callao (INEI, 2023). This fact also tips the balance in favour of the coastal region being the most densely populated in the country, with about 59% of the country's population living in what represents 12% of the land area (INEI, 2023). The highlands, considering all people located above 500 m.a.s.l. on the western side and 1500 m.a.s.l. on the eastern side (INEI, n.d.), shelter about 26.8% of the population (INEI, 2023). The eastern lowlands, which account for about 60% of the territory (Ministerio de Agricultura y Riego Peru, n.d.), only hold 14.2% of the population (INEI, 2023).

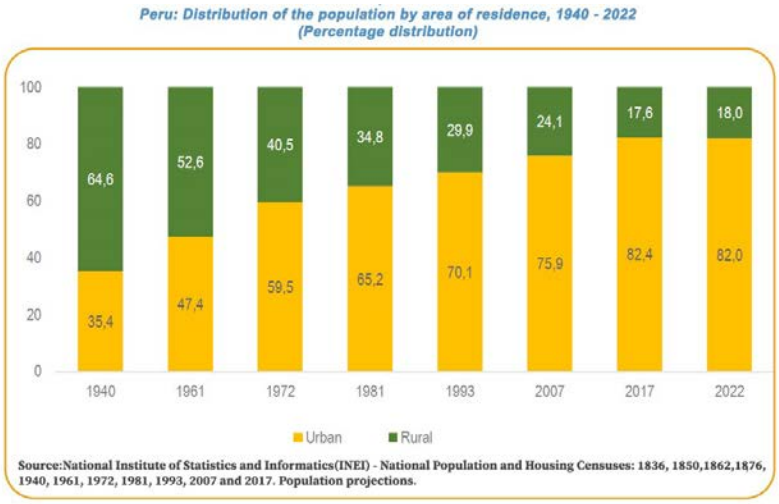


Figure 2.2.3: *Evolution of Population distribution in urban and rural areas in Peru*
Source: INEI (2022)

Peru is also highly urbanised, with official figures reckoning that 82.6% of the population lives in urban settlements (INEI, 2023). The country experienced rapid urbanisation between the 1940s and the 1990s, which increased the number of inhabitants living in urban areas from 35.4% in 1940 to 70.1% in 1993 (Figure2.2.3). This process was mainly driven by incipient mid-century industrialisation and disastrous policies and natural events affecting the countryside in the 50s and 60s. This deepened the inequality between urban and rural areas and produced a massive migratory movement to major cities. Centralisation and consequential enhanced opportunities made Lima the leading destination of this movement. Successive governments could not handle such enormous population growth, and therefore, vast

regions of Peru's most significant cities grew out of self-help housing initiatives.

Peru: Cities with over 100,000 inhabitants, 1940-2022
(Thousands of inhabitants)

Cities	Censused population							Projected population 2022
	1940	1961	1972	1981	1993	2007	2017	
Lima	562,9	1 632,4	2 981,3	4 164,6	5 706,1	7 605,7	8 575,0	9 943,8
Callao		213,5	321,2	443,4	639,7	876,9	994,5	1 154,2
Arequipa		163,7	306,1	442,9	629,1	806,5	1 009,0	1 142,9
Trujillo		102,3	240,3	364,4	537,5	766,1	919,9	1 111,3
Chiclayo			177,3	263,2	393,4	527,2	552,5	609,4
Chimbote			174,2	231,6	291,4	320,2	354,3	402,6
Iquitos			110,2	169,1	274,8	367,2	377,6	451,2
Piura			126,0	202,1	272,2	377,9	473,0	562,3
Huancayo			126,8	171,8	279,8	385,4	465,8	547,2
Cusco			121,5	177,6	255,6	348,9	428,5	523,3
Ica				112,5	161,5	238,1	289,9	348,1
Sullana				117,0	149,1	170,3	184,9	206,2
Tacna					174,3	242,5	286,2	337,5
Pucallpa					172,3	272,6	327,6	410,8
Juliaca					142,6	216,7	276,1	328,7
Huánuco					118,8	146,8	196,6	233,9
Chincha					112,2	153,6	181,5	215,0
Ayacucho					105,9	151,0	216,4	268,3
Huacho					104,3	133,6	156,8	185,2
Cajamarca						161,2	201,3	243,8
Puno						119,1	128,6	143,1
Tarapoto						116,6	144,8	171,7
Huaraz							118,8	140,0
Puerto Maldonado								116,5
Tumbes								108,2
Talara								102,4

Source:National Institute of Statistics and Informatics(INEI)-National Population and Housing Censuses:1940, 1961,1972,1981,1981, 1993, 2007 and 2017. Population projections.

Figure 2.2.4: *List of Cities with more than 100k inhabitants in Peru since 1940*
Source: INEI (2022)

As a middle-income country, Peru faces several obstacles that limit its growth and development. A chronic condition of the country is its economy's significant levels of informality. Although figures show that unemployment has remained relatively low for the last couple of decades (Figure 2.2.6), most people receive their income from activities carried out in the informal sector (Figure 2.2.5). Therefore, most Peruvian workers do not enjoy social welfare benefits, such as time off, health insurance and retirement pensions. Despite this, poverty figures (Figure 2.2.6), although high, do not correspond with the levels of labour informality. This indicates that working in

the informal sector is not necessarily a synonym for poverty.

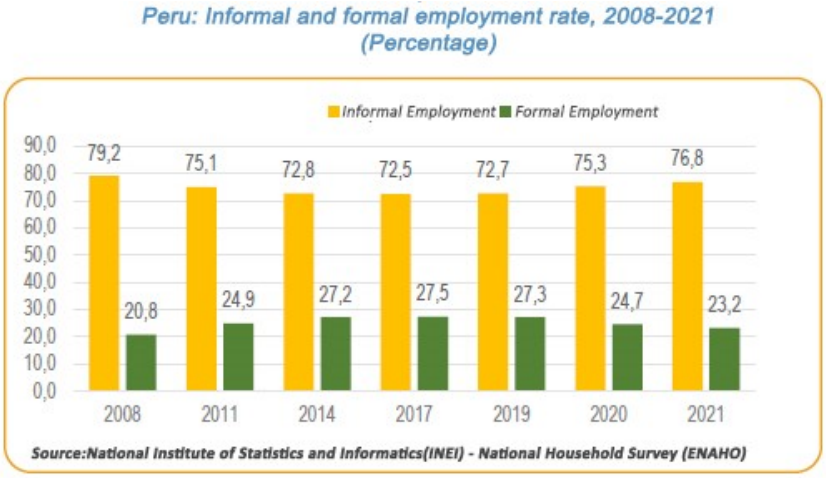


Figure 2.2.5: *Formal and informal employment figures in Peru*
Source: INEI (2022)

An essential driver of the prevalence of the informal sector in the economy is the lack of complete control of the state over its territory and resources. As a result, private initiatives may decide to remain on the fringes of the law for several years with only minor consequences. Additionally, there is a competitive advantage derived from lower operation costs in informality. This causes a vicious cycle in which entrepreneurial dwarfism prevails. Successful "lawful" businesses either profit from informality and remain small to hinder their detection or become formal, invest in growth and face unfair competition from their informal counterparts. This, of course, impacts productivity and innovation as, to achieve profits, investment in technology and workforce training seems secondary.

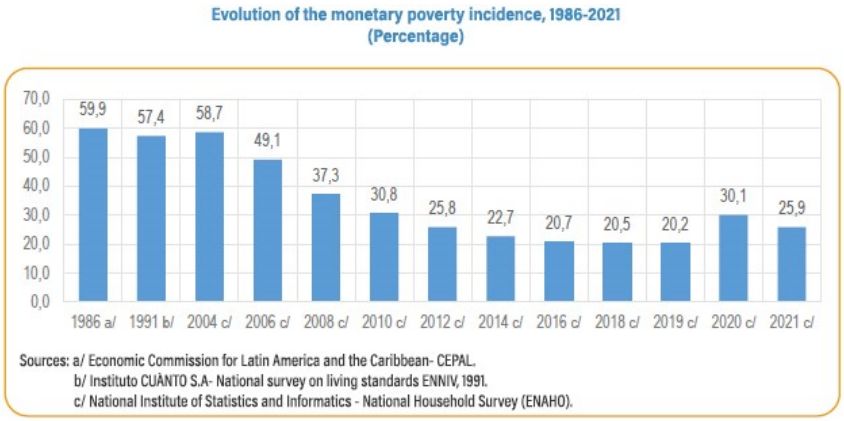


Figure 2.2.6: *Evolution of monetary poverty in Peru*
Source: INEI (2022)

2.3 Basic Module Developments in Peru

2.3.1 A brief history of basic-module developments

Despite occurring in diverse contexts, the incremental strategy in the global south has traditionally been associated with informal residential development due to its incremental nature. Until the 1960s, there was the widespread idea that slums only brought evil to cities and that they and the associated ideas had to be cleared to give way to "modernity". It was not until John F.C. Turner's writings on his experience in Latin America (particularly in Peru) became popular among practitioners in the early '70s that the ideas on how to deal with (and learn from) slums started to shift (Fernández Maldonado, 2015; Bromley, 2003). The impact was so relevant that, in a matter of just a few years, organisations such as UN-Habitat were promoting internationally some programmes suggested in his writings as feasible solutions to the lack of proper housing in the global south (Kozak, 2016). As a result, several "aided self-help" programmes were implemented in various countries worldwide in the following years.

Turner did not invent "aided self-help", but he did become the leading promoter and defender of a range of initiatives that had been taking place in various Latin American countries since the 1960s (Harris, 2003). As mass migration from the countryside to the urban areas occurred, governments faced increasing pressure to take action on illegal land occupations and slum development. Faced with political and economic factors that limited intervention choices, many decided to negotiate with illegal occupants and slum dwellers. As a result, a degree of coordination and, eventually, cooperation between government officials and slum dwellers developed in some places. This was the case of various cities in Peru, extensively described in Turner's writings (Turner & Fichter, 1972). One of the popular initiatives when Turner visited Peru was what later became known as the "lots and services" schemes. In essence, government bodies, unable to provide fully finished dwellings, gave home seekers property rights, communally or individually, with a low or no charge, of public land destined for urban development. Government intervention continued by providing access to essential services (water, electricity and sewage) and a planned layout to occupy the land, including spaces reserved for public recreation and ad-

ditional infrastructure. The development of the dwelling infrastructure itself was mainly reserved for the now owner-occupier, who could receive technical or financial support from the government or the community but was solely responsible for taking the initiative to start and continue the construction. As with the dwellings, the public infrastructure was developed incrementally and sometimes collaboratively. In these initial schemes, households could spend decades living in sub-standard temporary accommodation solutions until, eventually, they could raise the money to start developing a more durable and proper dwelling. Consequently, an alternative developed with time, in which an expandable dwelling core was provided in addition to property rights (Gyger, 2019a).

This other type of aided self-help, based on initial "basic extendable modules", also became globally relevant thanks to a design competition in Peru. Developed between 1968 and 1975 (García-Huidobro, Torres Torriti, & Tugas, 2011), PREVI (an acronym in Spanish for "Experimental housing project") drew the attention of several internationally renowned architects of the time, who competed to present a residential project where a house would "not be considered as a fixed and unchangeable unit, but as a structure with a cycle of evolution." (Gyger, 2019a). Although small in scale (only 467 units were built on the first and only stage of the initiative), the impact of PREVI locally and globally was considerable. In Peru, "basic modules" became the epitome of the social housing typology for several decades. [1].

2.3.2 Contemporary social housing policies in Peru, the ABC model

The era of innovations in housing policies in Peru gradually faded away in the 80s as the country entered a deep and challenging economic and political crisis. The subsequent neoliberal reforms of the '90s obliterated the few remnants of social hous-

[1] "Social housing" is used here in the sense of "housing provided or subsidised by the state and aimed at low-income sectors", thus discriminating alternative state provision strategies such as "sites and services" and programmes aimed at middle-income sectors. "Epitome" is used as a synonym of "archetype", "paradigm", "prototype", or "classic example/case". The claim thus implies that the prototypical typology of state-provided buildings to house people in low-income sectors in Peru was, for many decades, the basic module. This assertion is based on diverse literature, including Gyger (2019b); Fernández-Maldonado and Bredenoord (2010) and Fernandez-Maldonado (2014) who report the prevalence of this housing typology in the policies of various administrations in Peru.

ing provision policies, limiting the government's action to formalise informal land occupations (via the Commission for the Formalization of the Informal Property - COFOPRI) and offering low-interest loans for construction materials and technical help for self-construction (via the Materials Bank – BANMAT). Only in 2001, with the reinstatement of the Ministry of Housing, subsequent governments became committed to supporting housing provision in the country (Fernández-Maldonado & Bredenoord, 2010). Since then, the Peruvian housing policy has been based on the ABC (an acronym in Spanish for Savings-Bonus-Credit) model. Based on housing provision by ownership, this model asks that potential owners have an initial minimal amount of *Savings* to buy a property, complemented by a cash *Bonus* given free of charge by the government. *Savings* plus *Bonuses* make the down payment for a *Credit*, a mortgage to be obtained in the private financial system. The property to buy can be used or new and, in any case, provided by the free market. This model is currently managed by "Fondo MiVivienda" (FMV), a state-owned investment company under private law that acts as guarantor (so the loans come at a lower interest rate) and directs the bonus funds (Peruvian Law Nº 28579, 2005).

Although, initially, the ABC model seemed successful, it has not been able to supply enough dwellings for the lower-income sectors (Leon, 2022). This is, on the one hand, because as a system relying on financial credit, it cannot reach those who depend on the informal economy, while, on the other, the dwelling costs determined by the market are well above the average income of the poorer households (Calderon Cockburn, 2014). What this policy did achieve was to offer better conditions for the middle classes to access a mortgage, and due to the contained demand from decades of neglect and the contemporary economic growth, the construction industry saw a golden era. Lima and some other large cities in the country saw rapid vertical development in their most notorious middle-class areas (Fernández-Maldonado & Bredenoord, 2010). At the same time, middle-size cities, particularly coastal regions, saw a rapid urban expansion thanks to single-family middle-income dwelling developments.

Because of the weaknesses of its approach, FMV was revamped in 2006. As part of this, the "TechoPropio" programme was established as an ABC alternative for the lower-income sectors; meanwhile, the "MiVivienda" programme, which had al-

Table 2.1: Bonus brackets according to property retail value under the MiVivienda programme

Property value	Minimum Deposit	Non-refundable top-up	Maximum loan to property value rate
65,200 – 93,100 PEN (16,300 – 23,275 USD)	7.5%	25,700 PEN (6,425 USD)	65%
93,101 – 139,400 PEN (23,275 – 34,850 USD)	7.5%	21,400 PEN (5,350 USD)	77%
139,401 – 232,200 PEN (34,850 – 58,050 USD)	7.5%	19,600 PEN (4,900 USD)	84%
232,201 – 343,900 PEN (58,050 – 85,975 USD)	7.5%	7,300 PEN (1,825 USD)	89%

Source: Ministerio de Vivienda Construccion y Saneamiento Peru (2022)

Table 2.2: Bonus brackets according to property retail value under the Techo Propio AVN programme

	Top Monthly Household Income	Cap on property value	Minimum deposit	Non-refundable Top-up	Maximum loan
Prioritised Single-family Dwelling	2,071 PEN (518 USD)	55,000 PEN (13,750 USD)	0.3%	43,240 PEN (10,810 USD)	11,595 PEN (2,899 USD)
Non Prioritised Single-family Dwelling	3,715 PEN (929 USD)	96,000 PEN (24,000 USD)	0.3%	40,250 PEN (10,062 USD)	55,462 PEN (13,865 USD)

Source: Ministerio de Vivienda Construccion y Saneamiento Peru (2021b)

ready gained a reputation among consumers and developers, became "Nuevo Credito MiVivienda" (NCMV) and continued to aim at middle-income sectors (Fernández-Maldonado & Bredenoord, 2010). To achieve its goal, the TechoPropio programme caps beneficiaries' incomes and the retail price of housing units (Table 2.2). Additionally, it also presents products aimed at the self-builder, such as "Construction on own site" (CSP) and "Housing improvement" (MV), which offer Bonuses, low-interest credits and technical help for those who have a property to improve, extend or develop. The TechoPropio product comparable to that of NCMV is "new dwelling acquisition" (AVN), which finances the purchase of a property below a capped value.

Even after these changes, and with TechoPropio having existed as an ABC programme for almost 20 years, the model's outcomes still seem insufficient. Recent reports (Figure 2.3.1) show that, out of the total offer of newly built housing in the Peruvian urban market in recent years, only 4% come from schemes financed by TechoPropio's AVN. Meanwhile, NCMV performs slightly better with 7%. Ahead

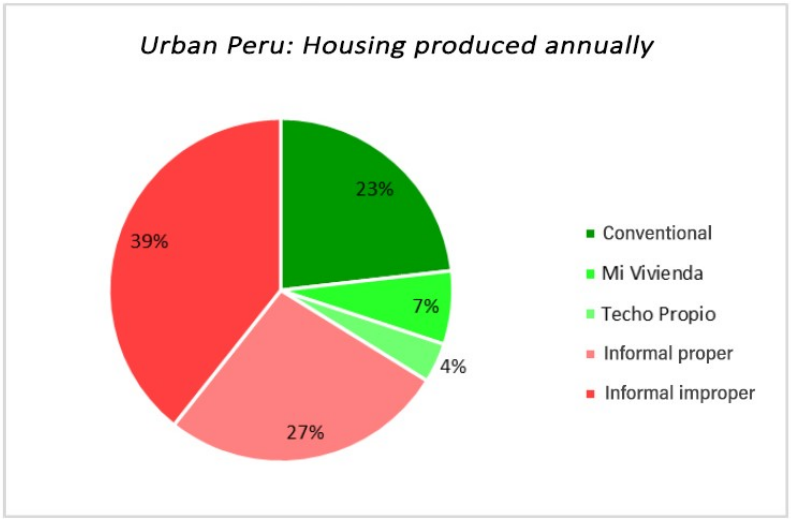


Figure 2.3.1: *Distribution of modes of housing production in Peru per year*
Source: Grupo de Analisis para el Desarrollo (2020)

are traditional mortgages (without governmental help) and the elephant in the room, informal development, which, with its two typologies combined (adequate and inadequate), represent more than 2/3 of the yearly production of new housing in the country. As expected, self-help provision overruns any governmental effort to supply adequate housing. Many structural problems contribute to the incapacity of the formal market to cope with the demand. Particularly relevant is the lack of well-positioned and well-sized lands for development in larger cities, which would allow diminishing costs, and the informality of the job market, which limits the eligibility of low-income individuals to qualify for a mortgage (and thus for government help under the ABC model) (Calderon Cockburn, 2014).

2.3.3 Characterisation of current basic-module developments in Peru

Now that the concept and origins of basic module developments in Peru have been established, we can proceed to characterise them in current-day Peru. This section does so by first identifying who the most likely consumers of this type of housing development are, then where these developments are mostly located, and finally, what are their material and spatial characteristics, both at individual dwelling and neighbourhood scales.

To characterise basic-module developments in Peru, we must first know who

currently promotes them and whom they are aimed at. With this information, it is possible to locate and size them, get the average market values of the dwellings, and their spatial and technical characteristics.

First, we can fairly say that the bulk of basic module developments are promoted by governmental programmes aimed at the lower-income sectors. Although examples exist in the Peruvian context of private developers selling incremental modules without governmental involvement, these exceptions could be assumed to follow a model led by those working in cooperation with the public realm.

Second, we can reasonably assume that they are directed to lower-income sectors and thus promoted via TechoPropio's AVN. Although no official figures classify the dwelling financed by AVN as "incremental" or "not incremental", the caps on property costs established by the programme do give a hint to confirm this guess. These caps and the government's bonuses to households (Table 2.2) are differentiated according to whether the household belongs to a prioritised group and what type of property they aim at (single-family dwelling or multifamily residential). Knowing this, it is fair to say that, due to caps on the retail value of units, most single-family dwellings financed with the AVN bonus will be in basic-module developments. This information also gives an idea of the cost of the "seed" housing units.

By using the project search tool available on the FMV webpage (Fondo MIVIVIENDA S.A., n.d.-b), it is possible to confirm this guess. Nevertheless, it is also possible to note that some developments promoted via the NCMV programme are also incremental (Fondo MIVIVIENDA S.A., n.d.-a). The latest available figures (Fondo MiVivienda S.A., 2017) show that, at the national level, 36% of consumers of NCMV used the bonus to buy a house. This figure increases in regions other than Lima and ranges between 50 – 72% of the properties purchased with FMV's products. Of course, it would be wrong to say that all of these are located on basic module developments first because these include both new and used properties and second because even in the 'new' category, there might be other types of development. Despite this, we must acknowledge that basic-module developments aimed at the middle-income market exist.

Nevertheless, this phenomenon seems to be limited in geographical scope. By checking the project search tool (Fondo MIVIVIENDA S.A., n.d.-a), it is noticeable

that most NCMV basic module developments are located in agricultural-export-dependent coastal regions like Piura, La Libertad, Lambayeque, and Ica. It is no wonder that, outside Lima, as land values become lower, and particularly in coastal regions, where there is plenty of relatively flat accessible land, developers can offer single-family dwellings for middle-income households within caps imposed by NCMV (Table 2.1). Of course, keeping costs within limits might require further sacrifices, which means selling basic modules instead of fully finished homes. Market preference for a house rather than a flat and the disinclination of FMV to fund land acquisition (probably to avoid land speculation) might give space for this market to flourish in these locations. Of course, to keep up with the market targeted, these basic modules likely come with better quality finishings than those promoted by AVN.

To sum up, in today's Peru, it is possible to find basic-module developments promoted by government programmes or without their support. Finding those aimed at the middle-income market and those in lower-income sectors is also possible. Nevertheless, despite this variety, most projects seem to be financed by TechoPropio's AVN. Additionally, due to the burden of land value in the retail cost of units and the strict caps imposed by AVN, most of these projects seem to prefer regions outside metropolitan Lima. As a result, it is highly likely that by profiling AVN single-family dwellings projects outside Lima, we get an accurate picture of the current situation of basic module developments in Peru.

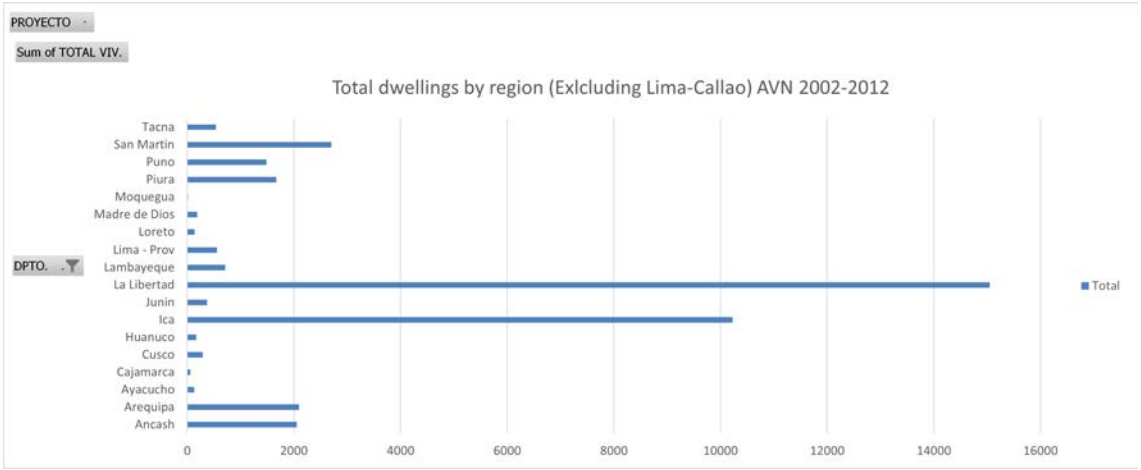


Figure 2.3.2: *Total dwellings per region (outside Lima-Callao) applying for AVN 2002-2012*
Source: MiVivienda SA (by request)

Now that we have identified our research target, we can proceed to characterise

incremental projects that exist within AVN. Information on these projects is only available (by request) to those registered with the programme between 2002 and 2012. Although ten years have passed since the latter data, that information is enough to get a perspective on the location and dimension of these developments in recent years.

Figure 2.3.2 shows that most AVN housing units registered outside Lima were in coastal regions, particularly in La Libertad and Ica. Data filtered for those regions reveal that the most significant projects were "Alto Salaverry II" in La Libertad, with 4346 registered dwellings and "Las Casuarinas de Ica" in Ica, with 2641 registered dwellings. An internet search confirms these were basic module developments and reveals their fate. While "Las Casuarinas" was developed in stages and has become a thriving community with a buoyant property market, "Alto Salaverry II" shows fewer results. We can only confirm that the beginning of its development was planned for 2014 (Gobierno Regional de La Libertad, 2014).

Despite the spectacular number of dwellings offered by these developments, these are at the highest end and do not represent what is happening in the whole country. The average number of dwellings by project by region (Figure 2.3.3) shows that, depending on the location, projects had between 10 and 540 dwellings, with a national average of 251.

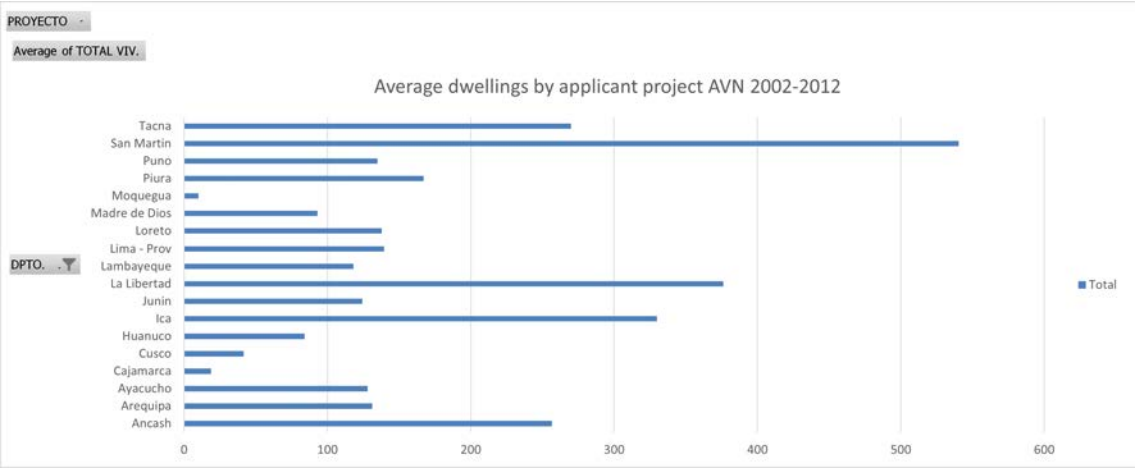


Figure 2.3.3: *Average dwellings per project by region (outside Lima-Callao) for AVN 2002-2012*
Source: MiVivienda SA (by request)

Now, regarding the materials and technologies used for construction, these depend on the location and the available resources. The country's location in an

earthquake-prone area, the availability of materials and the typical training available for the local workforce result in two construction systems that predominate for building massive housing developments (including basic modules). These are confined masonry and limited ductility walls. The earlier is a construction system where the masonry walls are assembled first, then concrete columns and beams are poured in to confine them. On the latter, the structural system comprises concrete slabs and thin walls (10-12cm) without any confinement at their extremes. Limited ductility walls are reinforced with electro-welded wire meshes to achieve seismic resistance. They are planned using a strict modularisation that does not allow long distances between vertical supports (typically, less than three meters in a single direction). While limited ductility walls are of relatively recent implementation, they are becoming preferred to produce low-cost housing as they reduce construction times and the area lost to structural elements.

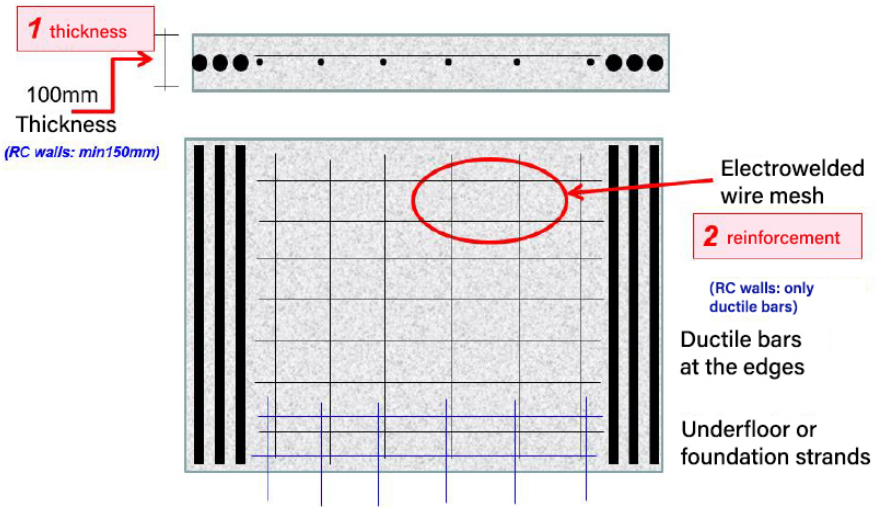


Figure 2.3.4: *Limited ductility wall detail*
Source: Diaz Figueroa (2021)

The construction method, the budget, and the legal framework determine the spatiality of the seed dwellings. A special code rules social housing developments in Peru (Supreme Decree Nº 002-2020-VIVIENDA), which allows smaller lot sizes (no less than 70m^2), reduced road sections (no less than 5.4m as roadway and 1.8m per sideway), and higher densities and allowed heights. Basic-module developments, therefore, implement minimal or close to minimal street widths unless the local urban development plan requires otherwise (when higher hierarchy roads are crossing the location). By law, areas must be reserved for parks, recreation and complementary

infrastructure (health, education, safety, etc.). Although developers must comply with the provision of the land for public spaces, the special code guiding social housing developments allows "progressive" implementation of the urban features, so developers might sometimes elect to leave this for municipalities to finish the job. It might then be the case that these areas remain vacant lots until the local government decides to invest in them (Figure 2.4(b)).

(a) Las casuarinas de Ica, Ica, Peru (b) Villa Marina in Salaverry, La Libertad, Peru

Figure 2.3.5: Satellite views of the most recent available state of (a) and (b)
Source: Google Earth

Even when there are differentiated rules for urban development for social housing projects in Peru, the construction standards remain the same for all buildings. Nevertheless, due to the cap on the retail price of the units, the usual finishings for seed modules are minimal. They typically include plywood doors, single-glazed windows, plastering (when masonry), internal and external paint coating and kitchen and bathroom tiling. The bathroom must also be fully equipped, and the kitchen needs to have at least an installed sink. Spaces are usually minimal and include an integrated kitchen-dining-living and at least one bedroom (Figure 2.3.6). Open spaces in the property are also important, as they allow for vertical or horizontal expansion.

2.4 Energy, thermal comfort, and sustainability in the urban residential sector in Peru.

An increased interest in the sustainability of the built environment has characterised the 21st century. Governments worldwide have implemented measures to ensure that new and existing buildings minimise their environmental impact. The

(a) Los Sauces de Tarapoto, Tarapoto, San Martin, Peru

(b) Las retamas de San Luis, Huancayo, Peru

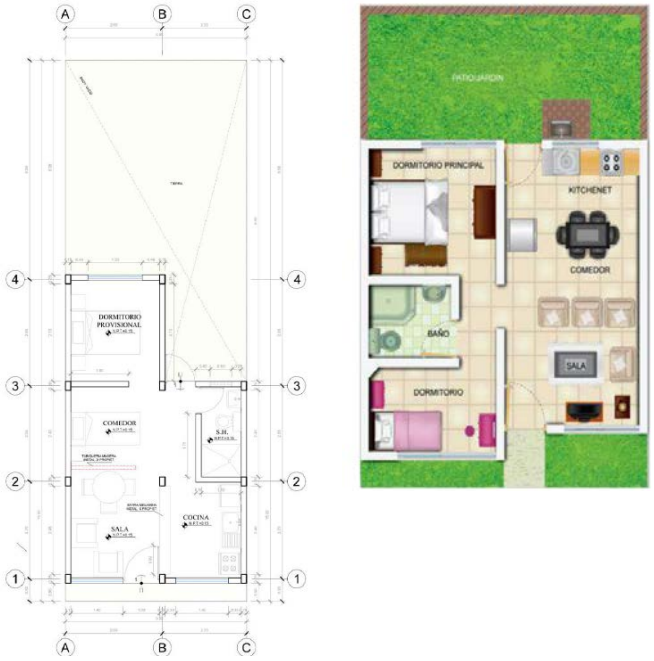


Figure 2.3.6: Plans of two representative AVN projects
Source: (a) Torres García and Urraca Coa (2017) and (b) Avila Saldana (2020)

global south has not been oblivious to this trend. Several policies have been enacted in Peru in the last few years to keep up the pace. These have come in the form of updated building codes, economic benefits for the promoters and consumers of "green" newly built dwellings and guidelines to promote energy efficiency in residential appliances. Nevertheless, the predominant form of building production in the country (incremental self-help) and improving existing building stock seem to be left behind from most discussions regarding the environmental sustainability of the built environment. This section will critically approach the existing policies around sustainability in the urban residential sector in Peru to present them to the reader and reveal the shortcomings that allow for a bottom-up incremental approach to residential sustainability in the Peruvian context. This allows us to conclude that utilising the proposed workflow in Peru is not only useful for the aims of this thesis but also as a means to affect the mindset of decision-makers in the country.

2.4.1 Energy sources and Greenhouse gas emissions reduction commitments in Peru

As part of the Paris Agreement, Peru is committed to reducing its greenhouse gas (GHG) emissions by 40% by 2030 (Ministerio del Ambiente Peru, 2021). This commitment will be achieved in two parts: 30% through non-conditioned investments and expenses with public and private internal and external resources and an additional 10% subject to international external financing. This implies that Peru must not exceed 179 million tons of carbon dioxide equivalent in 2030. Currently, most of the emissions in Peru come from land use, land use change and forestry (LULUCF), which generate almost half of the total emissions (Figure 2.4.1). The energy sector, meanwhile, comes in second place, with nearly a third of the country's emissions. A large portion of this share comes from transport (12.2% out of the total), in which the use of fossil fuels predominates. Emissions coming from electricity production are considered within the "Energy Industries" subcategory (Intergovernmental Panel on Climate Change, 1996), which makes up 6% of the total country emissions. Emissions from fuel combustion in households are included in the "other sectors – energy" category (Intergovernmental Panel on Climate Change, 1996), making up 2.6% of the total country emissions.

In Peru's urban residential sector, three energy sources prevail: Electricity, Liquified Petroleum Gas (LPG), and, increasingly, Natural Gas. In urban areas, 99% of households have access to electricity, and in the median, they use 113 kWh/month (De La Cruz, Salazar, & Santos, 2021). The most common uses for electricity among those who have access to it are refrigeration (68%) and lighting (100%), with very few using it for climatisation/water heating (2%) and cooking (6%) (De La Cruz, Salazar, & Santos, 2021). Meanwhile, LPG is the preferred energy source for cooking in urban areas, where 81% of households have access to it and 99.9% use it for cooking (De La Cruz, Salazar, & Coello, 2021). Urban households with access use an average of 1.06 LPG bottles of 10kg per month, equivalent to 0.48 MMBTU (De La Cruz, Salazar, & Coello, 2021). Lastly, natural gas requires a network connection and has only a few decades of being implemented in the residential sector. Due to this, it is only available in certain areas of major cities. In the Lima and Ica regions, which have direct access to the central processing facility, 1 out of 4 and

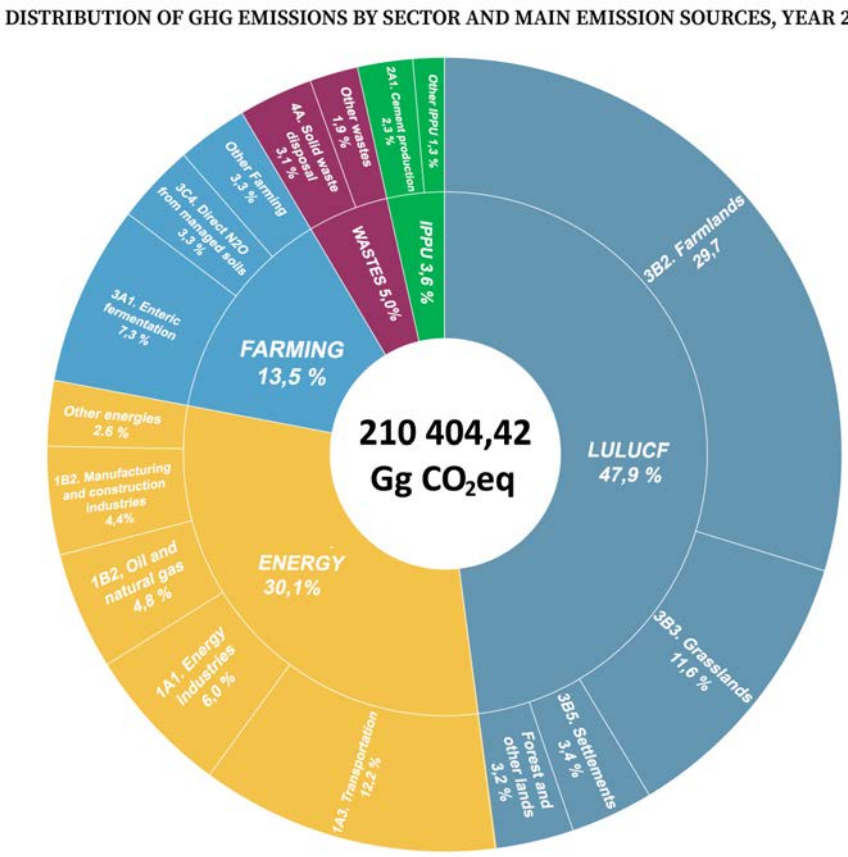


Figure 2.4.1: *GHG Emissions by sector, Peru 2019*
Source: Ministerio del Ambiente Peru (2021)

1 out of 5 households, respectively, use it (De La Cruz, Salazar, & Romero, 2021). In other regions with access, all on the western side of the country, residential connections reach less than 10% of households (De La Cruz, Salazar, & Romero, 2021). The median use of natural gas is 17m^3 per month, and those with access use it primarily for cooking (99.9%), with only 1.1% using it for water heating (De La Cruz, Salazar, & Romero, 2021).

In terms of electric generation, thanks to its geography, Peru can source most of it from hydropower. Increasingly important is Gas, which in the last couple of decades has overgrown Oil as the second source of electricity production and, if the current trend continues, might reach first place in a few years (Figure 2.4.4). Alternative energy sources such as Bioenergy, solar and wind have been recently implemented but still contribute minimally to the energy mix (Figure 2.4.3). Although Gas is thought to be much cleaner than other fossil fuels, it might still be a significant contributor to global warming (Howarth, 2014), and the increasing reliance of the country on this source might put in danger its goals for emissions reduction.

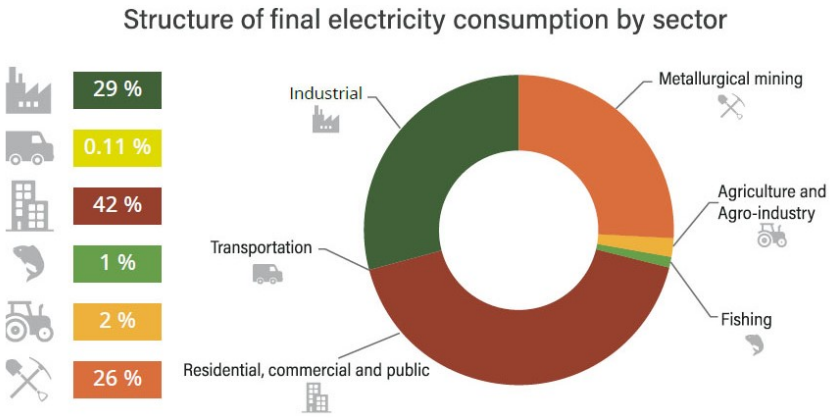


Figure 2.4.2: *Electricity use by sector, Peru 2015*
Source: Ministerio de Energia y Minas Peru (2014)

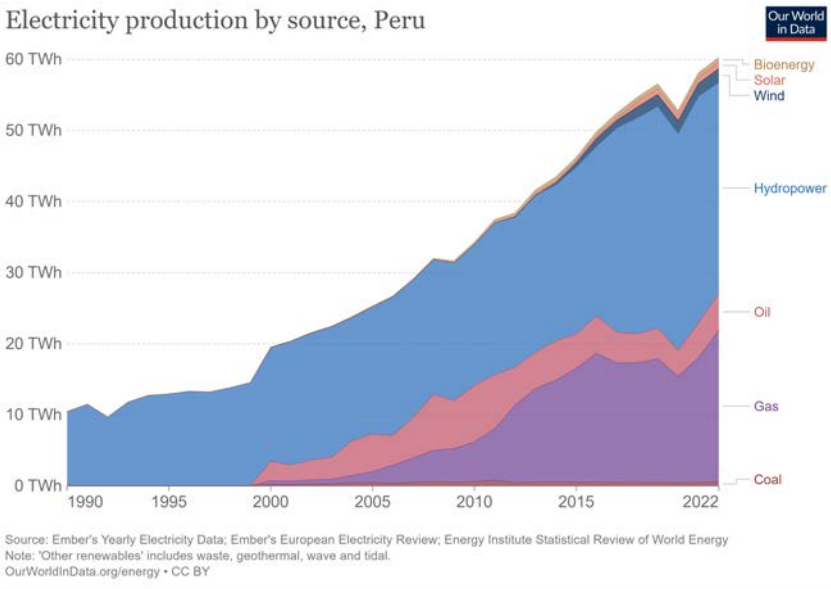


Figure 2.4.3: *Electricity production by source in Peru*
Source: Ritchie, Roser, and Rosado (2022)

2.4.2 Policies and programmes in the urban residential energy sector in Peru

Although Peru's residential sector emissions are relatively small, successive administrations have implemented policies to reduce them and promote energy efficiency. In 2000, through Law No. 27345, the Promotion of the Efficient Use of Energy was declared of national interest. Ten years later, through Supreme Decree Nº 064-2010-EM, the long-term National Energy Policy (2010 – 2040) was approved to ensure the supply of energy, protect the consumer, promote the competitiveness of the national economy and reduce the negative environmental impact of the use and consumption of energy. Therefore, guidelines were published to encourage energy efficiency in

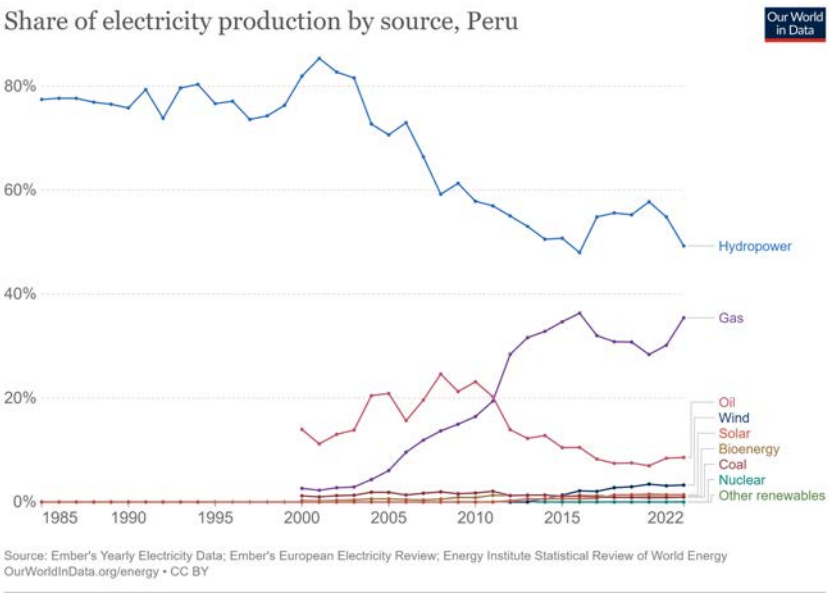


Figure 2.4.4: *Evolution of the share of the sources for electricity production in Peru*
Source: Ritchie et al. (2022)

Table 2.3: Strategies and actions at household scale in the residential sector to achieve GHG emission reduction commitments in Peru

Strategy	Actions
Solar Water Heater Installation	Progressive replacement of 200,000 electric water heaters with solar heaters in the residential sector, representing 29% of the current electric water heater stock.
Replacement of Incandescent light bulbs	Replacement of 4 million incandescent light bulbs with the most efficient technology currently available on the market (LED).
Replacement of Fluorescent Lamps	Replacement of 3 million fluorescent lamps with fluorescent lamps with improved efficiency (T5) and progressively with LED fluorescent lamps in the residential sector.
Installation of Improved stoves in rural areas	Installation of improved stoves in 500,000 homes over five years (100,000 homes/year). Each improved stove has a useful life of 5 years and allows for increasing the efficiency of firewood use, reducing the volume used by 40%.

Source: Ministerio de Energia y Minas Peru (2014)

the industrial, commercial, services and residential sectors. The guide for residences is aimed to "raise awareness, guide and organise the evaluation and quantify the rational use of energy in the different socio-economic levels on which households in the national territory are categorised." (Ministerio de Energia y Minas Peru, 2014). As a document made by the Ministry of Energy, its focus is on domestic appliance efficiency. Nevertheless, it recognises that appliances must act only complementary

to bioclimatic building design in terms of environmental comfort. A relevant point from this guide is that it reveals the critical actions promoted by the ministry at the household scale, which include replacing boilers with solar water heaters, replacing incandescent and fluorescent light bulbs with LED lighting and installing improved kitchens in rural areas (Table 2.3).

The Ministry of Housing, as the entity in charge of promoting specific policies for the housing sector, has also focused on energy efficiency and emissions reduction in urban areas. There are three central policies in the last decade directed towards this objective: The publication of the Technical Code for Sustainable Construction (2015, modified 2021), the addition of the "Thermal and Lighting Comfort with Energy Efficiency" section to the Peruvian Building Code (2014, revised 2022) and the promotion, since 2015, of the "Bono MiVivienda Sostenible", which derives from NCMV and helps families to access dwellings that implement energy and water saving criteria.

Although it might seem that there is a comprehensive policy to achieve comfortable dwellings while minimising energy emissions, the truth is that these policies are entangled and seem to lack a unified direction. A clear example is the initially called "Thermal and Lighting Comfort with Energy Efficiency" section (EM.110) of the National Building Code (RNE) (Ministerio de Vivienda Peru, 2022). Although the name implies that the topics of thermal comfort and energy efficiency must be tackled, the chapter mainly regulates maximum building envelope transmittance values according to a proposed bioclimatic territorial categorisation. As side topics, the chapter includes a formula to ensure that indoor condensation will be avoided, maximum air permeability standards by location, a method to calculate minimal window areas to ensure natural indoor lighting, and standards to communicate the hygrothermal characteristics of construction materials. A final informative annexe aims to guide designers on implementing shading devices to avoid indoor overheating and excessive contrast and reflection from external light sources. Although all these variables contribute to a comfortable and energy-efficient indoor environment, there is an evident lack of consideration for the broader concept. Additional variables, such as passive design measures, are partially ignored. This vagueness was so apparent that in 2022, the name of the section was changed to "Thermal envelope"

(Ministerio de Vivienda Peru, 2022), a title that still does not describe all the topics tackled in the text, but at least is more accurate than the previous one.

Initially, compliance with EM.110 was optative, and so it remained between 2014 and 2021. That year, under Supreme Decree 014-2021-VIVIENDA, the Technical Code for Sustainable Construction (Ministerio de Vivienda Construccion y Saneamiento Peru, 2021a) was updated to make its application compulsory for specific building types. As the energy section of this code refers extensively to EM.110, the latter also became partially mandatory for the same set of buildings. The "sustainable" Code, which has existed since 2014, considers itself complementary to RNE. It was developed to set a base standard for MiVivienda Verde certifications and promote sustainable construction in public infrastructure, remaining just a guide for other project types. It has a more comprehensive approach towards sustainability, including a section on buildings and another on urban developments. The buildings section has chapters on energy and hydric efficiency, green spaces, trash management, eco materials and cycling infrastructure. The chapter on energy efficiency tackles building envelopes, lighting, ventilation, heating and cooling, electromechanical appliances, and gas connections. On the energy section, the only additions to what was already established in EM.110 are setting maximum reflectance values for envelopes, compelling the use of LED lighting and movement sensors, establishing minimal efficiency standards for HVAC appliances, and setting a minimum number of connection points for gas piping in residential buildings. The code also allows the use of independent calculations to prove that a building design will use less energy or water than established standards; nevertheless, as there are no thermal comfort standards in this code or EM.110, it is possible to deliver calculations that disregard any climatisation appliance despite thermal discomfort.

For MiVivienda Verde certifications, FMV establishes additional requisites. This certification is carried out under the "MiVivienda Verde" programme, launched in 2015 due to the cooperation between FMV and the *Agence Française de Dévelopment* (AFD) (Global Alliance for Buildings and Construction, 2020). The programme allows households to benefit from a subsidy if their home is certified to meet specific water and energy-saving criteria. As such, the price of certified "green" housing remains competitive in relation to those lacking certification. In 2018, this initial

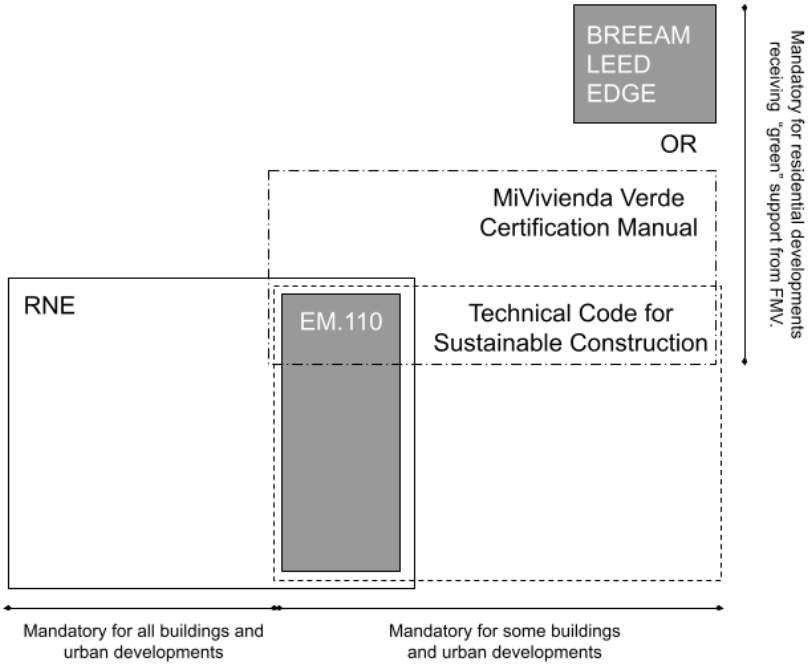


Figure 2.4.5: *Relationship between the Peruvian building codes mentioned in this chapter*

benefit was complemented by an interest rate reduction (Bono MiVivienda Verde) on mortgages devoted to acquiring certified green housing. (Global Alliance for Buildings and Construction, 2020). The certification comes in three levels (Table 2.4), and the criteria to classify projects in these result from considering the local sustainable construction market and the regulatory and technical context (Global Alliance for Buildings and Construction, 2020). Although the internal guide to certify projects (Fondo MIVIVIENDA S.A., 2022b) considers NCMV and AVN developments as potential recipients of the certification, the latest official reports from FMV seem to limit the application of the programme to NCMV products (Fondo MIVIVIENDA S.A., 2022a).

In any case, the certification criteria seem to encompass most variables affecting energy and water savings. In the case of Bioclimatic design, this aims to offer occupants thermal comfort and thus promotes the use of Bioclimatic Strategies according to Climate Zones. Compliance with these criteria can be demonstrated through two possible pathways (Fondo MIVIVIENDA S.A., 2022b): through a bioclimatic study complying with national and/or international standards (ASHRAE 55:2017, LEED, EDGE, BREEAM or Passivhaus) or through the demonstrated application of the

Table 2.4: Certification level and criteria to qualify under the MiVivienda Verde Programme

Grade I+	Domestic water conservation technologies. Energy efficiency technologies in lighting and water heating. Bioclimatic design. Waste management and recycling plans for construction works. Use of eco-materials. Sustainable mobility accessibility and fibre connection.
Grade II+	Grade 1+ conditions. Renewable power generation equipment (PV panels). Energy-efficient electromechanical equipment.
Grade III+	Grade 2+ conditions. Residual water reuse. Mixed uses and communal spaces.

Source: Fondo MIVIVIENDA S.A. (2022b)

guidance notes included in the same document. These notes include several measures to reduce heat gains or control heat losses according to the climatic zones. There are also measures to promote indoor ventilation and reduce the heat island effect.

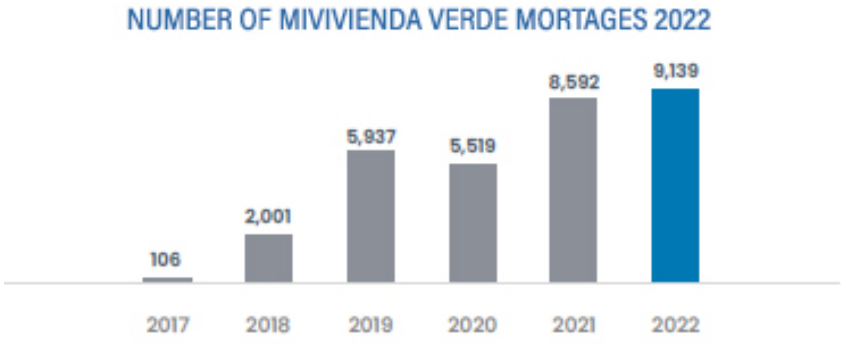


Figure 2.4.6: *Number of MiVivienda Verde loans given in 2022*
Source: Fondo MIVIVIENDA S.A. (2022a)

In addition to the support offered by the central government (if developed within the NCMV framework), buildings accredited with internationally recognised certifications also receive benefits from some local authorities. They usually allow increased built areas when certified by BREEAM, LEED, EDGE and others (Granda, 2023). Although these are interesting measures, their extension is still geographically limited as only some local authorities consider them. At the same time, the cost implied in the certifications suggests that most of these will be high-profile residential projects with few to no social housing components.

2.4.3 Other policies and programmes in Peru

Exemplary initiatives taken in the country but outside the scope of this study are the National Rural Housing Program (PNVR) implemented by the Ministry of Housing and the Guide to Bioclimatic Design Strategies for Thermal Comfort published by the Ministry of Education (Ministerio de Educacion Peru, 2023). The latter is a referential document issued to be used in designing private and public educational centres at every level in the country.

The PNVR, meanwhile, was created in 2012 and modified in 2013 (Supreme Decree No. 016-2013-VIVIENDA). It has the purpose of improving the quality of life of those living below the line of poverty and extreme poverty and whose residence is located in a rural settlement or a dispersed arrangement of dwellings. This is achieved through actions that provide or improve their housing units. The main line of action of this programme has been the construction of "Sumaq Wasi" (Beautiful house) dwellings, whose design prioritises thermal comfort, bearing in mind the climatic regions in which they lay, adapting the use of technological and traditional materials in the components to achieve an ideal bioclimatic model (Portal del Estado Peruano, n.d.). This programme has focused on developing new dwellings based on unique models per region with areas between 26 and 33 m2. So far, there are two areas of intervention, one in the context of the Multisector Plan against Frost and easterly Cold fronts, where up to 24,000 warm seismic-resistant dwellings are being built in rural areas prone to extremely low temperatures, and on the other in the context of the Comprehensive Reconstruction Plan with Changes where up to 8,000 homes located in 12 regions belonging to families affected by the 2017 El Niño Costero phenomena, are being rebuilt with climate-resilience and thermal comfort criteria in mind (Portal del Estado Peruano, n.d.).

2.5 Conclusions

To sum up, basic module developments are a form of aided self-help, present in several countries worldwide but whose history is heavily linked to Peru. As any aided self-help initiative, it has two components: top-down and bottom-up. The earlier is represented by urban layouts, basic infrastructures and "seed" dwellings designed

INDOOR AREA IN ADOBE ROOMS PMAHF 2019

ROOM	FLOOR AREA	
Bedroom 1	5.10	m2
Bedroom 2	5.64	m2
Kitchen-dining	9.08	m2
Buffer area	1.05	m2
Pavement	2.12	m2

Figure 2.4.7: *Adobe model of Rural housing for cold climates*
Source: Portal del Estado Peruano (n.d.)

to be extended. The latter is the incremental project itself, a likely lifelong project in which the owner-occupier embarks once they have taken possession of the property. As a testing ground to explore the possibilities and limitations of incremental development, basic-module developments offer several benefits that respond to their top-down component. As urban layouts and initial dwelling states are certain and fixed, the bottom-up variable can be explored without the noise that uncertainty on these two early variables poses in fully bottom-up developments. At the same time, as public monies are invested in their promotion and development, decision support tools can be used to optimise the resource use devoted to these initiatives. Finally, as the bottom-up component of basic modules is simply incremental development, some conclusions drawn from researching them can be extended to other contexts in which this is also present.

In addition to being strongly linked to the history and evolution of basic-module developments, Peru also shows some physio-geographic characteristics that make it an ideal testing ground for the proposed workflow. The variety of climates in its territory implies that different environmental conditions could be tested using a socio-economic model fed from a single national dataset. Although regional socio-economic differences are present and cannot be ignored, a single national dataset facilitates their identification and provides grounds for their inclusion or exclusion in the model.

Finally, Peru is a country committed to reducing its residential sector emissions.

Various policies and programmes promoted by successive governments demonstrate this. Nevertheless, on the one hand, there does not seem to be a joint approach to this topic and on the other, there is apparent favouritism in promoting ready-finished new developments for the middle classes as the sole carriers of sustainability. The earlier is noted as a lack of a centralised corpus of sustainability standards in the built environment. Instead, several parallel codes exist, with an apparent lack of a joint objective. This makes regulations entangled and confusing, which might be discouraging and complicating their application where no compliance is required. In this parallel coding system, MiVivienda Verde certification standards seem to be the most complete set of guidelines to achieve comfortable residential indoor environments with energy efficiency. Nevertheless, its shortcomings are similar to those of its parent programme, NCMV: It prioritises middle-income households over lower, new developments over retrofit and densification and ready-finished dwellings over incremental development.

Although the MiVivienda Verde certification guidelines imply that the criteria could be applied to AVN, official documents, such as the latest institutional report of FMV, seem to confirm that these are more expectations than reality. Additionally, even if, in the short term, private developers can deliver "green" housing products below the price cap established by AVN, measures taken to achieve a passive design in basic modules will be fruitless, as the modules are designed to grow and change, rendering ex-ante assessments lapsed. As such, if the ABC model is to remain a low-cost housing provision strategy in Peru, a new approach is needed. One that considers how incremental and passive design strategies can interact to bring everyone comfortable and sustainable living spaces. In this sense, the proposed workflow can not only be served by the Peruvian context to test its capabilities but also become a tool that allows the implementation of that approach.

References

Avila Saldana, I. M. (2020). *Implementacion de un programa modelo de desarrollo sostenible para mejorar los proyectos de vivienda social techo propio y mi vivienda en la residencial las retamas de san luis, huancayo-2018* (Master's thesis). Retrieved from http://hdl.handle.net/20.500.12894/6623

Beck, H. E., Zimmermann, N. E., McVicar, T. R., Vergopolan, N., Berg, A., & Wood, E. F. (2018, 10). Present and future köppen-geiger climate classification maps at 1-km resolution. *Scientific Data*, *5*, 180214. doi: 10.1038/sdata.2018.214

Bromley, R. (2003). Peru 1957-1977: How time and place influenced john turner's ideas on housing policy. *Habitat International*, *27*, 271-292. doi: 10.1016/S0197-3975(02)00049-8

Calderon Cockburn, J. (2014). Techo propio, vivienda nueva.¿ por qué falla? *WASI. Revista de estudios sobre vivienda*, *1*(1), 35–55.

Central Intelligence Agency. (2023a, 9). *The world factbook.* Retrieved from https://www.cia.gov/the-world-factbook/field/area/country-comparison/ (Visited: 18-09-2023)

Central Intelligence Agency. (2023b, 9). *The world factbook.* Retrieved from https://www.cia.gov/the-world-factbook/countries/peru/#geography (Visited: 18-09-2023)

De La Cruz, R., Salazar, C., & Coello, F. (2021). *Informe de resultados. consumo y usos de los hidrocarburos líquidos y glp. encuesta residencial de consumo y usos de energía – 2019-2020.* Retrieved from https://cdn.www.gob.pe/uploads/document/file/2560069/ERCUE%20Hidrocarburos%20%202019-2020.pdf?v=1638461831

De La Cruz, R., Salazar, C., & Romero, M. (2021). *Informe de resultados. consumo y usos del gas natural. encuesta residencial de consumo y usos de energía – 2019-2020.* Retrieved from https://cdn.www.gob.pe/uploads/document/file/2665716/ERCUE%20Gas%20Natural%20%202019-2020.pdf?v=1640884110

De La Cruz, R., Salazar, C., & Santos, W. (2021). *Informe de resultados: Consumo y usos de la electricidad 2019-2020.* Retrieved from https://cdn.www.gob.pe/uploads/document/file/2691020/ERCUE%

20Electricidad%202019-2020.pdf

Diaz Figueroa, M. A. (2021, 7). *RevisiÓn de criterios de diseÑo de edificaciones con muros de ductilidad limitada, establecidos en las normas de estructuras peruanas y planteamiento de propuesta de actualizaciÓn.* Retrieved from https://cdn.www.gob.pe/uploads/document/file/2089254/Informe%20de%20la%20revisi%C3%B3n%20de%20criterios%20de%20dise%C3%B1o%20de%20edificaciones%20con%20muros%20de%20ductilidad%20limitada.pdf

Fernandez-Maldonado, A. M. (2014). Incremental housing in peru and the role of the social housing sector. In J. Bredenoord, P. Van Lindert, & P. Smets (Eds.), *Affordable housing in the urban global south: Seeking sustainable solutions* (pp. 271–285). Routledge.

Fernández Maldonado, A. M. (2015, 6). Las barriadas de lima como estímulo a la reflexión urbana sobre la vivienda: revisitando a turner y de soto. *Universidad Nacional de Ingeniería*.

Fernández-Maldonado, A. M., & Bredenoord, J. (2010, 7). Progressive housing approaches in the current peruvian policies. *Habitat International*, *34*, 342-350. doi: 10.1016/j.habitatint.2009.11.018

Fondo MIVIVIENDA S.A. (n.d.-a). *Buscador de viviendas - nuevo credito mivivienda.* Retrieved from https://www.mivivienda.com.pe/PORTALWEB/usuario-busca-viviendas/buscador-home.aspx?op=ncmv (Visited: 10-2022)

Fondo MIVIVIENDA S.A. (n.d.-b). *Buscador de viviendas - techo propio.* Retrieved from https://www.mivivienda.com.pe/PORTALWEB/usuario-busca-viviendas/buscador-home.aspx?op=nstp (Visited: 10-2022)

Fondo MiVivienda S.A. (2017). *Perfil del cliente fondo mivivienda 2014, 2015 y 2016.* Retrieved from https://www.mivivienda.com.pe/PortalCMS/archivos/documentos/Perfil%20del%20cliente%20FMV%202017.pdf

Fondo MIVIVIENDA S.A. (2022a). *Memoria institucional 2022.* Retrieved from https://www.mivivienda.com.pe/PORTALCMS/archivos/documentos/8585170892055792261.PDF

Fondo MIVIVIENDA S.A. (2022b). *Procedimiento de certificaciÓn de proyectos*

del programa mivivienda sostenible. version 9.0. Retrieved from https://cdn.www.gob.pe/uploads/document/file/4083413/Procedimiento%20de%20Certificaci%C3%B3n%20vigente%20-%209no%20Procedimiento.pdf

García-Huidobro, F., Torres Torriti, D., & Tugas, N. (2011, 5). The experimental housing project (previ), lima: The making of a neighbourhood. *Architectural Design*, *81*, 26-31. doi: 10.1002/ad.1234

Global Alliance for Buildings and Construction. (2020, 9). *Green mortgages for households – fondo mivivienda.* Retrieved from https://globalabc.org/resources/publications/stimulus-programmes-green-buildings-best-practice-examples

Gobierno Regional de La Libertad. (2014, 10). *MÁs de 4 mil familias podrÁn contar con vivienda propia en alto salaverry.* Retrieved from https://www.regionlalibertad.gob.pe/noticias/regionales/4700-mas-de-4-mil-familias-podran-contar-con-vivienda-propia-en-alto-salaverry (Visited: 18-09-2023)

Granda, C. (2023, 6). *Incentivos gubernamentales para edge.* Retrieved from https://leaflatam.com/incentivos-gubernamentales-para-edge/ (Visited: 18-09-2023)

Grupo de Analisis para el Desarrollo. (2020, 11). *Hacia una nueva política de vivienda en el perú: Problemas y posibilidades.* Retrieved from https://adiperu.pe/wp-content/uploads/Documento-Base-Hacia-una-nueva-Politica-de-Vivienda-en-el-Peru-Problemas-y-posibilidades.pdf

Gyger, H. (2019a, 1). Building a better barriada, 1968–1975. In *Improvised cities: Architecture, urbanization, and innovation in peru* (p. 242-287). University of Pittsburgh Press. doi: 10.2307/j.ctvdjrpj7.13

Gyger, H. (2019b). Other paths, 1980–1986. In *Improvised cities: Architecture, urbanization, and innovation in peru* (pp. 329–373). University of Pittsburgh Press. Retrieved 2024-03-06, from http://www.jstor.org/stable/j.ctvdjrpj7.15

Harris, R. (2003). A double irony: The originality and influence of john f.c. turner. *Habitat International*, *27*, 245-269. doi: 10.1016/S0197-3975(02)00048-6

Howarth, R. W. (2014, 6). A bridge to nowhere: methane emissions and the

greenhouse gas footprint of natural gas. *Energy Science and Engineering*, *2*, 47-60. doi: 10.1002/ese3.35

INEI. (n.d.). *Definiciones y conceptos censales básicos.* Retrieved from https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1541/definiciones.pdf

INEI. (2022). *Peru: 50 años de cambios, desafios y oportunidades poblacionales* (D. Carhuavilca, A. Sánchez, C. Gutiérrez, D. Durand, & D. Mendoza, Eds.). Instituto Nacional de Estadistica e Informatica (INEI). Retrieved from https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1852/libro.pdf

INEI. (2023, 7). *Situación de la población peruana al 2023. una mirada hacia los jóvenes.* Retrieved from https://cdn.www.gob.pe/uploads/document/file/4824324/Situaci%C3%B3n%20de%20la%20Poblaci%C3%B3n%20Peruana%20al%202023.%20Una%20mirada%20hacia%20los%20j%C3%B3venes%3A%20Contenido%20y%20Presentaci%C3%B3n.pdf

Intergovernmental Panel on Climate Change. (1996). Understanding the common reporting framework. In *Revised 1996 ipcc guidelines for national greenhouse gas inventories: Reporting instructions* (Vol. 1). Intergovernmental Panel on Climate Change. Retrieved from https://www.ipcc-nggip.iges.or.jp/public/gl/guidelin/ch1ri.pdf

Kozak, D. (2016, 12). John f.c. turner y el debate sobre la participación popular en la producción de hábitat en américa latina en la cultura arquitectónico-urbanística, 1961-1976. *URBANA: Revista Eletrônica do Centro Interdisciplinar de Estudos sobre a Cidade*, *8*, 49. doi: 10.20396/urbana.v8i3.8646011

Leon, G. (2022, 11). *Políticas habitacionales benefician a personas con más ingresos, mientras aumenta el déficit habitacional en sectores d/e.* Retrieved from https://data.larepublica.pe/politicas-habitacionales-benefician-a-personas-con-mas-ingresos-mientras-aumenta-deficit-en-sectores-D-E-bonos-vivienda/

Ministerio de Agricultura y Riego Peru. (n.d.). *Perfil ambiental del perú.* Retrieved from https://www.midagri.gob.pe/portal/datero/60-sector-agrario/introduccion/343-perfil-ambiental-del-peru (Visited: 18-

09-2023)

Ministerio de Educacion Peru. (2023). *Guía de estrategias de diseño bioclimático para el confort térmico.* Retrieved from https://hdl.handle.net/20.500.12799/9323

Ministerio de Energia y Minas Peru. (2014). Sector residencial. In *Elaboración de proyectos de guías de orientación del uso eficiente de la energía y de diagnóstico energético* (Vol. 1). Ministerio de Energia y Minas. Retrieved from https://www.minem.gob.pe/minem/archivos/file/DGEE/eficiencia%20energetica/publicaciones/guias/7_%20guia%20sector%20residencial%20DGEE-1.pdf

Ministerio de Vivienda Construccion y Saneamiento Peru. (2021a, 7). *Decreto supremo que aprueba el código técnico de construcción sostenible.* Retrieved from https://busquedas.elperuano.pe/dispositivo/NL/1976353-3

Ministerio de Vivienda Construccion y Saneamiento Peru. (2021b, 12). *Modifican el reglamento operativo para acceder al bono familiar habitacional, para la modalidad de aplicación de adquisición de vivienda nueva.* Retrieved from https://busquedas.elperuano.pe/dispositivo/NL/2024851-1

Ministerio de Vivienda Construccion y Saneamiento Peru. (2022, 5). *Decreto supremo que actualiza los valores de las viviendas y actualiza excepcional y temporalmente los valores del bono del buen pagador.* Retrieved from https://busquedas.elperuano.pe/dispositivo/NL/2063154-1

Ministerio de Vivienda Peru. (2022). Norma tecnica em.110 envolvente termica. In *Reglamento nacional de edificaciones (rne).* Ministerio de Vivienda, Construccion y Saneamiento - Peru.

Ministerio del Ambiente Peru. (2021). *Contribuciones determinadas a nivel nacional del perÚ. reporte de actualizaciÓn periodo 2021 – 2030.* Retrieved from https://cdn.www.gob.pe/uploads/document/file/1675213/Actualizaci%C3%B3n%20de%20las%20NDC%20del%20Per%C3%BA%20al%202030.pdf?v=1663622045

NASA Earth Observatory. (2023, 9). *The highest settlement in the world.* Retrieved from https://earthobservatory.nasa.gov/images/145864/the-highest-settlement-in-the-world (Visited: 18-09-2023)

Portal del Estado Peruano. (n.d.). *Programa nacional de vivienda rural. información institucional.* Retrieved from https://www.gob.pe/institucion/pnvr/institucional (Visited: 18-09-2023)

ProjectPeru. (2023). *Maps of peru.* Retrieved from https://projectperu.org.uk/about-peru/maps-of-peru/ (Visited: 18-09-2023)

Ritchie, H., Roser, M., & Rosado, P. (2022). Energy. *Our World in Data.* (https://ourworldindata.org/energy)

Torres García, I., & Urraca Coa, C. (2017). *Proyecto inmobiliario de vivienda social techo propio y lotizaciones en la ciudad de tarapoto* (Master's thesis). Retrieved from http://hdl.handle.net/20.500.12404/8554

Turner, J. F. C., & Fichter, R. (1972). *Freedom to build : dweller control of the housing process.* Macmillan.

Chapter 3

Socio-economic Model

The current chapter presents the development of a plausible socio-economic model able to feed the intermediary component of the Thesis methodology with country-specific socio-economic data for multiple simulation years. This model deals with the "uncontrolled" situations faced by a household owning an incremental dwelling (such as the variation in the number of members and their financial circumstances). To determine these variables at each time step, the model relies on event probabilities, categorical probability distributions and transition probabilities among categories, taken mainly from Peru's national household survey (ENAHO), complemented by other specific surveys and official projections and datasets.

The model uses microsimulation as a method and works at two complementary scales: at the individual level, a demographic model simulates the lifecycle of individuals within a household, determining when they are born, give birth, become economically active, move out and die. At the household level, an economic model predicts the likely category for each household in relation to the mean income and expenses per capita for their corresponding socio-economic level (SEL). The results of both models combined give a complete picture of the likely demographic and financial progression over 25 years for 48 synthetic households living in a generic intermediate city in Peru. The chapter ends by presenting conclusions and recommendations on implementing this model in the general workflow and for further development and improvement.

3.1 Microsimulation as a method

The methodology (Figure 1.3.1) assigned workflow Part A to a socio-economic model whose aim was to feed the intermediary component with plausible data on "need" and "financial capacity" for all the periods requested. While the concept of "financial capacity" was tackled in the introductory chapter, the concept of "need" was kept aside for future development. As our modelling task requires an objective and measurable concept (so quantitative variables can be derived), the present research bounds "need" to "space need" and derives the latter from the concept of overcrowding. This choice was made as overcrowding implies two measurable variables: The number of people inhabiting a dwelling and the size of that dwelling. At the same time, it allows easily deriving simple categories, a dwelling can be "overcrowded" or "not overcrowded". In that sense, an overcrowded dwelling is in urgent need of expansion, while a not-overcrowded one could be if other conditions are met.

These additional conditions and the dwelling size variable are left for the interaction between the intermediary component and the workflow Part B. As such, the socio-economic model deals with the two following variables:

- The number of people inhabiting a dwelling, responding to the "need" factor, and
- Their joint financial capacity.

As these variables are needed for multiple periods that depict the dwelling evolution, the socio-economic model obtains these by:

- Simulating the demographic evolution of households, and
- Simulating the evolution of their financial circumstances

Other additional requirements set by the methodology are an Agent-Based approach and the use of site-specific empirical data. Given these and available datasets (presented further in this chapter), the ideal candidate method was microsimulation, a social simulation-based tool "with a micro-unit of analysis" (O'Donoghue, 2014). Although closely related to other individual-level modelling approaches, like cellular automata and agent-based models (ABMs), microsimulation is characterised

by its focus on simulating the socio-economic attributes of the agents rather than their interactions. Even if they are not the same, ABMs and microsimulation are increasingly moving towards a common ground thanks to their commonalities (Li, O'Donoghue, & Dekkers, 2014) and allow the creation of workflows that complement their strengths.

Microsimulation models can be broadly classified as static and dynamic. While the earlier category models "the day after effect" of a policy change, thus involving "the interaction of population and policy complexity, but abstract from behavioural response or other endogenous change" (O'Donoghue, 2014), the latter models the behaviour of micro-units over time. To achieve this, Dynamic microsimulation typically projects population samples over time so that "one can, for example, examine future income distributions under different economic and demographic scenarios." (Li et al., 2014). This implies that any dynamic microsimulation model has an underlying model of demographic behaviour.

Unsurprisingly, these demographic models are themselves a form of dynamic stochastic microsimulation. There, "the unit of observation is the individual, and the main purpose is to create simulated populations with kinship networks to answer questions of interest to demographers and other social scientists." (Mason, 2014). To achieve this, demographic microsimulations simulate "all of the usual demographic events" (births, marriages, divorces and deaths) and transitions between arbitrarily defined subpopulations (Mason, 2014). Their stochastic nature lies in their reliance on distribution and transition probabilities to achieve their simulation task. Demographic microsimulation usually implies an ageing process, which consists of "updating a database to represent current conditions" or "projecting a database for one or more years to represent expected future conditions." (Li et al., 2014). This ageing process can be static or dynamic, the earlier of which implies "adjusting the weights of the observations so that the simulated population distribution matches the macro-aggregates" (Li et al., 2014). At the same time, the latter "changes the attributes of the individuals instead of altering their weights." (Li et al., 2014). Whilst a dynamic approach complies more closely with a purely agent-based nature, it usually depends on an alignment mechanism to "keep its aggregate outputs in line with predictions from macro-models." (Li et al., 2014).

Dynamic microsimulation can be classified according to their underlying concept of time step and dependence on new agents beyond their seed community (Li et al., 2014). Regarding the latter, dynamic models can be either open or closed. A model is considered 'closed' if it uses a fixed set of individuals to create and maintain social links (but usually allows new individuals as new-borns and immigrants), whilst an open model "starts with a base population and new individuals are generated exogenously" if required (Li et al., 2014). In demographic models, this differentiation is particularly relevant in spouses' selection, as in closed models, couples can only be formed between existing agents. Although an open approach better depicts a real-world scenario (where individuals of different communities can form couples), this method may produce a sample that does not represent the studied community (Li et al., 2014). For this reason, a closed model approach is recommended when simulating household-level variables.

Regarding their relationship with the temporal dimension, Dynamic microsimulation models can be discrete or continuous. In discrete time microsimulation, individual attributes are changed once per period (usually based on transition probability matrices), whilst, in continuous time microsimulation, we rely on survival models, which determine the length of time an individual will face their current state (Li et al., 2014). Discrete-time models show some limitations, including their unrealistic assumption of a sequential order of life events, the lack of consideration of the transition path happening in between the registered time steps, and the difficulties implied in ordering the application of transition matrices (like the probabilities of getting pregnant after marriage or marrying after a premarital pregnancy) (Li et al., 2014). Beyond these theoretical weaknesses, most available datasets have data collected at discrete time steps, thus facilitating the construction of discrete-time models over continuous ones.

To interact with macroeconomic models, Microsimulation models increasingly rely on either an alignment process or computational general equilibrium (CGE) feedback (Li et al., 2014). Alignment is "a process of constraining model output to conform more closely to externally derived macro-data" (Scott, 2001). Applying this process has become standard over the past decade, as it allows "the use of dynamic microsimulation models for the policy assessment together with (and making

use of) the simulation results of macroeconomic models" (Li et al., 2014). This is particularly relevant for continuous variables (such as earnings or expenses), which align with a fix ratio to meet a projected average or distribution (Li et al., 2014).

In conclusion, microsimulation is a social simulation modelling technique based on the micro-unit, be that the individual, a household or a company. Although not precisely the same, microsimulation models and ABMs are closely related and can complement each other due to their agent-based nature. Microsimulation is broadly classified as static and dynamic, the latter of which is more relevant to the current research as it deals with the time dimension. Most dynamic microsimulation models rely on a demographic model, which, by definition, implies the agents' ageing process. These models are usually stochastic and allow us to analyse demographic trends and their consequences as emergent phenomena.

As such, Dynamic Microsimulation (DM) seems to be able to generate data that complies with all the requisites for our socio-economic model: It can simulate the demographic evolution of households and the change in their members' numbers. It also simulates the growth of income, expenses, and the shift in employment status of the individuals. At the same time, we are dealing with an agent-based modelling approach, which agrees with the one used in methodology.

To be more precise, the current research uses discrete-time and closed demographic DM. The economic simulation component, meanwhile, depends on an alignment process to keep up with aggregate-level projections; since income and expense figures are usually affected by macro effects such as inflation and economic growth.

3.2 Data sources

Due to its focus on the micro-scale, microsimulation relies on individual or household data. The usual sources include household surveys, administrative, census and synthetic data. Panel data is preferred as it records changes over time (Li et al., 2014). Concerning income data, literature warns against using self-reported information, promoting instead the use of administrative data as tax report statistics (Li et al., 2014). Nevertheless, this may be a misleading recommendation for our case study. Administrative reports are highly reliable in developed countries, where

labour markets are extensively formal, and income tax is the primary source of government revenue. In the developing world, where informality is the rule and government are funded mainly by trade and consumption taxes, administrative data only registers a small part of the reality. Finally, the sample size is essential when dealing with dynamic microsimulation models. Beyond the total number of respondents, one must be aware of the respondents within groups and sub-groups that allow for determining more specific transition probability matrices. In this sense, a model can only be as detailed as the origin dataset allows.

3.2.1 ENAHO

Given these recommendations, Peru's National Household Survey (ENAHO) was selected as the primary source dataset. This survey is executed regularly by Peru's National Institute of Statistics and Informatics (INEI), which since 1995 has been monitoring the indicators that allow us to know the evolution of poverty, well-being and living conditions of households in Peru. Since 2003, the survey has been conducted continuously throughout the year. It consists of five questionaries with a total of about 350 questions that range from describing the characteristics of the dwelling, the household and the individuals' perceptions of democracy and participation. Since 2011, there has been the inclusion of a Panel component. The panel survey aims to include about 30% of the participant households of the non-panel survey. The survey's microdata is freely available to download on INEI's webpage and thus has been widely explored by various research.

In this research, ENAHO's data serves to construct distribution and transition probabilities to project the characterisation of the population in time. It is also used to determine the likely income/expenses at the household level during various years. Finally, the dataset serves to obtain simple trend-line projections of income and expenses to be utilised in an alignment process for the economic component of the model.

Two factors were considered in selecting the appropriate time range for the data to be used. On one hand, it was vital to avoid relying on data generated during the worst part of the COVID-19 pandemic. This is because various socio-economic variables were heavily affected by the measures taken to combat this disease. Sec-

ondly, to allow results validation, we had to rely on the maximum period of panel data published in a single dataset. As panel data comes in five-year period batches, it was decided that the selected datasets, both panel and non-panel, will cover the period starting in 2015 and finishing in 2019.

The downloadable data of ENAHO, both in its panel and non-panel format, is divided into six modules, the first five covering each of the independent questionnaires applied at the individual level and a sixth aggregating the data at the household level. From all these, three are useful for the current model. The summary (aggregated) module serves us well considering that the household is the core of the agent-based approach. Meanwhile, the so-called Module 200, which records the population's general characteristics, served to construct the base of the demographic model. Finally, Module 500, which records the income-generating status of surveyed adults, allowed the inclusion of this variable, useful for both the demographic and economic components of the model.

Summary Dataset

The summary dataset was used both on its panel and non-panel format. The following relevant variables were selected from this module:

- Social strata of the household (from now on referred to as socio-economic level)
- Location strata of the household (from now on referred to as urban agglomeration size of the location)
- Number of household members
- Number of income-generating household members
- Total net monetary income of the household in the last twelve months and
- Total household monetary expenses in the last twelve months.

These six variables can be classified into three categories:

- Filtering variables (socio-economic level and urban agglomeration size)
- Demographic variables (Number of household members and income-generating household members), and

- Economic variables (income and expenses)

The first group of variables allows filtering data so that the surveyed population characteristics get close enough to that of the case studies without affecting the statistical relevance of the resulting probability matrices. These matrices go to up to three levels of specificity, so it was essential to keep the filters minimal.

The first of these filters refers to the socio-economic level (SEL) of a household, which is a categorical index composed of five levels (A to E, where A is correlated with the highest incomes and E with the lowest) widely used in sociological, political and market studies in Peru. Created by APEIM (Peruvian Association of Market Intelligence Companies), this classification is calculated based on material and immaterial household resources that accurately predict income, consumption and living standards (APEIM, 2020). As such, it is used by governmental organisations to target population segments with housing programmes (Instituto CUANTO, 2018). Techo Propio's New Dwelling programme (AVN) is focused on SELs C and D, while Nuevo Credito MiVivienda (NCMV) does so on SELs B and C (see Chapter 2 for further details). Nevertheless, as not all projects financed by these programmes are incremental, a further level of specificity is needed.

Thanks to statistical bulletins from MiVivienda fund (Fondo MIVIVIENDA S.A., 2022), we know that most single-family dwellings financed by AVN are located either in the peripheries of Metropolitan Lima or in medium-sized cities in the rest of the country (see Chapter 2). As the workflow aims at climate diversity to test its functioning, three intermediate cities with diverging climate patterns were selected for the final run of the workflow (Chapter 6). As such, the socio-economic model relies on the medium-sized cities category to filter the data. In statistical terms, these locations fall in categories three and four of ENAHO's variable. This corresponds to locations with a population between 20k and 100k inhabitants (or between 4k and 20k dwellings in older survey versions). Further filtering was not considered as the number of participating households was already falling worryingly, particularly for panel data (Table 3.1).

The second group of variables, referred to as demographic, register the number of household members and of income-generating household members. As for household membership, this is determined by considering those who self-reported as belonging

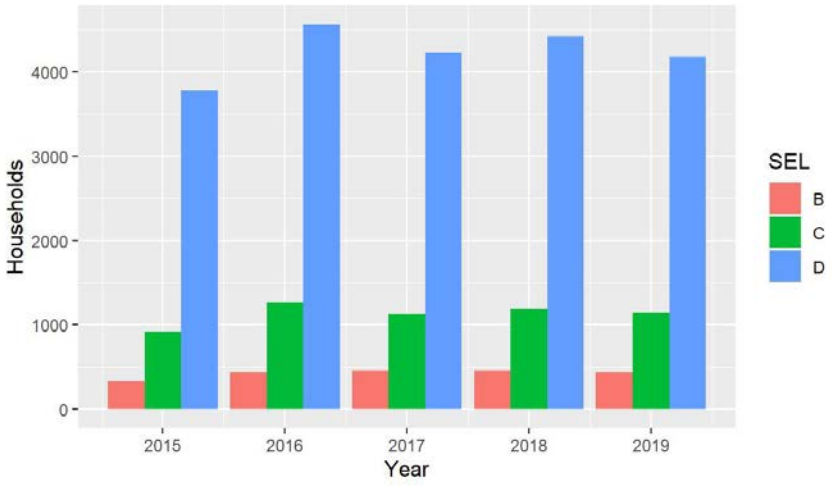


Figure 3.2.1: *Number of participant households after applying filters in ENAHO's non-panel data by year and SEL*
Source: Data taken from INEI (2016b, 2017a, 2018a, 2019b, 2020a) SUMARIA dataset

Table 3.1: Number of participant households in ENAHO's SUMARIA panel data for two consecutive years (2015-2016) by socio-economic level (after applying filters).

SEL	**Participating households**
B	97
C	225
D	1143
Total	1465

Source: Data taken from INEI (2019a) SUMARIA dataset

to the household minus those who self-reported as domestic workers or salaried by the household chief (INEI, 2016a). Figures 3.2.2 and 3.2.3 show the number of household members in non-panel and panel data, respectively.

The income generator status of a member is determined by applying a formula defined in the summary dataset methodological document (INEI, 2016a). This formula omits domestic workers, individuals salaried by the household chief and those under 14. Those not discriminated by those filters can be considered income generators if they have had any income in the last 12 months, be this ordinary or extraordinary, monetary or in kind. For illustrative purposes, the current document includes a ratio of income generator members over the total number of members (Figure 3.2.4 for non-panel data and Figure 3.2.5 for panel data).

The last group of variables from ENAHO's summary dataset are those referred to the household economic status. Household income registers the sum of all ordinary and extraordinary monetary income after taxes self-reported by every household

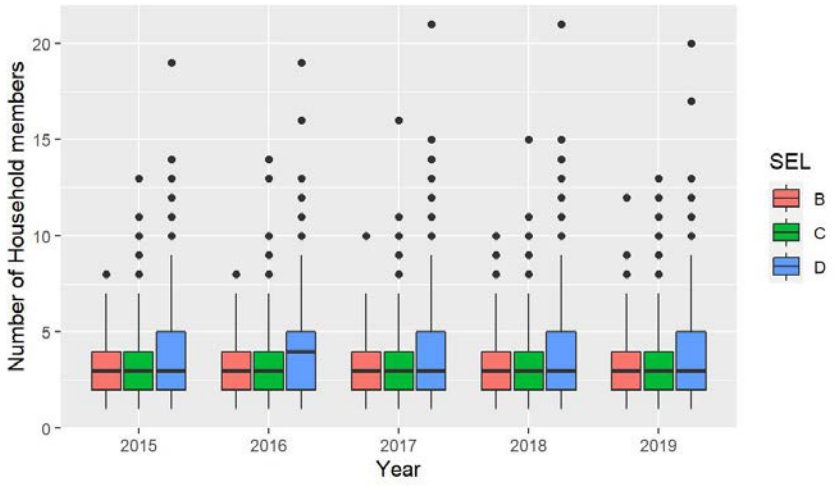


Figure 3.2.2: *Number of household members in participant households by year and SEL, ENAHO non-panel*
Source: Data taken from INEI (2016b, 2017a, 2018a, 2019b, 2020a) SUMARIA dataset

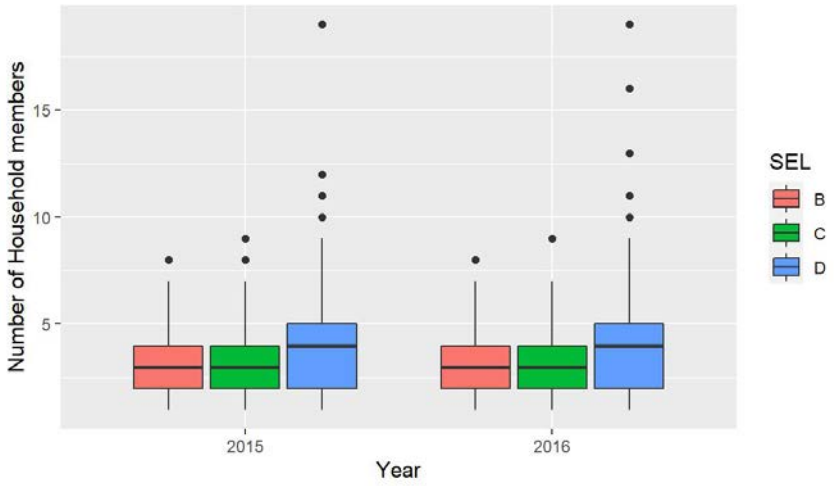


Figure 3.2.3: *Number of household members for participating households in two consecutive years (2015-2016) by year and SEL, ENAHO panel.*
Source: Data taken from INEI (2019a) SUMARIA dataset

member for the last twelve months (INEI, 2016a). Similarly, household expenses are the sum of all the monetary expenses self-reported by every household member for the previous twelve months (INEI, 2016a).

As this thesis deals with the interrelationship between available monetary means and the number of occupants of a dwelling, two normalised variables are generated from the existing data. First, by dividing the income among the number of income generators in a household, we get the average income each income generator member produces. Equally, by dividing the household total expenses among the number of household members, we get the average expenses that each member incurs to the

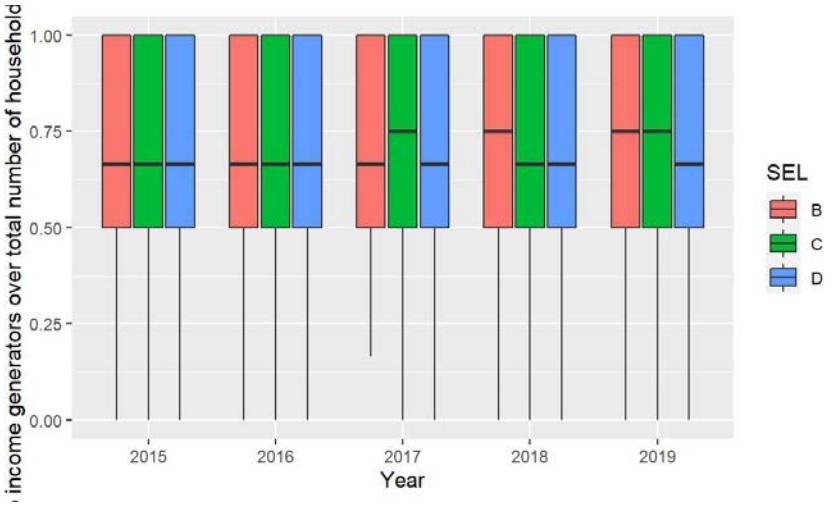


Figure 3.2.4: *Number of income generators over total household members for filtered participant households, ENAHO's non-panel data.*
Source: Data taken from INEI (2016b, 2017a, 2018a, 2019b, 2020a) SUMARIA dataset

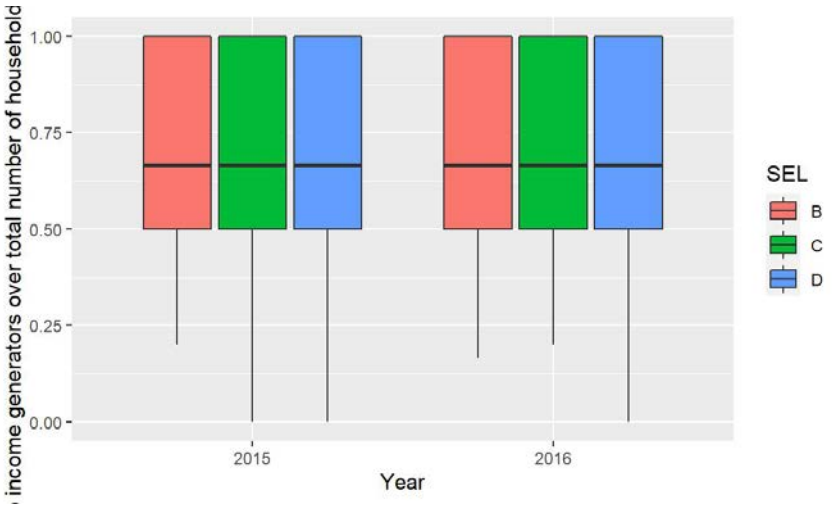


Figure 3.2.5: *Number of income generators over total household members for filtered participant households in 2015 and 2016, ENAHO PANEL*
Source: Data taken from INEI (2019a) SUMARIA dataset

household budget. By taking out the outliers, representing extraordinary income or expenses from rare events such as property transactions, we can get the average by SEL. Figure 3.2.6 registers the evolution of these averages per SEL for every year taken from non-panel datasets. We can notice a clear trend of growth for both figures in all socioeconomic levels, which agrees with the macro-economic growth of the country in those years. The differentiation of these figures among SELs is also made explicit.

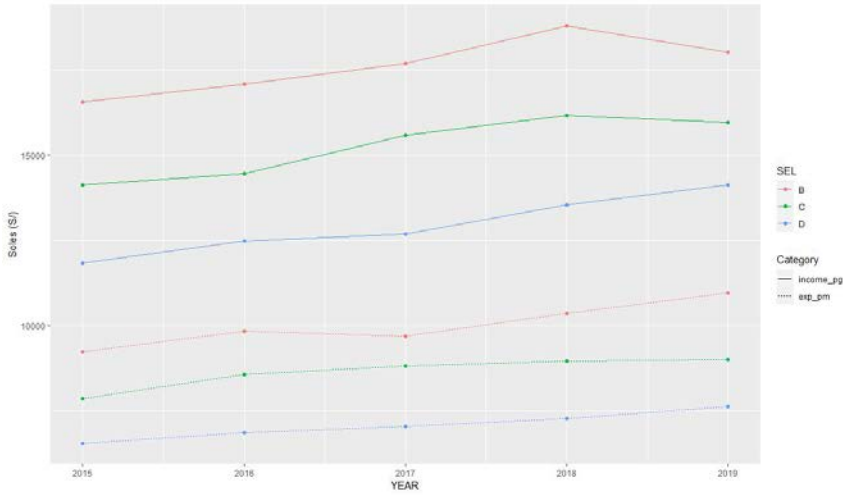


Figure 3.2.6: *Average yearly income per income generator in a household and average annual expenses per household member (in local currency, filtering outliers) for filtered participating households in ENAHO non-panel data by year and SEL.*
Source: Data taken from INEI (2016b, 2017a, 2018a, 2019b, 2020a) SUMARIA dataset

Module 200

The second dataset taken into account from ENAHO is the so-called module 200. This registers general data of household members at the individual level. The non-panel version of this dataset is used to generate distribution probabilities for synthetic individual generation, whilst the panel version allows generating moving-out probability tables of individuals by age and gender categories. The selected variables from this dataset are:

- Location strata of the dwelling (size of the urban agglomeration by population)
- Age of the individual
- Gender of the individual

Module 200 does not contain information about the SEL of individuals, as this variable only classifies whole households. Because of this, the only filter variable for this module was the location strata, so we only included intermediate cities according to the same filter applied to the summary dataset. Age categories were assigned in five-year periods to integrate ENAHO's data with the other datasets. As non-panel data determines the characteristics of the individuals at the starting time step, only data from the earlier study period is needed. So, after applying the only

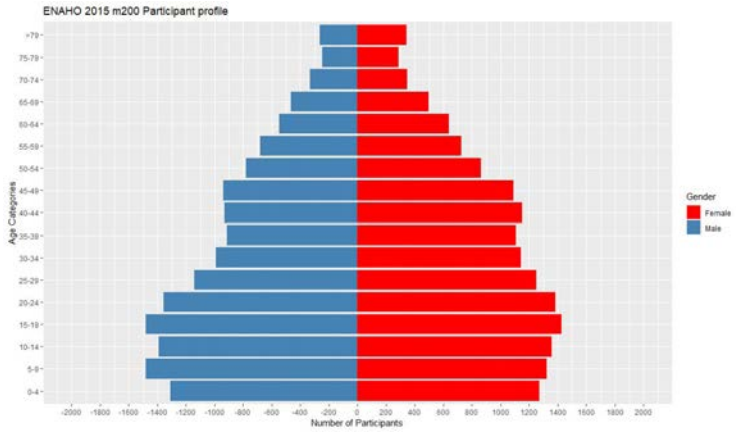


Figure 3.2.7: *Population pyramid for filtered participants.*
Source: Data taken from INEI (2016b, 2017a, 2018a, 2019b, 2020a) Modulo-200 dataset

filter to 2015 non-panel data, we ended up with 31,452 participants, categorised by age and gender according to Figure 3.2.7.

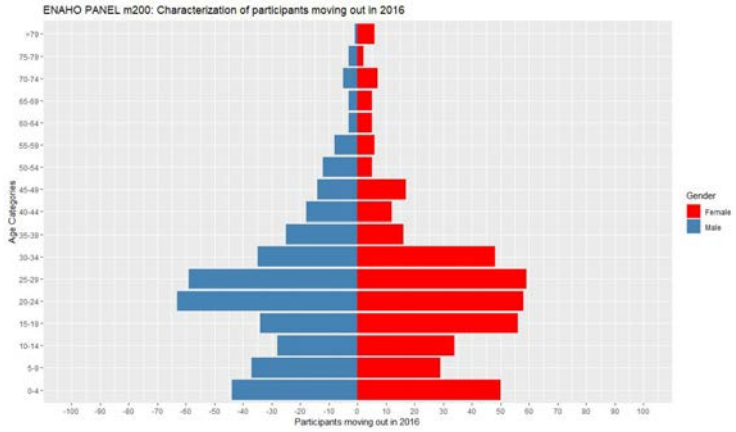


Figure 3.2.8: *Population pyramid of participants who were present in 2015 but reported moving out to a different household in 2016.*
Source: Data taken from INEI (2019a) Modulo-200 dataset

As panel data is needed to calculate the move-out probabilities of individuals, and the datasets come in five-year batches, we need to apply an additional filter, showing only the individuals from households participating in two consecutive years. We chose participating households in 2015 and 2016 to align with the previously selected years. By filtering those individuals who were present in 2015 but were reported to have "moved out to another household" in 2016, we get a profile of those who have moved out (807 individuals in total) by age and gender category (Figure 3.2.8).

Module 500

The third and last dataset used from ENAHO is module 500. This part of the survey applied exclusively to individuals 14 or older, inquiries about their income and work status. From the non-panel version of this module, the model obtains categorical distribution for the income-generator status. In contrast, the panel version gets transition probabilities for the same variable. As this status was calculated initially only at the summarising stage of the dataset creation, it only exists as an aggregate number for the household. Due to this, the variable outcome was recalculated for individuals using the formula in the summary methodological guide (INEI, 2016a). Thus, the selected variables for this module included the ones needed for the formula, plus:

- Age of the individual
- Gender of the individual
- Location strata of the dwelling (urban agglomeration size by population)
- Relationship of the individual with the household chief

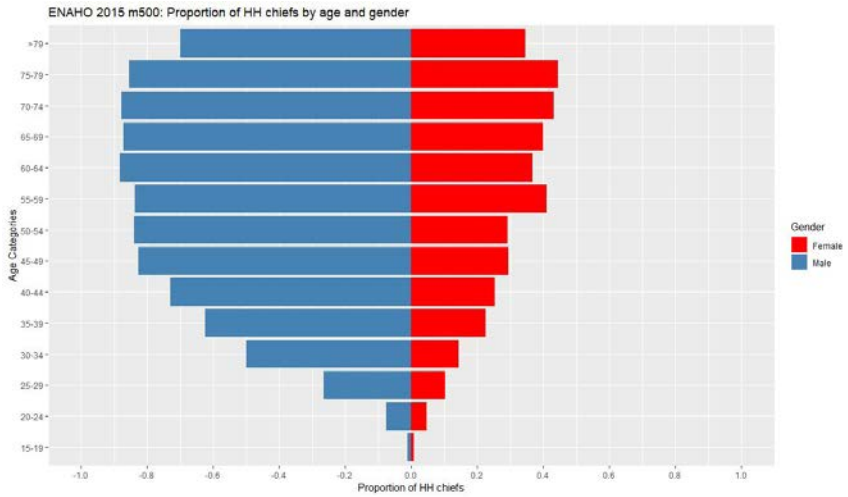


Figure 3.2.9: *Proportion of participants by age/gender categories who are household chiefs.*
Source: Data taken from INEI (2016b, 2017a, 2018a, 2019b, 2020a) Modulo-500 dataset

The individual's relationship with the household chief was selected as the chief status significantly affects the chances of an individual becoming and staying an income generator (Figure 3.2.12). Additionally, the chief status allowed a model that

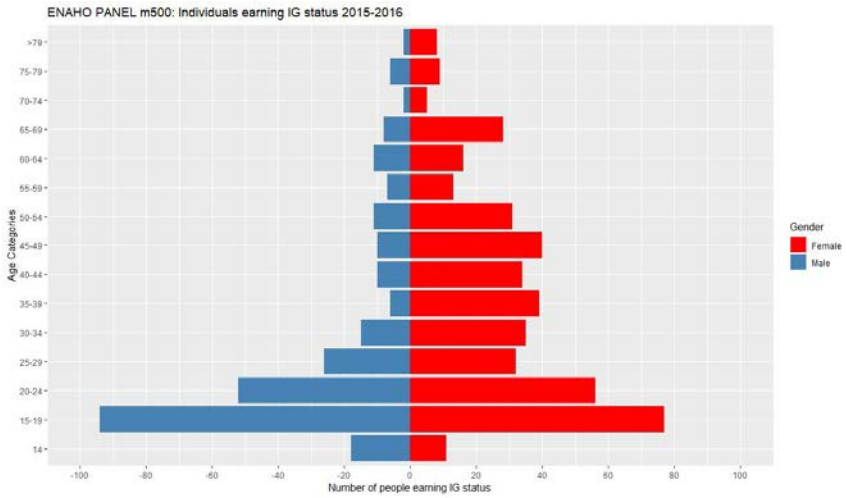


Figure 3.2.10: *Population pyramid of individuals that were not income generators in 2015 but became in 2016.*
Source: Data taken from INEI (2019a) Modulo-500 dataset

avoids children-only households. As with module 200, only the urban agglomeration size was used as a filter variable for datasets corresponding to 2015 (non-panel) and 2015-2016 (panel). After applying the filter, 23,197 individuals remained participants in the 2015 non-panel dataset, classified according to their age, gender, chief and income generator categories as shown in Figure 3.2.12. This dataset was also helpful in determining the characteristics of the household chief so that the distribution probabilities for this category could be calculated (Figure 3.2.9). Meanwhile, Panel data allowed characterising those earning or losing income-generating status according to their household chief status (Figures 3.2.10 and 3.2.11).

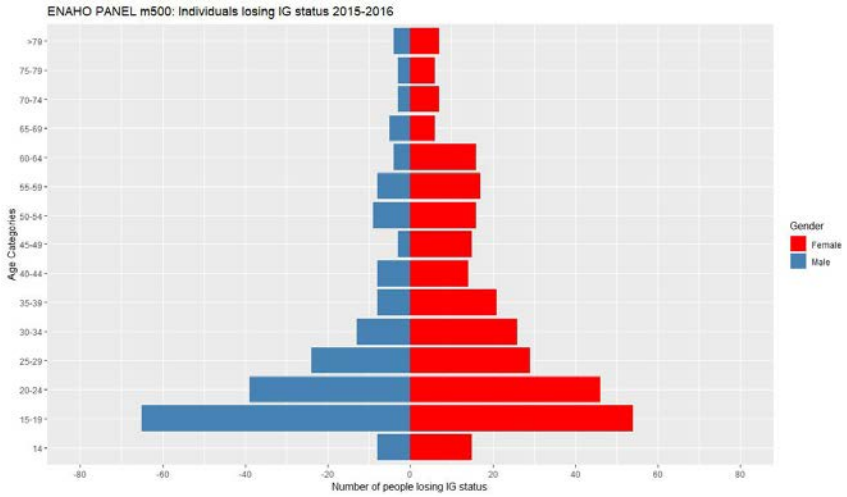


Figure 3.2.11: *Population pyramid of individuals who were income generators in 2015 but lost their status in 2016.*
Source: Data taken from INEI (2019a) Modulo-500 dataset

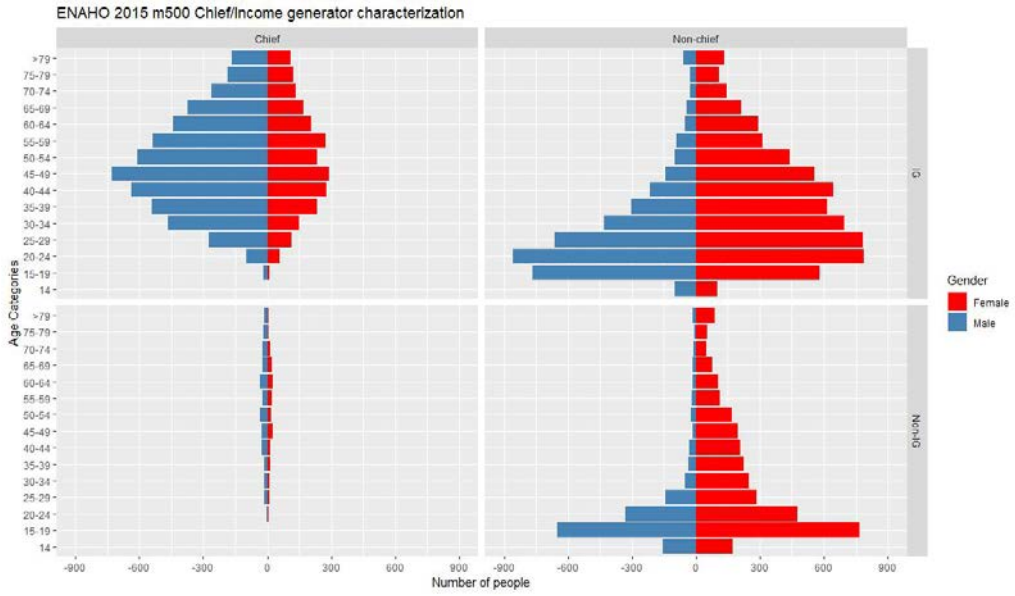


Figure 3.2.12: *Absolute number of filtered participants by age/gender category who are/are not income generators and/or household chiefs.*
Source: Data taken from INEI (2016b, 2017a, 2018a, 2019b, 2020a) Modulo-500 dataset

3.2.2 Other Datasets

Although complete, ENAHO misses some data needed to construct a full microsimulation demographic model. Deaths are recorded in panel data, but as these are self-reported, they could hide some cases, including deaths of individuals absent on the previous survey visit. Meanwhile, births could be detected by filtering new individuals in a two-consecutive-year panel household with less than a year of age, but this hides births between survey visits.

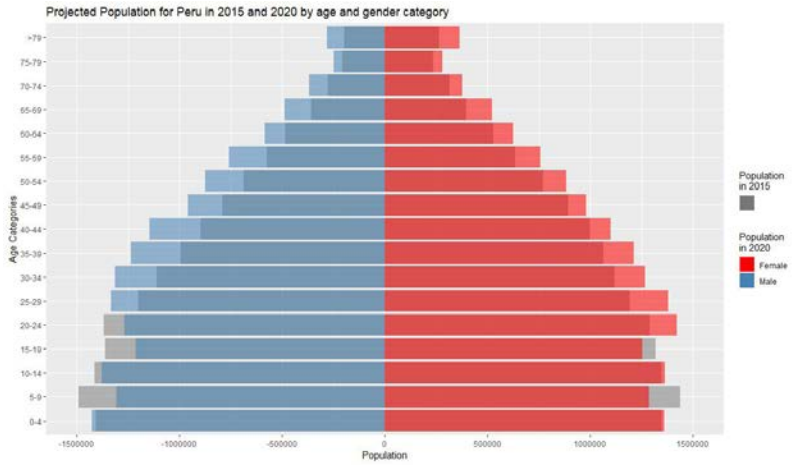


Figure 3.2.13: *Projected population for Peru in 2015 and 2020 by age and gender category*
Source: Data taken from INEI (2019d) dataset

As such, to calculate death probabilities by age and gender categories, this model relies on population estimates and projections for Peru by calendar year and simple age for the period between 1950 and 2050 (INEI, 2019d) and on the data released by the national informatics system for defunctions (MINSA, n.d.), which records all the registered deaths in the country starting from 2017.

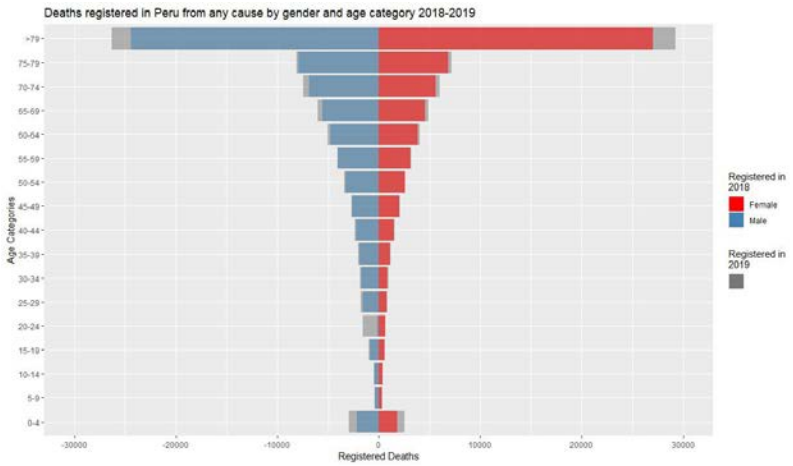


Figure 3.2.14: *Registered deaths in Peru in 2019 and 2018 by age and gender category*
Source: Data taken from MINSA (n.d.) dataset

Figure 3.2.13 shows the comparative population pyramids for the two extreme years considered from ENAHO's datasets (INEI 2016b, 2017a, 2018a, 2019b, 2020a). The model uses the registered deaths in those years to get approximate age and gender-specific death probabilities from population projections. By getting the number of deaths in year 1 (Y1) (Figure 3.2.14) and dividing it by the population projected by age categories for the previous year (Y0), we get an estimate of the share of the population that was alive on Y0 but died on Y1 (Figure 3.2.15). Given that COVID-19 considerably altered the deaths registered during the pandemic, we only account for valid death data between 2017 and 2019. This can be projected by age and gender categories and using simple extrapolation to cover all intended years of simulation.

Finally, to calculate the birth probabilities for women in their fertile age, the model relies on the national survey on demography and family health (ENDES). This dataset is based on a continuous survey performed by INEI, which specialises in health issues, particularly women and their sexual-reproductive aspect. Although ENDES' microdata is fully available to download, data presented in summary reports (INEI 2016c, 2017b, 2018b, 2019c, 2020b) is more than enough for general

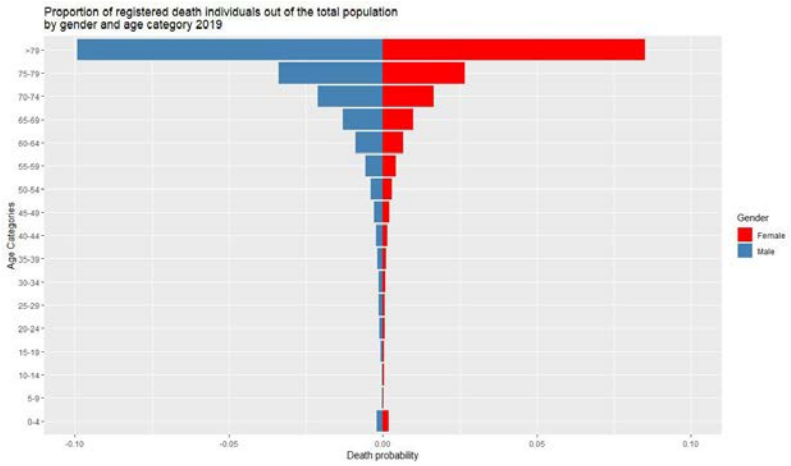


Figure 3.2.15: *Proportion of deaths over the total population by age and gender category 2019.*
Source: Data taken from INEI (2019d) and MINSA (n.d.) datasets

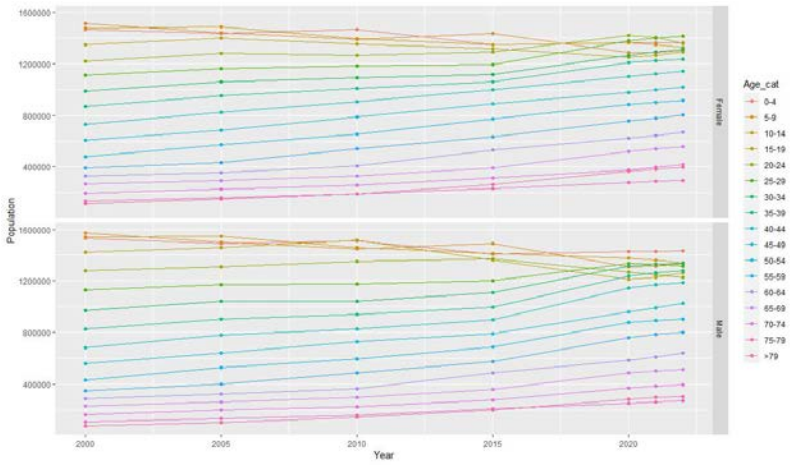


Figure 3.2.16: *Evolution of the population in Peru by age and gender category between 2000 and 2022.*
Source: Data taken from INEI (2019d)

fertility probabilities. Additionally, unlike ENAHO, ENDES does not account for a panel version. ENDES summaries provide useful figures, such as the number of children born alive by women's age category per year (Figure 3.2.17) and the time elapsed (in months) between the latest birth and the previous one according to the most recent child ordinal number. The earlier is helpful to determine distribution probabilities for the number of children by women according to their age category. Meanwhile, to get the number of children per woman, we can extrapolate the number of children by age category for the number of years the simulation needs (Figure 3.2.18).

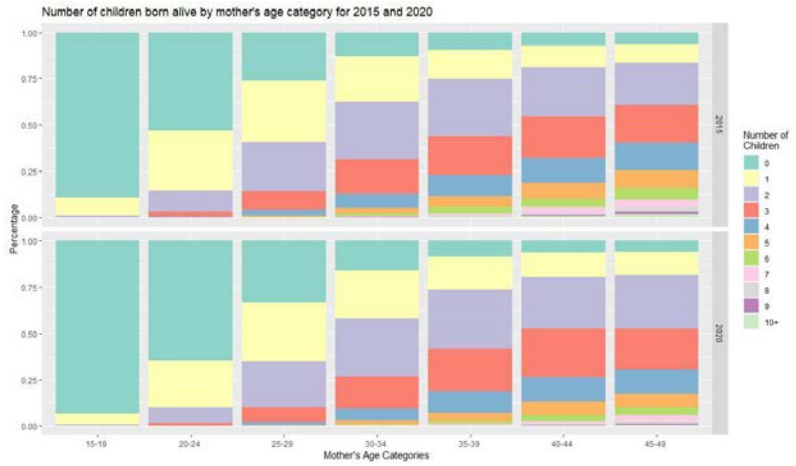


Figure 3.2.17: *Number of children born alive by women's age category for 2015 and 2020.*
Source: Data taken from INEI (2016c, 2017b, 2018b, 2019c, 2020b) dataset

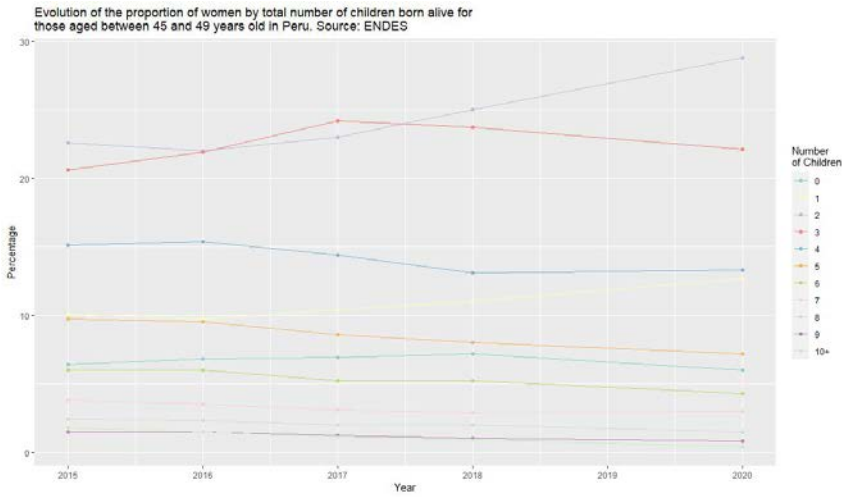


Figure 3.2.18: *Women between 45 and 49 years old in Peru. Evolution of the proportion according to number of children born alive.*
Source: Data taken from INEI (2016c, 2017b, 2018b, 2019c, 2020b) dataset

3.3 The model

Two core components form the proposed socio-economic model. The first one is the economic model, which predicts the categories of income and expenses per capita in relation to the mean of their SEL for every participating household, thus leading to determining the financial capacity of the household. The second component is the demographic sub-model. This has two outputs: the expected total number of household members for every year of simulation and the expected number of household members who are income generators for the same period. The earlier goes directly to feed the intermediary component of the general workflow, the latter

supports the economic model by adjusting the household income to the number of income-generating members.

3.3.1 Economic sub-model

To facilitate the execution of this model, the processed variables are:

- Per capita, as this is the most straightforward method to profit from immediately available aggregate figures at the household level.
- Relative to a mean by SEL, as this allows for alignment with an external complex economic model.
- And categorical, as this simplifies the generation of distribution and transition probabilities.

Inflation and GDP growth are some of the macroeconomic factors that have a direct impact on household income and expenses. However, it is challenging for micromodels to monitor these variables. Therefore, outsourcing to an external macro model is the most practical solution in this case. In the current setup, the macro model predicts income and expense means by SEL as continuous variables, whilst the micro model only deals with categorical variables relative to SEL means. As the latter are relative to the earlier, a communication channel exists allowing the transformation of categories to continuous values. To allow introducing the number of household members to the equation, and thus the impact of size on household finances, all variables are per capita, in the case of income relative to the number of income-generating members, and in the case of expenses relative to the total number of household members.

In summary, we can say that this economic model is not designed to predict future household income or expenses by itself but to predict likely household classification concerning mean income and expenses per capita values. Considering that macro global variables such as inflation and economic growth are tackled by an external model, this micro model only deals with processes internal to the household that affect their per capita income and expenses. As these are various, random, and uncontrollable, they can be considered stochastic and are managed by transition

probability matrices. We can also assume then that the membership of a household to one of the categories in question is a mere result of their present "luck", which could change randomly from one time step to the next.

The present model assumes five of these "luckiness" categories. These result from dividing the continuous values of the two variables (income and expenses per capita) among the means of their SEL in a given year. As such, a value of 1 in this new variable will indicate that the continuous datapoint for that household was the same as the mean for its SEL. A higher value will show that the continuous observation was above the mean by a factor, and a lower value indicates that it was below it by a fraction.

With these new values, we define the categories so that any observation with values between 1 and 1.5 can be said to be "around the mean" for their SEL. Two additional categories are set with similar value differences above and two below. After categorising all observations for the base year of the simulation (Table 3.2), it is possible to get distribution probabilities for income categories (Table 3.3). Because expenses are dependent on income, expense category distribution probabilities are calculated according to the belonging of the household to an income category (Figure 3.3.1). To get transition probabilities, the model utilises panel data (Figure 3.3.2).

Table 3.2: Number of cases for relative income and expenses categories.

SEL	Category	1 [0:0.5)	2 [0.5:1.0)	3 [1:1.5)	4 [1.5:2)	5 [2:Inf)
B	Income	67	134	56	32	35
	Expenses	36	148	66	43	32
C	Income	208	354	174	81	97
	Expenses	153	358	218	97	93
D	Income	817	1402	815	355	381
	Expenses	492	1520	966	419	385

Source: Based on data taken from INEI (2016b) SUMARIA dataset

Finally, a macro model of means per capita by SEL is needed to get continuous variables from the predicted categorical. To rationalise the macro modelling process, this exercise uses a simple extrapolation model, which takes the non-panel data for every one of the five years available (2015-2019), calculates the mean income and expenses per capita per SEL and projects it to 25 years using trend lines (Figure 3.3.3) Although this model oversimplifies the complex macro-economic system, it is enough to test the workflow presented in this thesis. If required, it could be

Table 3.3: Distribution probabilities for relative income categories according to SEL.

Income Category	B	C	D
1	0.207	0.228	0.217
2	0.414	0.387	0.372
3	0.173	0.190	0.216
4	0.099	0.089	0.094
5	0.108	0.106	0.101

Source: Based on data taken from INEI (2016b) SUMARIA dataset

		Expenses Categories				
SEL B		1	2	3	4	5
Income Categories	1	0.313	0.582	0.090	0.015	0.000
	2	0.097	0.582	0.209	0.090	0.022
	3	0.000	0.375	0.304	0.232	0.089
	4	0.031	0.281	0.219	0.250	0.219
	5	0.000	0.029	0.229	0.257	0.486
SEL C						
Income Categories	1	0.385	0.514	0.087	0.005	0.010
	2	0.172	0.494	0.229	0.071	0.034
	3	0.046	0.270	0.379	0.201	0.103
	4	0.025	0.185	0.407	0.210	0.173
	5	0.000	0.134	0.186	0.196	0.485
SEL D						
Income Categories	1	0.349	0.499	0.130	0.015	0.007
	2	0.109	0.512	0.272	0.080	0.027
	3	0.045	0.328	0.374	0.153	0.099
	4	0.014	0.234	0.299	0.248	0.206
	5	0.008	0.115	0.176	0.215	0.486

Figure 3.3.1: *Distribution probabilities for expenses relative categories according to SEL and close income category.*
Source: Based on data taken from INEI (2016b) SUMARIA dataset

eventually replaced by a more accurate and complex model.

The main drawback of the proposed modelling approach is the lack of data to simulate transitions between SEL categories. This is because SEL categories are static for households in the base panel dataset. This could be because the SEL categorisation was copied for every year at the end of the summarising process, or it could be that the survey did not find any change in the living conditions of the households during the years of study.

		Income categories at Tn+1				
SEL B		1	2	3	4	5
Income categories at TN	1	0.375	0.625	0.000	0.000	0.000
	2	0.234	0.553	0.191	0.021	0.000
	3	0.067	0.333	0.400	0.067	0.133
	4	0.000	0.364	0.273	0.182	0.182
	5	0.000	0.000	0.250	0.125	0.625
SEL C						
Income categories at TN	1	0.481	0.407	0.093	0.019	0.000
	2	0.163	0.450	0.250	0.113	0.025
	3	0.053	0.211	0.447	0.105	0.184
	4	0.000	0.160	0.400	0.280	0.160
	5	0.000	0.037	0.148	0.148	0.667
SEL D						
Income categories at TN	1	0.549	0.353	0.082	0.004	0.012
	2	0.208	0.506	0.210	0.052	0.024
	3	0.069	0.417	0.291	0.113	0.109
	4	0.017	0.224	0.336	0.198	0.224
	5	0.010	0.121	0.131	0.101	0.616

Figure 3.3.2: *Transitions probabilities for relative income categories.*
Source: Based on data taken from INEI (2019a) SUMARIA dataset

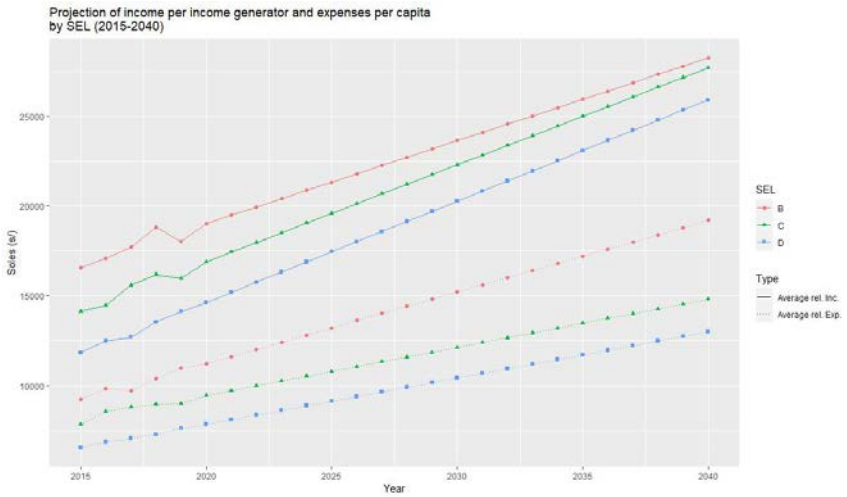


Figure 3.3.3: *Projection of relative income and expenses.*
Source: Based on data taken from INEI (2016b, 2017a, 2018a, 2019b, 2020a) SUMARIA datasets

3.3.2 Demographic sub-model

This sub-model has two outputs: the expected total number of household members for every year of simulation and the expected number of household members who are income generators for the same period. The earlier serves to, on the one hand, calculate an occupancy index (which eventually leads to a "need" variable triggering dwelling expansion) and, on the other, support the economic model so that it can output household annual expenses per household member as continuous values. The

latter only supports the economic sub-model, but in this case, with the household's annual income.

To generate these outputs, this sub-model relies on the basic variables determining demographic evolution: births, deaths, and migration. Additionally, it includes a variable that registers the income-generating status of every individual every year of the simulation. It relies on a stochastic approach to deal with demographic events, so every event is associated with a probability matrix.

To allow further specificity, probabilities are differentiated according to population characteristics. Birth probabilities, for example, are affected by the mother's age and the number of children she has had in her fertile lifetime. Death probabilities, meanwhile, are determined by the age and gender of the individual in question, factors that are also determinants in assigning emigration probabilities. In the case of assigning an income generator status, more specificity is needed, as beyond age and gender, being a household chief affects the likelihood of one being an income generator. Assigning a household chief also allows us to avoid having children-only households, as every household has one chief who must be an adult.

In this demographic model, every individual starts with values for each one of the following variables:

- Individual ID
- Dwelling ID
- Dwelling SEL category
- Household ID
- Chief role
- Discrete age
- Categorical age
- Gender
- Number of children born alive
- Income generator status

In the first time step, values for each variable are assigned or assumed. In the case of the number of children born alive, only women in their fertile age get a number above 0. Meanwhile, only individuals above 14 can be income generators, and only those above 19 can be household chiefs.

Every year of the simulation one year is added to individual discrete ages. Consequently, the categorical age is immediately re-evaluated. The two first demographic events simulated are the ones that have to do with losing household members: deaths (Table 3.4) and emigration (Table 3.5). These are the simplest, as they happen only once in the life of the simulated individuals and are determined solely by their age and gender categories. Once an individual is assumed to be dead or has moved out, they won't participate anymore in future time steps of the simulation and will not count towards simulation outputs. While death probabilities are taken from population projections and death registries, move-out probabilities come from ENAHO panel data for 2015 and 2016. Therefore, they represent the static conditions for inter-household mobility during that interannual period.

After determining the individuals departing from the dataset, it is time to assess the survival of whole households. For this, it is essential to bear in mind that every household must have one chief who is over 19 years old. We first check each household to see if they have remaining individuals; those who do not are eliminated from the dataset. Those who do are checked again to determine if their chief remains. If no chief is present, the next eldest adult takes the post; if no adult is present, the household is removed from the dataset.

The third demographic event simulated is birthing. Every woman has a probability of adding one child per time step to their record according to their age category and the number of children she has on record. The model does not register kin relations; thus, the number of children at T0 is just an abstract figure to differentiate probabilities of adding newborn members to a household and does not imply that the children a woman registers before the first simulated birth are household members.

For the number of children per woman by age category, the model uses ENDES, which does not register transition probabilities, only distribution ones. Because of this, the model relies on aggregate probabilities, which imply that the probability of

Table 3.4: Death Probabilities by Age and gender categories.

Age Categories	Male	Female
0-4	0.002	0.002
05-9	0.000	0.000
10-14	0.000	0.000
15-19	0.001	0.000
20-24	0.001	0.000
25-29	0.001	0.001
30-34	0.001	0.001
35-39	0.002	0.001
40-44	0.002	0.001
45-49	0.003	0.002
50-54	0.004	0.003
55-59	0.006	0.004
60-64	0.009	0.007
65-69	0.013	0.010
70-74	0.021	0.017
75-79	0.034	0.027
79+	0.099	0.085

Source: Based on data taken from INEI (2019d) and MINSA (n.d.) datasets

Table 3.5: Probabilities of leaving the household by age and gender categories.

Age Categories	Male	Female
0-4	0.118	0.134
5-9	0.092	0.084
10-14	0.064	0.091
15-19	0.075	0.141
20-24	0.156	0.143
25-29	0.188	0.176
30-34	0.131	0.146
35-39	0.089	0.051
40-44	0.066	0.037
45-49	0.053	0.049
50-54	0.056	0.018
55-59	0.040	0.027
60-64	0.018	0.025
65-69	0.020	0.032
70-74	0.051	0.067
75-79	0.037	0.022
79+	0.012	0.052

Source: Based on data taken from INEI (2019a) Modulo-200 dataset.

a woman giving birth to an additional child in the current year is equal to the sum of probabilities of having more children than she currently has. The probabilities of remaining at her current state equal the difference between the probabilities of adding one child and the unit. With this assumption, a transition probability table can be derived from the original data. To track the broader societal fertility trends, five years of ENDES data were used to extrapolate probabilities until 2040 (Figure 3.3.4). This data makes it possible to derive probabilities of adding one extra child to the count according to the mother's age (Figure 3.3.5). Once the model determines a woman has an additional child, a new synthetic member is added to the household and dwelling with age 0 and equal probabilities of being male or female. As newborns, they cannot be chiefs, income generators or mothers.

After adding and subtracting members, it comes time to determine the new income generator status of the remaining adult members. Transition probabilities (Appendix 1) are based on ENAHO panel data for 2015 and 2016 and are specific to age, gender and role categories. Because of their origin, these probabilities are

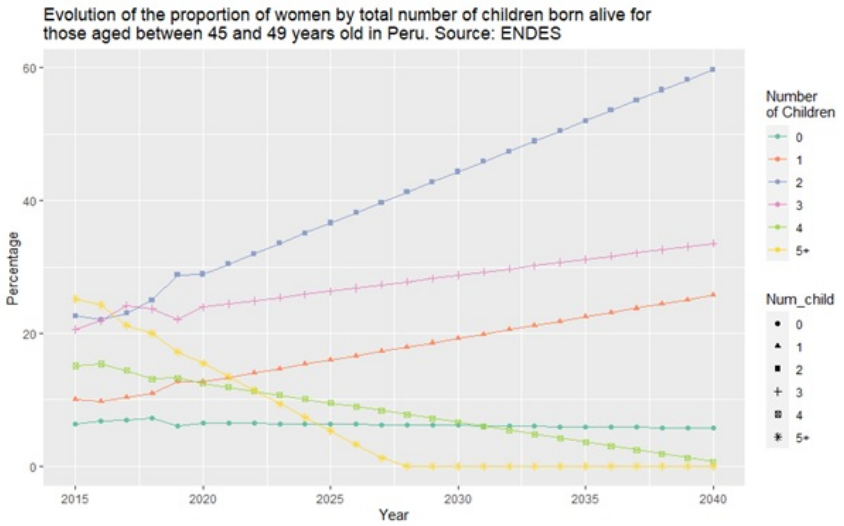


Figure 3.3.4: *Evolution of the proportion of women by total number of children born alive for those aged between 45 and 49 years old.*
Source: Based on data taken from INEI (2016c, 2017b, 2018b, 2019c, 2020b)

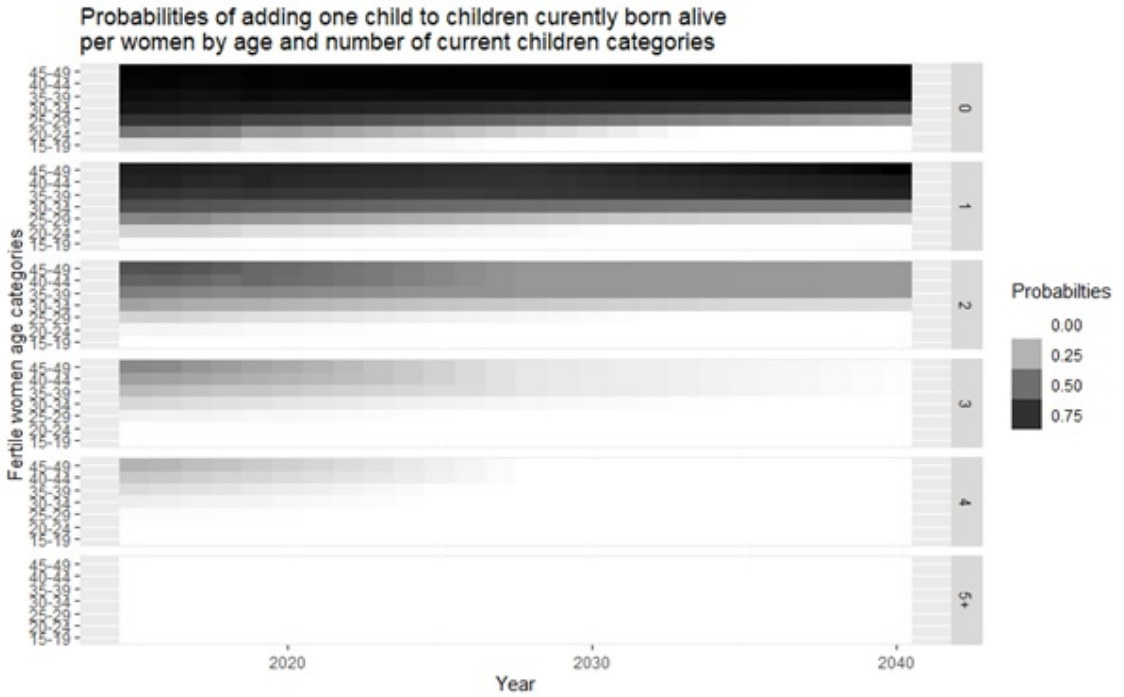


Figure 3.3.5: *Probabilities of adding one child to the current count by women's age category. Current count (number of children) as facets.*
Source: Based on data taken from INEI (2016c, 2017b, 2018b, 2019c, 2020b)

likely biased by the macro state of the economy in the country during those years. Nevertheless, it is hoped that the effect of this external factor is limited for the aims of the present simulation.

After extinguishing and adding individuals, new households must be created to replace those without members. To create a new household and its members, we

first set the number of members the new household will have according to distribution probabilities taken from ENAHO's summary module and specific to each SEL category (Table 3.6). This implies that the SEL is assigned to dwellings, not households, so an extinguished household is replaced only by a household of the same SEL category. This intends to represent the reliance of SEL classification on built environment variables and the relationship between property valuations and the income level of incumbent resident households. It also implies that the discrete number of household members in the base dataset is translated to categories, truncated at seven members per household due to low probabilities beyond that number.

After their quantity is determined, synthetic individuals are generated. First, we assign chief roles so that there is always one chief per household; then, according to distribution probabilities, gender (Table 3.7) and age categories (Appendix 2) are in that order. Finally, an initial income generator status is set according to age, gender, and role (Figure 3.3.6). Women are assigned a starting number of children according to the probabilities of the previously determined simulation year. The categorical age, which comes in 5-year cohorts used in most source datasets, determines the starting discrete age. This is simply a random integer between the limits of the category.

Table 3.6: Categorical distribution probabilities for the number of household members by SEL category.

	B	C	D
1	0.117	0.116	0.127
2	0.191	0.185	0.165
3	0.194	0.238	0.209
4	0.277	0.220	0.225
5	0.117	0.135	0.138
6	0.074	0.052	0.071
7+	0.031	0.053	0.066

Source: Based on data taken from INEI (2016b) SUMARIA dataset

Table 3.7: Categorical distribution probabilities for Gender according to Chief role category

	Chief	Non Chief
Male	0.687	0.413
Female	0.313	0.587

Source: Based on data taken from INEI (2016b) Modulo-200 dataset

Once new household members are generated, these are added to the primary dataset, and the simulation for a single year is done. As this is repeated for each year required, we get the needed outputs. We get the number of household members and income-generating household members for each dwelling and household identification

		Chief		Non-chief	
Gender		Male	Female	Male	Female
Age Category	0-4	0.000	0.000	0.000	0.000
	5-9	0.000	0.000	0.000	0.000
	10-14	0.000	0.000	0.391	0.363
	15-19	1.000	0.909	0.540	0.430
	20-24	0.990	0.934	0.723	0.623
	25-29	0.952	0.926	0.824	0.732
	30-34	0.969	0.949	0.892	0.736
	35-39	0.977	0.951	0.899	0.733
	40-44	0.962	0.952	0.870	0.757
	45-49	0.963	0.924	0.894	0.738
	50-54	0.946	0.936	0.821	0.725
	55-59	0.957	0.932	0.827	0.734
	60-64	0.927	0.896	0.781	0.737
	65-69	0.940	0.885	0.729	0.740
	70-74	0.926	0.911	0.692	0.745
	75-79	0.902	0.952	0.743	0.677
	>79	0.923	0.957	0.782	0.603

Figure 3.3.6: *Probabilities of being an income generator at T0 according to age, gender and chief status categories.*
Source: Based on data taken from INEI (2016b) Modulo-200 and Modulo-500 datasets

code for every simulation year. Once these are obtained, it is possible to return to the economic model outputs and, with the aid of the external macro model of projected means, get the households' income and expenses as continuous values.

3.4 Results

Table 3.8: SEL distribution for synthetic households at T0.

SEL	**Number of households**
B	10
C	14
D	24
Total	**48**

To test the model, a set of 48 synthetic households was generated using a customised distribution probability of SELs (Table 3.8), which aims to represent the imagined distribution of SELs within an incremental housing neighbourhood in an

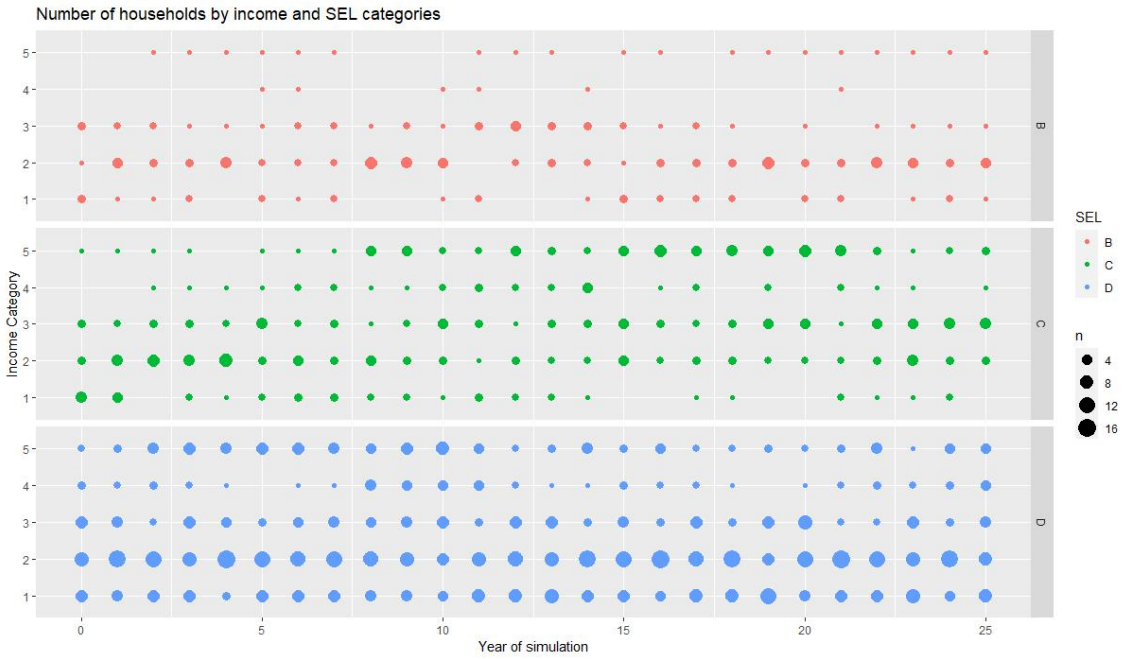


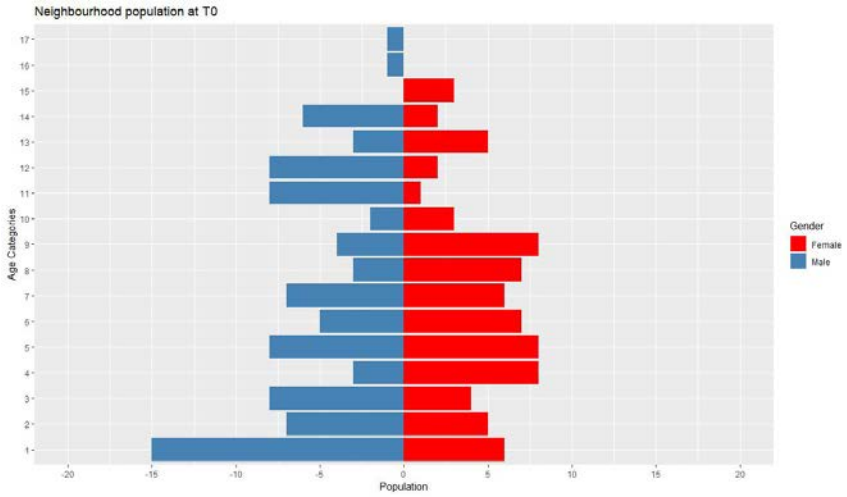
Figure 3.4.1: *Number of households by income and SEL categories every year of the simulation.*

intermediate city in Peru. With these SEL values, initial "luckiness" categories for relative per capita income and expenses were assigned and then projected for 25 years using the previously described economic model (Figure 3.4.1). As expected, the static transition probabilities generate similar distributions by SEL for all the years of simulation. Disaggregated results by household for this simulation can be seen in Appendices 3 and 4.

After using ENAHO's distribution probabilities, an initial number of household members is assigned to each household. With these numbers, a total of 164 individuals are generated for T0. Figure 3.1(a) shows their age and gender characterisation. Once the demographic model is run for 25 years, we see how the population decays on every SEL, reaching a total of 90 inhabitants in year 25, with a stabilisation trend on levels C and D (Figure 3.4.3). Whilst the population ageing process is expected, there is an unexpected predominance of male inhabitants in the older cohorts in the last years of the simulation (Figure 3.1(b)). This might have to do with the emigration probabilities, which are less in the case of household chiefs, who tend to be older males.

The most important output of the demographic model, the number of members per household, also shows a decaying trend, as expected in an ageing society (Figure

(a) Age and gender characterisation for synthetic individuals at simulation T0



(b) Age and gender characterisation for synthetic individuals at simulation T25

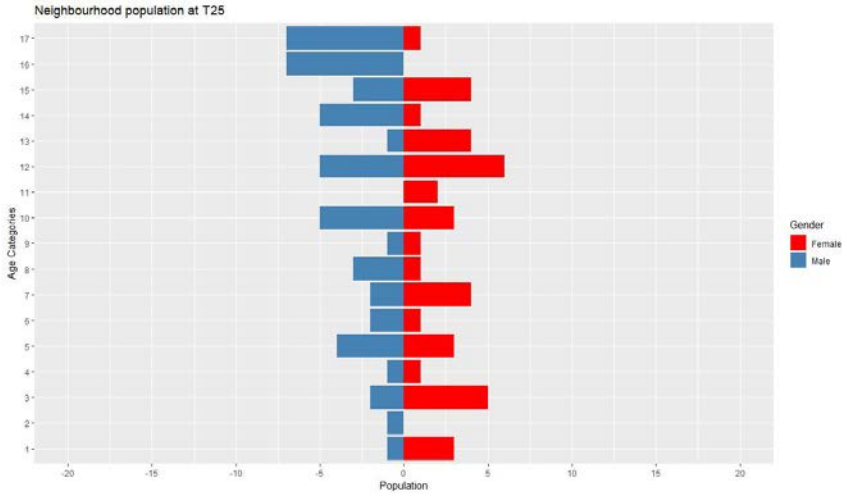


Figure 3.4.2: Population pyramids at T0 and T25 of the demographic simulation

3.4.4). Nevertheless, wave patterns emerge, likely indicating generational changes or birth waves. The other relevant outcome from this model, the number of income generators, is better understood when shown relative to the total number of household members (Figure 3.4.5). As expected, diminishing the household sizes results in a higher ratio of income generators out of the total number of household members, which would determine higher household yearly financial balances.

Once the results of the demographic simulation were ready, it became possible to complete the economic model by transforming the predicted categorical variables to continuous ones by using the previously described trendline model of mean income and expenses by SEL. Figure 3.4.6 shows the evolution of average yearly savings

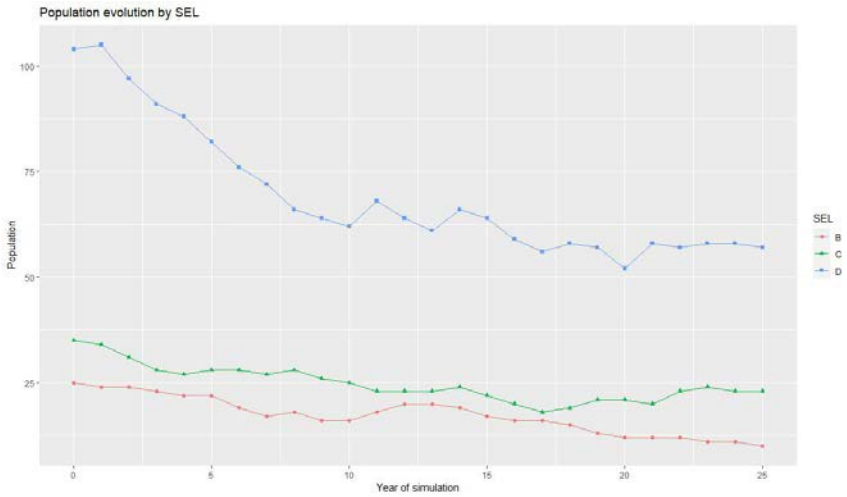


Figure 3.4.3: *Population evolution by dwelling's SEL.*

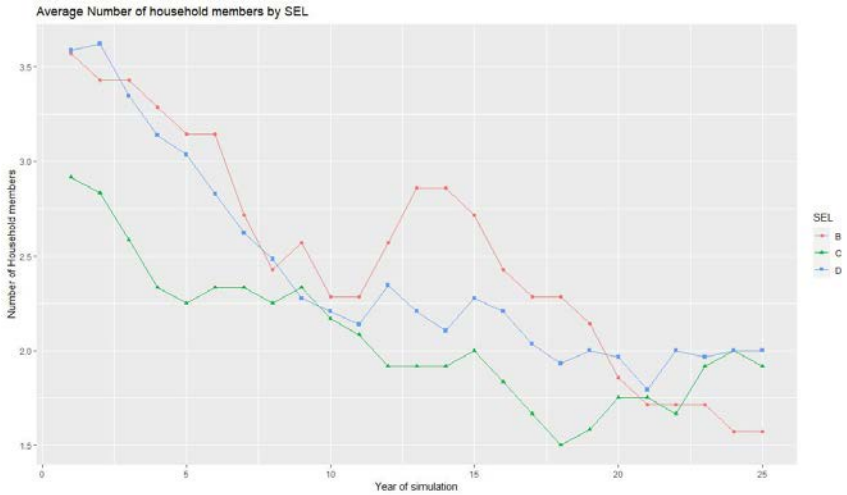


Figure 3.4.4: *Average number of household members by dwelling's SEL category*

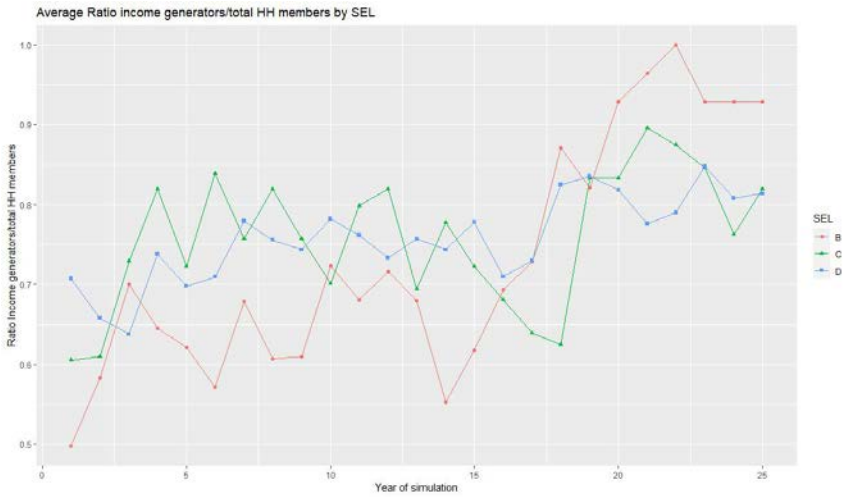


Figure 3.4.5: *Average ratio of income generators out of total household members by dwelling's SEL category.*

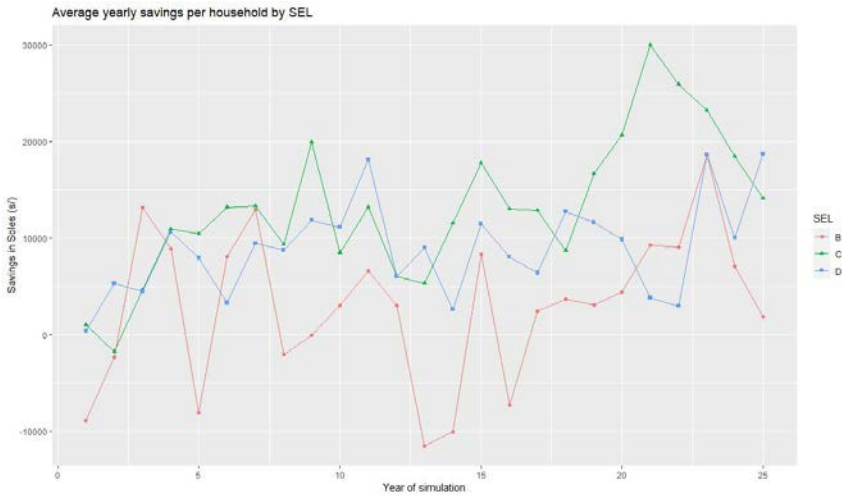


Figure 3.4.6: *Average yearly savings per Household by SEL.*

by dwelling's SEL. As expected, there is an upward trend in all SELs following the external model. The high variation between consecutive years of savings is something that draws attention. This effect is more pronounced in the SELs with fewer households, so it's likely the result of a few outlying data points affecting the classes' mean. The lower starting point for SEL B is also noticeable, probably related to its higher average number of non-working household members on the initial time steps.

3.5 Conclusions

The present chapter has dealt with the issue of proposing a plausible socio-economic model able to integrate with the general workflow of the thesis. It has presented micro-modelling as a widely used and highly developed method to model socio-economic phenomena using an agent-based approach. Consequently, it posed a micro-model consisting of two sub-models, one purely economic and another demographic. While the earlier predicts future income and outcome per capita categories, the latter predicts the total number of household members and income-generating household members. When working together, they can output the two desired variables to connect the socio-economic and the agent learning components. These variables are the number of household members and the yearly savings of the household. The plausibility of the proposed model lies in two factors. On the one hand, it follows a simple logic based on the basic events that drive demographic change

(births, deaths, migration). On the other, it uses real-world data to generate distribution and transition probabilities.

Although this basic plausibility is enough to feed the thesis workflow, some issues remain in the modelling approach chosen, most of which respond to data availability. First, it was not possible to include increasing life expectancy in the model (as it was done with decreasing fertility rates) just because publicly available death statistics in Peru only cover the period starting in 2017, including three years of abnormally high death rates due to the COVID-19 pandemic. Similar is the case for SEL transition probabilities, which were not possible to calculate as ENAHO panel summary data does not show differing SELs among years for participating households. Additionally, there is the case of evolving transition probabilities. These were managed in the economic sub-model by having relative categories dependent on an external model and in the birth rate by directly projecting the probabilities with a simple trendline model (as there were five years of available real-world data); nevertheless, the income-generating status, which depends on macro-economic growth, and emigration probabilities, which depends on broader societal changes, remained static for the simulation period. Although assuming purely static stochastic processes behind these two phenomena is plausible, an alignment procedure for these would increase the model's plausibility.

Concerning emigration probabilities, an odd but explainable phenomenon occurred in the test run: a predominance of men in the latter years of the simulation. Although such difference in gender distribution is unlike in a more comprehensive social system, it is explainable in a small closed system such as our neighbourhood, where this distribution could indicate the presence of household chiefs staying after other members have migrated. The lack of an immigration mechanism in the model could have magnified this effect. Such an approach requires assigning households an attractiveness factor that would add one or more members born outside the simulation. One of the candidate variables to calculate this factor would be annual household savings, assuming that new members would move in only if there is an excess in economic resources. Another approach is to include the relationship status of the inhabitants. Thus, non-married or divorced individuals would attract new members of a similar age. These were discarded as the earlier required an

infinite loop, in which the continuous results from the economic sub-model would have to feed the demographic sub-model, which feeds the continuous outcome from the economic sub-model. Meanwhile, the latter overcomplicates a model that has consistently avoided considering kin relationships due to its base dataset's format.

Finally, an interesting mechanism that could be included in the future is the capacity of the entire household to move out and be replaced by another. This would require a base dataset that includes this information to calculate event probabilities. Considering those changes, it should be possible to make the jump from a plausible model to a statistically accurate one. In such a scenario, a proper statistical validation procedure should take place.

References

APEIM. (2020, 10). *Niveles socioeconómicos 2020.* Retrieved from https://apeim.com.pe/wp-content/uploads/2022/08/APEIM-NSE-2020.pdf

Fondo MIVIVIENDA S.A. (2022). *Memoria institucional 2022.* Retrieved from https://www.mivivienda.com.pe/PORTALCMS/archivos/documentos/8585170892055792261.PDF

INEI. (2016a, 4). *Encuesta nacional de hogares 2016. metodologia para la obtencion de variables calculadas (sumaria).* Instituto Nacional de Estadistica e Informatica (INEI).

INEI. (2016b). *Encuesta nacional de hogares sobre condiciones de vida y pobreza - 2015 (enaho).* Instituto Nacional de Estadistica e Informatica. Retrieved from https://proyectos.inei.gob.pe/iinei/srienaho/Consulta_por_Encuesta.asp ([Data Set])

INEI. (2016c, 5). *Peru. encuesta demografica y de salud familiar 2015. nacional y departamental.* Instituto Nacional de Estadistica e Informatica. Retrieved from https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1356/index.html

INEI. (2017a). *Encuesta nacional de hogares sobre condiciones de vida y pobreza - 2016 (enaho).* Instituto Nacional de Estadistica e Informatica. Retrieved from https://proyectos.inei.gob.pe/iinei/srienaho/Consulta_por_Encuesta.asp ([Data Set])

INEI. (2017b, 5). *Peru. encuesta demografica y de salud familiar 2016. nacional y regional.* Instituto Nacional de Estadistica e Informatica. Retrieved from https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1433/index.html

INEI. (2018a). *Encuesta nacional de hogares sobre condiciones de vida y pobreza - 2017 (enaho).* Instituto Nacional de Estadistica e Informatica. Retrieved from https://proyectos.inei.gob.pe/iinei/srienaho/Consulta_por_Encuesta.asp ([Data Set])

INEI. (2018b, 5). *Peru. encuesta demografica y de salud familiar - endes 2017. nacional y departamental.* Instituto Nacional de Estadistica e Informatica. Retrieved from https://www.inei.gob.pe/media/MenuRecursivo/

publicaciones_digitales/Est/Lib1525/index.html

INEI. (2019a). *Enaho panel 2015-2019.* Instituto Nacional de Estadistica e Informatica. Retrieved from https://proyectos.inei.gob.pe/iinei/srienaho/Consulta_por_Encuesta.asp ([Data Set])

INEI. (2019b). *Encuesta nacional de hogares sobre condiciones de vida y pobreza - 2018 (enaho).* Instituto Nacional de Estadistica e Informatica. Retrieved from https://proyectos.inei.gob.pe/iinei/srienaho/Consulta_por_Encuesta.asp ([Data Set])

INEI. (2019c, 5). *Peru. encuesta demografica y de salud familiar - endes 2018.* Instituto Nacional de Estadistica e Informatica. Retrieved from https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1656/index1.html

INEI. (2019d, 7). *Peru: Estimaciones y proyecciones de la poblacion nacional, por año calendario y edad simple, 1950-2050. boletin especial no 24.* Retrieved from https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1681/

INEI. (2020a). *Encuesta nacional de hogares sobre condiciones de vida y pobreza - 2019 (enaho).* Instituto Nacional de Estadistica e Informatica. Retrieved from https://proyectos.inei.gob.pe/iinei/srienaho/Consulta_por_Encuesta.asp ([Data Set])

INEI. (2020b, 5). *Peru. encuesta demografica y de salud familiar - endes 2019.* Instituto Nacional de Estadistica e Informatica. Retrieved from https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Endes2019/

Instituto CUANTO. (2018). *Estudio de demanda de vivienda a nivel de las principales ciudades hogares no propietarios.* Retrieved from https://www.mivivienda.com.pe/PortalCMS/archivos/documentos/01.%20Estudio%20de%20Demanda%20de%20Vivienda%20Nueva%20de%20Arequipa.pdf

Li, J., O'Donoghue, C., & Dekkers, G. (2014, 10). Dynamic models. In *Handbook of microsimulation modelling* (p. 305-343). doi: 10.1108/S0573-855520140000293009

Mason, C. (2014, 10). Demographic models. In *Handbook of microsimulation modelling* (p. 345-365). doi: 10.1108/S0573-855520140000293010

MINSA. (n.d.). *Sistema informatico nacional de defunciones (sinadef).* Ministerio de Salud. Retrieved from https://www.minsa.gob.pe/reunis/data/defunciones_registradas.asp ([Data Set])

O'Donoghue, C. (2014, 10). Introduction. In *Handbook of microsimulation modelling* (p. 1-21). doi: 10.1108/S0573-855520140000293001

Scott, A. (2001). *A computing strategy for sage: 1. model options and constraints.* Simulating Social Policy in an Ageing Society (SAGE) London.

Chapter 4

Optimisation Environment

The current chapter deals with the optimisation environment of Part B in the methodology. As exposed in the first chapter, this part consists of an optimisation loop that needs an environment to allow the functioning of its Evaluator and Selector components. As such, this environment is formed by two parts: A geometry generation (GG) script that creates and modifies building envelope geometries to be evaluated and selected, and a development environment in which this script is implemented and that allows encompassing the other two components.

The present chapter then starts by introducing the requisites for the ideal development environment. It continues by describing the selected one, emphasising its compliance with the identified requisites. It then proceeds to describe the internal processes and functioning of the GG script, including the programming paradigm used for the development, the programming language, the required modules, the classes created and the interaction with the other components of the workflow. A detailed description of the procedural geometry generation method and its interaction with external applications that allow adding non-geometrical data to the outputs is also considered. Finally, the chapter presents conclusions and recommendations to implement the optimisation environment within a fully functional version of the general workflow.

4.1 The development environment

In software development, an environment is a workspace from which the software is written, tested and debugged. In our case, this software is the so-called optimisation loop, including its selector, evaluator and GG script components. Given the individual requisites of each of these and the required interaction between them and the rest of the workflow, a list of key requisites for the development environment was acknowledged:

1. The graphical nature of the geometry generation and manipulation process requires a Graphical User Interface (GUI), fully integrated at least during the development process of the GG script.

2. The development and integration of the selector component within the environment requires the ability to jump with ease from optimising one to tens of interacting agents.

3. There should be unambiguous communication between the three components of the optimisation loop and the other parts of the workflow. This requires available communication channels and minding the compatible formats accepted by the evaluator.

The last requisite sums up one of the main challenges presented in this chapter, which is communicating custom-written scripts with externally written software (the evaluator). In this sense, two factors will be tackled: communication channels and compatible formats. Regarding the latter, our selected evaluator, EnergyPlus (The National Renewable Energy Laboratory, 2017), accepts two text-based input file formats: IDF, recommended when the files are generated via a GUI, and epJSON, recommended when using a programming language (U.S. Department of Energy, 2021). Regarding communication channels, the optimisation phase requires auto-execution without the need for human intervention; as such, the standard approach is to rely on Application Programming Interfaces (API). The most straightforward way to control EnergyPlus using APIs is by using OpenSudio SDK, an open-source analysis platform and toolkit that facilitates integrated whole-building energy analysis and provides an API to access EnergyPlus (Guglielmetti, Macumber, & Long, 2011).

Altought this arrangement seems to comply with requisite number three, further adaptations are needed to comply with the two remainings. Requisite one, for example, demands a GUI compatible with the environment, at least during the GG script development phase. For this task, OpenStudio relies on SketchUp (Trimble Inc., 2000) via a plug-in. Whilst SketchUp allows scripting, this mode of using the software is not particularly popular and, therefore, lacks community support. Luckily, as OpenStudio is open-source, many independent initiatives profit from its API and Command-line interface (CLI) capabilities, thus expanding the graphical platforms able to interact with EnergyPlus. The most popular of these initiatives is Ladybug Tools (M. S. Roudsari & Mackey, 2017), a collection of computer applications that support environmental design and education. It is open-source, community-supported, integrated with the most currently popular 3D software and written in Python, the most popular programming language nowadays (TIOBE Software BV, 2023). Although, thanks to its current cross-platform capabilities, LadybugTools can be plugged into most existing geometry engines, it was initially developed as a Grasshopper plug-in, thus accounting for more experience, support and legacy options on that platform.

Grasshopper (McNeel and Associates, 2014) is a visual programming language and environment that runs within Rhino3D (Rhino), a commercial computer graphics and computer-aided design (CAD) software. Rhino's geometries are based on the NURBS mathematical model, focused on producing a mathematically precise representation of curves and freeform surfaces. This characteristic differentiates Rhino from most CAD applications and has made it popular within the procedural and parametric geometry generation communities. Being NURBS complex geometrical objects, designers usually rely on scripting rather than on Rhino's GUI to achieve accuracy. Despite this, being design a highly graphical exercise, users remain dependent on graphical outputs to keep a visual on their tasks. Grasshopper plays a significant role in covering this form of human-computer interaction. As a visual programming language, it offers a more relatable experience to designers who do not have "traditional" scripting training, whilst it uses Rhino's graphical output to show the resulting geometries in real-time. This also opens the doors to the straightforward practice of parametric design, which means generating geometries

within parameters rather than with absolute properties or inputs. Parametric design leads to the use of optimisation methods for geometry generation. As such, both the platform and the community are used to the integration of these in their workflows.

The Grasshopper development environment consists of a canvas, or working space, and several components with varying functions that can be placed on it. These components are equivalent to what we call functions in a traditional programming language. They accept inputs, execute an internal process and produce one or several outputs. Grasshopper includes a customisable component so that one can write a script within it (using C, Python or VisualBasic) and create a new process to be executed. This has allowed the surge of a constellation of community-generated components that can perform various very specific functions. LadybugTools, written in Python, started as part of this constellation (M. S. Roudsari, Pak, & Smith, 2013).

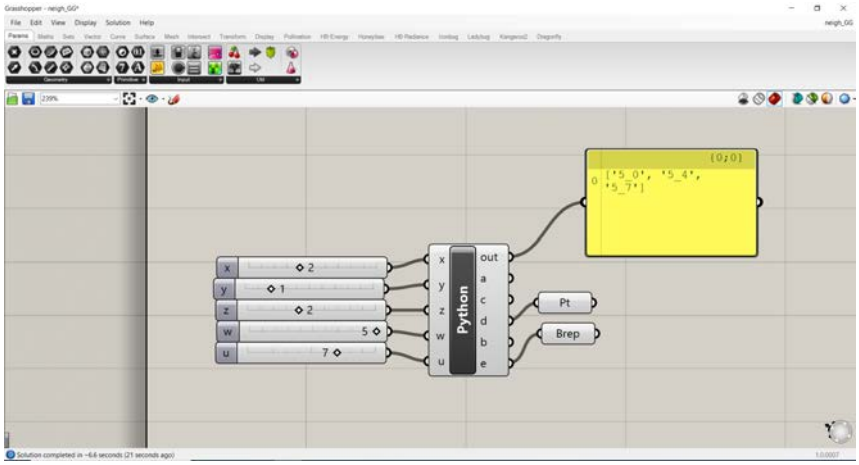


Figure 4.1.1: *Grasshopper Canvas and one Python component with its inputs and outputs*

As such, using Grasshopper as an environment, Rhino3D as GUI and OpenStudio (via Ladybug Tools) as API medium for EnergyPlus, theoretically allows complying with requirements 1 and 3. This setup requires writing the GG and the selector scripts as custom Grasshopper components, and given that Python is the native language of Ladybug Tools, it makes sense that these two use it as well. This setup was successfully used during the development phase of the selector component (shown in Chapter 5). Nevertheless, it showed shortcomings when scaling up the learning process to tens of agents, failing to comply with requisite 2. Two factors contributed to this limitation:

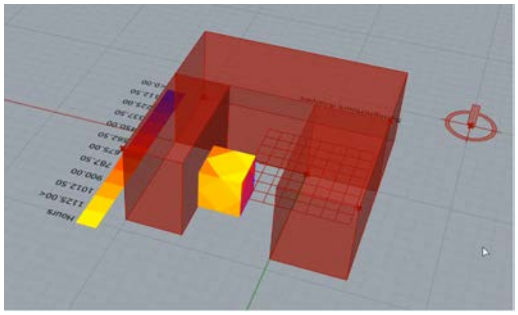
- First, in the Grasshopper canvas, an Evaluator component is needed for every agent participating in a learning process. This becomes very inconvenient and graphically confusing when dealing with tens of agents.

- Second, Grasshopper components are incapable of outputting intermediate states when commanded to execute a loop. In incremental housing, the intermediate states are precisely crucial.

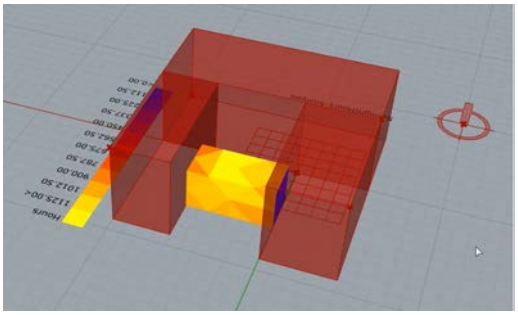
During the development phase, only up to four agents were used, and their performance was aggregated for evaluation, so the first shortcoming was completely avoided. The second one had a temporary workaround: Loop processing was taken outside of the component and into the Grasshopper canvas. This was achieved by having an external counter that computed the current iteration's ordinal so that Grasshopper components deal only with the tasks assigned to one iteration at a time. To update the component's output, the resulting solutions must expire. This requires a signal so that the canvas counter knows when to carry on with its task. To achieve that, it is possible to profit from "*sticky*", Grasshopper's default data collector that stores variables beyond solutions expiration. By writing and reading values from this collector, the counter and the components keep a communication channel. It is also possible to send data via "sticky" from the Evaluator to the Selector, so they coordinate their tasks. By setting geometries as outputs, this environment allows witnessing the learning process directly, which also allows a debug process on both the GG script and the Selector implementation.

Although the previously described environment was enough to achieve the results shown in Chapter 5, the thesis objectives demanded further interacting agents. As such, a change of environment became necessary. This involved completely detaching the script from the Grasshopper canvas and running it from an external development environment. This was possible as, in the previous setup, the GG script was never dependent on Grasshopper; it was just using it as a development environment to communicate with *EnergyPlus* via the *LadybugTools* components while profiting from its interactions with Rhino to output real-time graphics. Its dependencies to allow the latter and generate geometries compatible with *LadybugTools*, were limited to *Rhino.Geometry*, the Geometry namespace of *RhinoCommon*, the Python Software Development Kit (SDK) for Rhino (McNeel R. and Associates, 2023b).

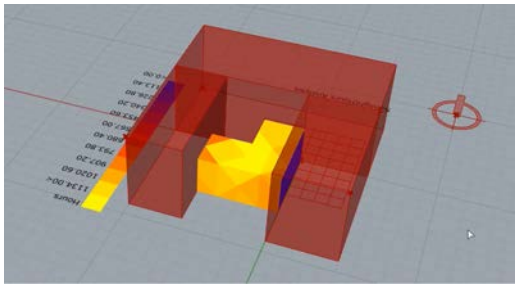
(a) Seed state



(b) State 1



(c) State 2



(d) State 3

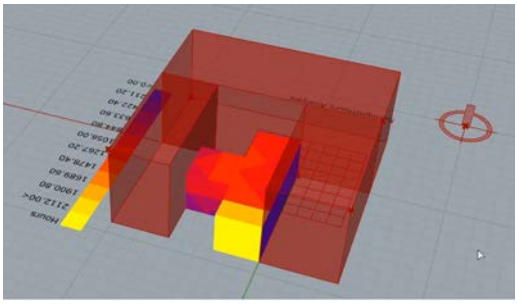


Figure 4.1.2: Frames of the graphical output of a single-agent optimisation process using Ladybug as Evaluator and Reinforcement Learning as Selector

As such, a straightforward change of environment can happen by using the *rhino3dm* module, a geometry library with a *RhinoCommon* style able to access and manipulate geometries independent of Rhino (McNeel R. and Associates, 2023a), or the *Rhino.Inside* module, which allows running Rhino inside many other programs, including those written in Python (McNeel R. and Associates, 2023c). With these alternatives, it is possible to execute the GG script developed for the previous setup outside Grasshopper. From the side of the Evaluator, meanwhile, although LadybugTools components are natively written in Python, a transformation from Grasshopper components to Python functions is still required. This is a relatively straightforward task which implies adapting the component contents to a pure Python script.

The disadvantage of this approach was that the live geometric outputs of the learning process were lost. Nevertheless, as the GG script was already developed to the point of deployment, the objective could still be attained. In exchange of this loss it was possible to overcome the two challenges previously presented:

- The intermediate states of loops became accessible for output and manipulation, including the possibility of sending intermediate-state geometries to the

Evaluator when required.

- The Evaluator could be called within Agent-based loops so it can execute energy simulations for every agent participating in a learning process.

An additional advantage of this setup is that it becomes easier to integrate High-performance computing (HPC) and cloud computing resources into the environment, thus potentially speeding the processing time of simulations.

To facilitate the portability of the GG script, *Rhino.Inside* was selected over *rhino3dm*, as the latter comes *in the style* of *RhinoCommon*, whilst the earlier literally calls *RhinoCommon* functions. The difference between the two is minimal, but it does exist so that the selected workflow avoids a second debugging process. This comes at a cost, as *Rhino.Inside* requires an installed and licensed version of Rhino to work, and as Windows and Mac are the only compatible platforms for the software, the execution of our GG script is also limited to them. This discards the option of executing the code fully on available HPC premises (at The University of Sheffield, currently, all HPCs run on Linux). Luckily, the mother company of LadybugTools offers *Pollination* (Ladybug Tools LLC, 2023), a cloud computing service that allows the processing of several energy simulations on the cloud in parallel. By using its API, it is possible to interact with it from a local computer and execute energy simulations on request, thus expediting the workflow execution time when working with tens of agents.

In summary, the current section has described two development environments for the optimisation process: One working within the Grasshopper canvas and another working in a pure Python environment. These two environments are used asynchronously, according to the requirements of a phased development process. While the earlier serves well to get the GG and Evaluator scripts to a deployment stage, the latter allows scaling up the workflow to handle tens of agents. In both environments, thanks to the presence of LadybugTools, either as Grasshopper components or Python functions, the resulting geometries from the GG script can be translated to a compatible format for EnergyPlus.

4.2 The Geometry Generation script

The following section describes the GG script in its entirety. This script is designed to receive and interpret the orders given by the Selector, transform them into geometrical data and pack these geometrical results, along with complementary non-geometrical data, in a compatible format readable by the Evaluator. This is done in two consecutive steps: the geometry generation itself and the addition of supplementary data. To achieve the first of these, the script is self-reliant, while for the second, it relies on functions from LadybugTools modules.

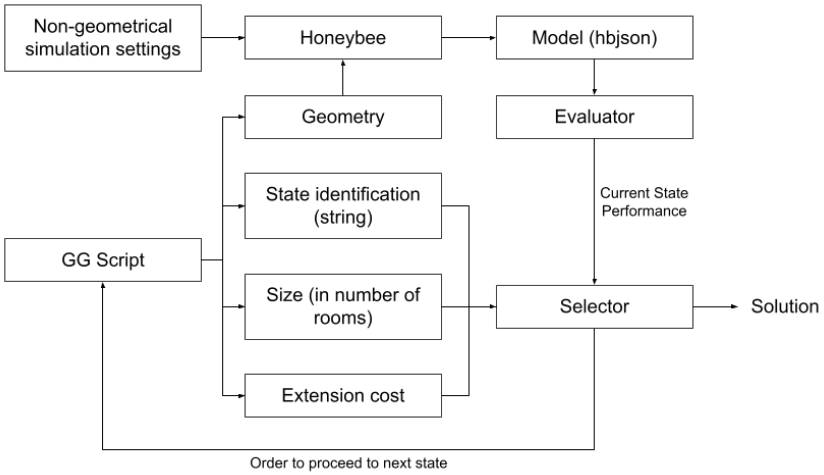


Figure 4.2.1: *Detailed optimisation loop, including the interactions with the GG script.*
This loop is triggered by the incremental development model shown in 1.1.1, which informs on the "need" and financial capacity of the agent to the *Selector*. In such a way, the feedback to the GG script becomes bounded in quantity and timing.

4.2.1 The geometry generation script structure

The geometry generation script relies on the Object Oriented Programming (OOP) paradigm, as this allows the creation of complex programmes by organising their design around objects able to interact with each other rather than relying on imperative instructions given by the programmer. Considering the methodological approach taken by this research, these objects can configure the agents at the core of our agent-based geometry optimisation problem. Objects in an OOP programme are instances of a class, and several classes might co-exist in the same programme. Objects also have attributes, which represent the state of an object, and methods,

which are the functions that the object can execute. Classes in a programme can be completely independent from one another, or they can have relationships. The GG script here presented is designed around the "*Agent*" class, from where geometrical objects representing dwellings are instances. The "*Agent*" class has a composition relationship with the "*Block*" class, as one block might contain one or more dwellings. This sort of arrangement allows a *Block* object to arrange the houses in order and to obtain their attributes and the possible interrelationships between them. It also enables it to simplify the application of methods to all its components and add new methods referring to their aggregate behaviour.

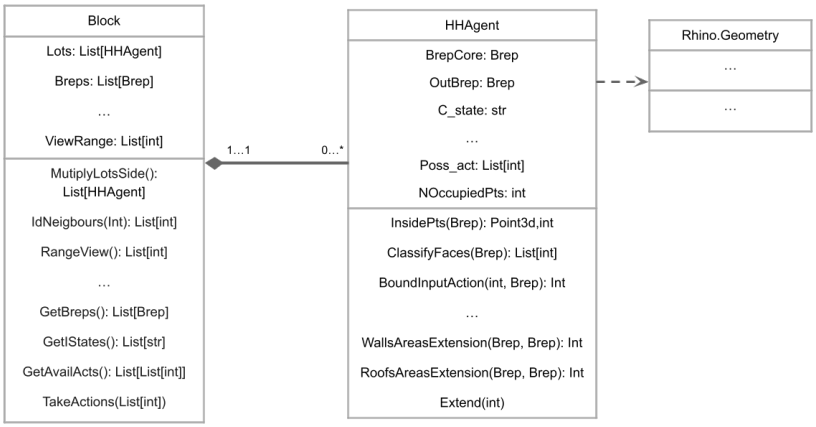


Figure 4.2.2: *UML class diagram with the most important attributes and methods of the Block and Agent classes*

In this OOP structure, geometries representing individual dwellings are attributes of the "*Agent*" class. To modify the geometrical structure and thus simulate the incremental process, we rely on the *Extend()* (Fig. 4.2.3) method of the said class. To generate and modify these geometries, the script uses objects belonging to classes native to the *Rhino.Geometry* module. As explained in the previous section, this allows portability between the development and the deployment stages. From this module, there is a short list of classes on which the GG script ultimately depends. These are:

- Points 3d: Represents the three coordinates of a point in three-dimensional space, using double-precision floating point values.
- Polylines: an ordered set of points connected by linear segments, which can be closed if the starting and ending points coincide.

- Vectors 3d: Represents the three components of a vector in three-dimensional space, using double-precision floating point numbers.

- Breps or Boundary Representations: Surfaces or polysurfaces with trim curve information.

- Meshes: a geometry type that is defined by vertices and faces.

The two latest are complex constructions with inner components (vertices, edges, faces, etc.) so that their attributes and methods can also refer to these objects. The geometry attribute of an "*Agent*" object is a *Brep* object, which represents the envelope of one incremental dwelling at a given point in time. As such, the script is structured around generating and manipulating these *Brep* objects. The agent's *Extend(int)* method achieves it using Rhino's *Extrusion* method, which creates an initial geometrical seed extruded from a *Polyline* footprint drawn inside a bounded lot and grows this seed by extruding a selected face from it. As a hollow shell representing the outer boundaries of the dwelling, all internal divisions are eliminated every time the expansion process is simulated. To make sure that the output remains a closed solid after dealing with the internal divisions, a *Brep-Mesh-Brep* transformation is used just before outputting the result as an *Agent* attribute. Incremental expansion is achieved when the resulting geometry is set as the initial state so that any further transformation is applied to the previous output.

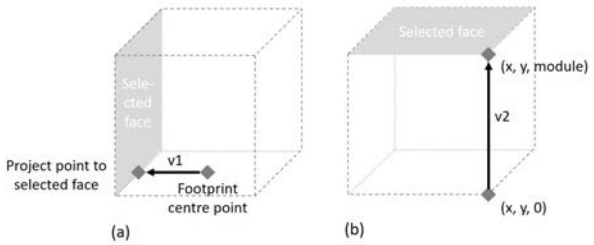


Figure 4.2.3: *Generation of vectors for extrusion as part of the Extend() function. To get its dimension, v1 is unitized and multiplied by the "module" input. V2 already shows its correct direction and dimension.*
Source: Poco-Aguilar, Wate, and Robinson (2022)

To place these objects in space, a coordinate system is needed. Rhino's coordinates use the unit system input by the user. While this facilitates relating a scaled model with its real-world inspiration, the GG script complements it with an additional modular unit system, which enables, on the one hand, the standardisation of dwelling sizes and, on the other, the communication with the rest of the components of the workflow. As such, dwellings are formed by modules that represent functional rooms. This allows it to understand the "need" input coming from the socio-economic model (which uses rooms as units of measure). Considering this, lots sizes and height construction limitations are given in modular units. To facilitate the process, modules are cubic and allow for mimicking a minimally functional residential room.

The agent class

The script is structured around the "*Agent*" class. It is, therefore, vital to understand its structure to understand the programme as a whole. An object belonging to the "*Agent*" class is initialised by inputting the following basic geometrical attributes:

- Location of the initial (lower left) point of the lot in Rhino's system of coordinates (X and Y),
- Size of the lot's basic module in Rhino's unit system,
- Size of the lot in X and Y in modular units,
- Location within the lots' modular grid of the initial point (lower left) of the footprint of the seed geometrical state of the dwelling.
- Size of the seed state of the dwelling in X and Y in modular units.
- Maximum height allowed for the vertical growth of the dwelling in modular units.

With this data, the script creates an "*Agent*" object whose seed geometric representation is obtained with the *BrepCore* attribute. To generate the seed geometry, the script first bounds the individual lot that limits in X and Y the growth of the

dwelling. For this, it takes the coordinates of the initial point of the lot, the equivalence of the modular dimension in Rhino's measurement unit system and the lot size in X and Y in modular units. With these ingredients, it locates the remaining three points, defining a quadrangle that represents the limits of a lot. Once these boundaries are determined, a grid of points is generated within the lot by considering its modular dimensions. A list of available coordinate points to start the seed geometry of the dwelling is created by taking out the points located on the upper and right boundaries.

This 3D geometry is achieved by obtaining the base points with a strategy similar to those used with the lot boundaries, then getting a closed polyline from the base points, and finally extruding a Brep surface obtained from this polyline. The extrusion demands a guiding line, which is obtained by connecting one of the base vertices with a projection of the same with a Z coordinate equal to a module's measure. The result is a Brep, either cubic or rectangular parallelepiped (depending on the base), whose height is always limited to one module. This seed geometry is permanently stored in the agents' *InitBrep* attribute.

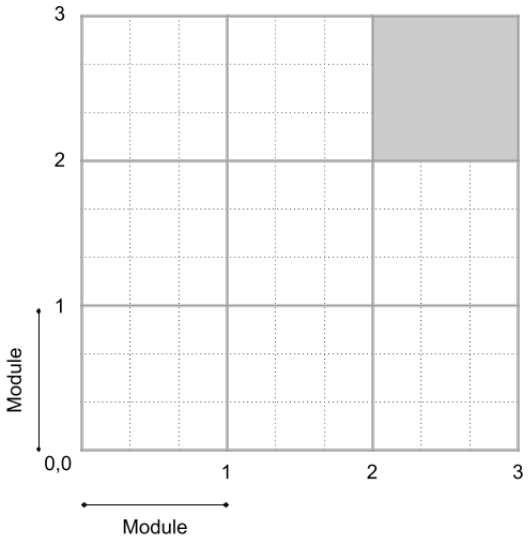


Figure 4.2.4: *Top view of the two measurement systems overlapping to locate initial geometrical states*
In this example figure, the GG script *Module* is equivalent to three system units (dotted grid). This generates a secondary modular grid (solid lines) which guides the location and development of building envelopes. Here, the initial building envelope state (grey fill) is the size of one room and is initialised in modular coordinates (2,2).

As the geometries represent an incremental dwelling, a vital method of the

"*Agent*" class is the *Expand(int)* function, which controls the application of the *Extrude* method from *Rhino.Geometry*. The *Expand(int)* function receives an integer as input, which represents an index in a list of face identifiers of the current geometrical state of the *Agent*. This list is filled with only the identifiers of those faces that are "extrudable". These faces either look to the sky (if their vertices have a Z coordinate below the height limitation) or not on the lot's boundary. These limitations allow keeping the dwelling's development within constraints in X, Y and Z axes. To discriminate growable faces from the rest, walls and roofs indices are first separated into different groups. This classification is done by obtaining the vertices forming the faces of the *Brep* and then getting the coordinates that they share. Vertices on the roof share Z coordinates, and depending on this shared value, we can determine if it is a growable roof. Wall vertices, meanwhile, share either an X or Y coordinate, and to be growable, this shared value should be different to the values of the lot's boundaries.

To allow any positive integer as input, the *BoundInputAction(int)* method converts any integer beyond limits to one within it by using the remainder of the division method. This function also subtracts a unit from the input when different to zero so that zero remains reserved to indicate nil extension. Negative integer inputs are allowed, but they are all turned to -1, indicating an order to return the seed geometrical representation and eliminate all the growth executed so far. As a result, and to achieve the progressive transformation, the *Extend(int)* method might alter the *OutBrep* attribute to a new geometry resulting from the transformation process or go back to the seed geometry, depending on the input. A complementary output of this method is the number of available actions after arriving at a given state. This results from measuring the length of the list with the IDs of the extrudable faces and is useful to construct an incremental Q-table, as explained in Chapter 5.

The extension of the dwelling is achieved by extruding the selected face in an outward direction perpendicular to its plane. To ensure that the extrusion is made in the correct direction when the chosen surface is a wall, the centre of the building's footprint is first determined using the footprint's vertices. The centre point is then projected to the ground projection of the wall. With that direction and the module as length, a vector can be traced from the centre point projection on the footprint

boundary so that the *ExtrudeFace* method from *Rhino.Geometry* has all it needs to perform its task. If the selected face is a roof, the process followed is similar to the one extruding the seed geometry in the Z direction. All internal divisions are deleted to simplify the energy modelling (considering that we are dealing with the optimisation of building envelopes only). So, after performing the extrusion, the original face is deleted, and only the outer boundaries of the geometry remain. To keep the geometry as a closed entity, a Mesh covering the outer boundaries of the geometry is generated; this new object is closed and then reconverted into a Brep. Before outputting the result, the method ensures that there are no duplicated faces and that we end up with a closed geometry with all their faces pointing outwards, as demanded by the Evaluator. Exceptionally, a non-manifold geometry will be generated. By definition, this is a non-closed geometry, but as its status does not depend on the lack of connection between the faces, there is nothing that the GG script can do to "geometrically solve" this output. Because of this, its management is left to the Selector, which penalises it when encountered (See Chapters 5 and 6).

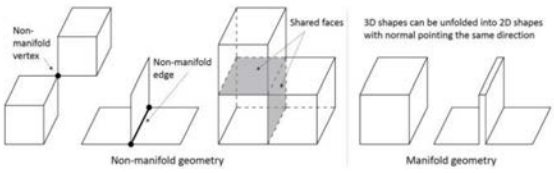


Figure 4.2.5: *Examples of Non-manifold geometries*
Source: Chatzivasileiadi, Lannon, Jabi, Wardhana, and Aish (2018)

While the geometrical output is the primary data type expected by the Evaluator, the Selector does not directly interact with it. Q-learning, the Selector method presented in Chapter 5, demands unique state identifiers to be used in the construction of a table where visited solutions are recorded and rewarded according to their performance. This implies that our Selector requires identifications rather than geometries. Additionally, the investing mechanism needs to keep track of the costs implied in the expansions so that it would be able to determine if an extension investment is carried out. Because of this, the "*Agent*" class comes with other attributes that allow fluent communication with these other components. The "*state*" attribute is a string formed by Ts and Fs. These result from a geometrical operation that starts by considering the maximum amount of modular units a dwelling could potentially occupy. At the centre of each of these units, one point is located so that

each has a unique index within a list. When required, these points are tested for their location in relation to the building envelope so that a "T" results from a point being within the envelope and an "F" from it being outside it. As these characters occupy in a string the location of the index of their test point in their respective list, a unique ID is achieved for every possible geometrical output. In the case of non-manifold geometries (which cannot be closed and therefore pass all points as being outside), they are identified with a string with as many "F"s as potentially available modules.

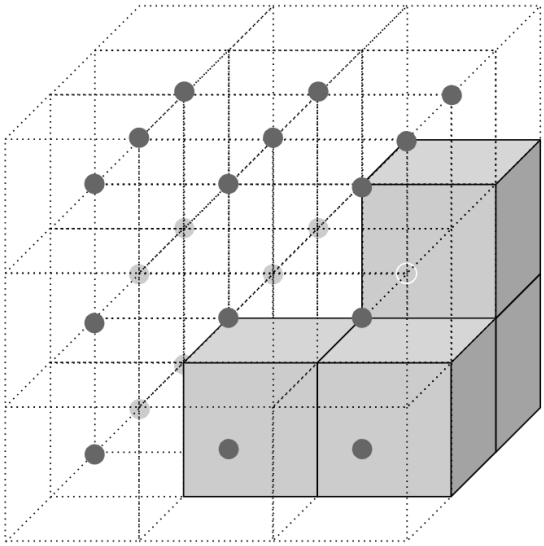


Figure 4.2.6: *Test points to get dwelling geometry size and state*
The maximum potentially occupied space by the dwelling is divided into modular cubes, each of which has a point in the centre. These points are tested so that if they are within the dwelling's envelope, they return a "T"; otherwise, they return an "F". These characters form a string, which identifies a geometrical state. Counting the "T" occurrences determines the current size of the geometry in modular cubes.

With this method, it is also possible to determine potential future states by identifying the closest external point to each of the extrudable surfaces and turning their corresponding character from "F" to "T". It also becomes possible to measure the size of dwellings in "room" units. This is achieved by simply counting the number of points inside the geometry and is presented using the *OccupiedPoints* attribute from the agent object.

To calculate the extension costs, the script differentiates the new faces from the pre-existing ones by comparing the two most recent geometrical states and returning the indices of the additional faces in the latter. Then, the new faces are classified

into walls and roofs using a *Rhino.Geometry* method, and then total areas by class are calculated. With an input indicating the approximate construction cost per square meter differentiated in the same classes, it is possible to get an approximate total price of the extension.

The Block class

As described earlier, the "*Agent*" class has a composition relationship with the "*Block*" class, so that a block is composed of dwellings (or agents in this case). This adds operability to manage multiple dwellings accumulated in a block or neighbourhood. As "*Agent*" is the basic class of the programme's structure, blocks were built on top of it so that a block instance initiates several agent instances, ordering them according to additional inputs. The Block class also profits from inputting some common properties shared among all the composing agents. As such, to initiate a block instance, we need the same inputs as we did for the Agent class (indicating the properties of its composing units) plus the following:

- An integer indicating the number of continuous lots on a single side of the block.
- A Boolean indicating whether the lots will be mirrored to form a double-sided block.
- A Boolean indicating if the initial row of lots will be done in X (True) or Y direction (False).
- A floating-point positive number indicating the view range of agents to identify their neighbours within the block.
- A floating-point positive number indicating the cost of building one square meter of walls.
- A floating-point positive number indicating the cost of building one square meter of roof.

The generation process of a *Block* object starts by creating an instance of the class *Agent* with a starting point coinciding with the block's initial point. The rest

of the agents in a line of houses are started by using their (unique) X or Y sizes (depending on the Boolean input) and calculating their starting points from there. This first line of dwellings is mirrored if the second Boolean input indicates so. With the range view input, direct neighbours are identified. This floating point positive number is used as a ratio to create a circle from the centre of the agent's lot. The other agents' lots vertices are evaluated in relation to that circle so that immediate neighbours are those who have at least one vertex within the circle. Their IDs, stored in a list, become an attribute of the initial agent. Neighbour identification is helpful for the Selector method, as it allows the construction of a reward table that only includes the immediate neighbours (see Chapter 5), thus saving memory resources and making possible the execution of the workflow.

4.2.2 Adding properties to the geometry with LadybugTools

As specified earlier, EnergyPlus, our selected Evaluator, requires, in addition to the geometrical information provided by the GG script, non-geometrical data, including usage schedules and occupancy ratios, among others. To supply this information and complement the GG script output geometries, the present workflow uses LadybugTools.

To understand LadybugTool's contribution, we first need to understand the structure and mechanics of this collection of applications. LadybugTools is formed by four software applications, each serving a different purpose but all related to assessing the built environment. These tools are Ladybug, Honeybee, Dragonfly and Butterfly. The most basic of them is Ladybug (M. S. Roudsari et al., 2013), a tool capable of importing and interpreting EnergyPlus Weather files (.epw) and performing different types of studies using that data (as solar shading, wind, and daylight among other analyses). Ladybug does not perform complex energy analysis as a tool designed for early-stage prototyping. Its studies are limited by the information taken only from geometries and weather files. Originally, Ladybug depended on *Rhino.Geometry* to perform its analytical operations, but today, it relies on its own Python library, Ladybug Geometry (Mackey, Garay, Roudsari, Dao, & Vasanthakumar, 2023). This library, which does not allow NURBS geometries, makes it

possible for Ladybug, and other applications built on top of it to be portable and interact with various CAD software.

Meanwhile, Honeybee (M. Roudsari et al., 2023), built partially on top of Ladybug and thus profiting from some of its functions, serves as a bridge between these various CAD software and Radiance [1] and EnergyPlus. To achieve its functionality, Honeybee uses OpenStudio's CLI and API capabilities. Meanwhile, to be able to manipulate geometries, Honeybee relies on the *Ladybug-Geometry* library and on its own class structure (Mackey, Roudsari, Vasanthakumar, Peng, & Dao, 2023), which, as required by the Evaluator software it communicates, adds non-geometrical properties to buildings' models.

DragonFly (Charan et al., 2021), allows the creation of district-scale models compatible with URBANopt (an energy simulation software based on EnergyPlus and OpenStudio), OpenDSS (an electric power distribution system simulator. Electric Power Research Institute Inc., 2023), REopt (a techno-economic decision support platform. The National Renewable Energy Laboratory, 2023), and the Urban Weather Generator (a Python application for modelling the urban heat island effect. Vasanthakumar, Mackey, & Dao, 2023). DragonFly has its own class structure, a simplified version of Honeybee's adapted to the urban scale. Finally, the most recent tool in the LadybugTools set is Butterfly (M. S. Roudsari, Dao, Mackey, Subramaniam, & Bachant, 2023), which allows creating and running advanced computational fluid dynamic (CFD) simulations using OpenFOAM.

Because of its simplicity and rapid execution, Ladybug was used in the development stage of the GG script to calculate annual solar irradiation availability (see Chapter 5). Nevertheless, as the objectives of the thesis demanded energy simulations, Honeybee had to be used at more advanced stages (See Chapter 6). This tool was preferred over DragonFly due to its more complete class structure and flexibility. Additionally, as EnergyPlus allows predicting the performance of each individual building, its implementation is fitted for an Agent-based approach.

To understand better the interaction between Honeybee and the GG script, it is vital first to understand Honeybee's JSON schema. A JSON schema is a declarative format describing the structure of other data, thus allowing communication

[1] Radiance (Ward, 1994) is a suite of tools that use ray tracing for lighting simulation

between programmes, even if they are written in different languages or work from different platforms. Honeybee uses it to describe the structure of 3D objects and complementary data in a format that OpenStudio and EnergyPlus can understand. As such, Honeybee's JSON schema is organised into five geometry objects (*Room*, *Face*, *Aperture*, *Door* and *Shade*) representing planar geometries (Mackey, Roudsari, et al., 2023). *Rooms* represent single closed volumes and map to a *zone* in EnergyPlus. These objects are composed of planar *Faces*, representing walls, roofs, and floors. *Apertures* (representing windows, light tubes, daylight domes or any other light-carrying opening) and *Doors* (opaque or transparent) can be assigned to Faces, while *Shades* can be assigned to any of the other four geometrical objects.

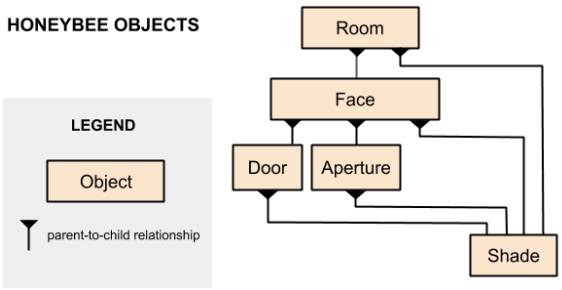


Figure 4.2.7: *The five geometric objects of the Honeybee schema*
Source: (Mackey, 2020)

These five objects are added to a Honeybee *Model* object, which is ultimately sent to EnergyPlus or Radiance using Honeybee's own JSON format (.hbjson). Honeybee *Model* objects can only be simulated if they contain at least one *Room*, being the rest of the base objects, optional data. The *Model* object and its constituent geometric objects are meant to be extended with properties (non-geometric data) for energy simulation. This is done through a "properties" key that each object possesses. These keys include properties for materials, constructions, construction sets, schedules, and program types. In addition to the *Model* object, Honeybee generates *SimulationParameter* objects, which represent a complete set of EnergyPlus simulation settings, including the type of calculation to run, running time, and results to record. The *SimulationParameter* JSON schema is wholly separate from the *Model* schema but still essential for running an energy simulation. The simulation only runs once a *Model* object, a *SimulationParameter* object and an EnergyPlus weather File (.epw) are provided.

To send the resulting geometries for one agent to EnergyPlus, the GG script

uses the Honeybee's *RoomSolid* function to transform the *BREP* representing a dwelling's envelope into a *Room* object. Once the *Room* is created, it is possible to add apertures to it; the easiest way is to use the *Apertures by ratio* function from Honeybee, which allows creating them with a simple aperture-to-wall ratio as input. Next, for each agent's *Model* object, the geometries of all the other agents in a Neighbourhood are added as *Shades*. Finally, the non-geometrical properties are added to the envelopes, which are single *Room* objects.

When the energy simulation is processed locally, Honeybee executes OpenStudio from the CLI and automatically returns the performance values as numbers; meanwhile, when the simulation is executed on the cloud, the script automatically uploads the required files and then waits for and downloads the results. As the results come in a JSON format, the Selector needs to open these files and find the relevant figures.

4.3 Conclusions

The current chapter has presented the optimisation environment, including both the development environment for the execution of the optimisation programme and the GG script for the generation and modification of the required building envelope geometries. By describing two alternative development environments to be used at different stages of the programme's development, it was determined that together, they are able to comply with all three requisites initially established.

Consequently, the chapter describes the structure of the GG script, dividing its tasks into two consecutive steps. First, the geometry generation itself, which is concerned with the management of Brep objects and the output geometry-related useful identifiers to feed the Evaluator and the Selector. Second, the addition of supplementary data, including finer geometrical details and non-geometrical information required by the Energy Simulation. The first of these is presented as an original contribution in the form of a script written in Python using the OOP paradigm. The latter, meanwhile, uses the functions and tools offered by Honeybee, one of LadybugTools' applications. As a result, for each agent, a Honeybee model and a Simulation Parameters object are delivered to EnergyPlus in a JSON format, thus

concluding the task assigned to the GG script.

Although the GG script is extensively described in this chapter, its functioning is not yet presented or tested. This is because this script cannot be tested in isolation but only as part of a workflow. Particularly relevant are its interactions with the Selector and the Evaluator. As such, the following two chapters are devoted to testing the GG script along with the two Development Environment setups presented in this section. Altought the results shown in both chapters are promising, some issues could have been tackled at this stage to have a more streamlined workflow.

For example, writing the GG script relying exclusively on *rhino3dm* or *Ladybug geometry* modules from the beginning would have allowed complete freedom from Rhino software and, thus, from the Windows platform. Altought this would have resulted in a lack of graphical output at the development stage of the GG script, this could have been supplied by an interaction of the script with other GUI alternatives available as Python modules. This could have given more flexibility to the workflow, including the capacity to be executed on Linux and, thus, on available HPC facilities.

A potential future implementation in these platforms is still possible, which opens the door to testing larger urban scenes and more complicated social interaction models without having to rely on costly commercial platforms. Despite this and other minor changes that could have improved marginally the efficiency of the workflow, the set of choices presented and justified in this chapter have catered for the needs of the thesis, so that they have been able to deliver enough results to arrive at the posed goals.

References

Charan, T., Mackey, C., Irani, A., Polly, B., Ray, S., Fleming, K., ... Goldwasser, D. (2021, 9). Integration of open-source urbanopt and dragonfly energy modeling capabilities into practitioner workflows for district-scale planning and design. *Energies*, *14*, 5931. doi: 10.3390/en14185931

Chatzivasileiadi, A., Lannon, S., Jabi, W., Wardhana, N., & Aish, R. (2018, June). Addressing pathways to energy modelling through non-manifold topology. In *Proceedings simaud 2018.*

Electric Power Research Institute Inc. (2023). *Opendss.* Retrieved from https://www.epri.com/pages/sa/opendss

Guglielmetti, R., Macumber, D., & Long, N. (2011). Openstudio: an open source integrated analysis platform.
([computer software])

Ladybug Tools LLC. (2023). *Pollination.* Retrieved from https://www.pollination.cloud/

Mackey, C. (2020, 1). *Model schema.* Retrieved from https://github.com/ladybug-tools/honeybee-schema/wiki/1.1-Model-Schema

Mackey, C., Garay, S., Roudsari, M. S., Dao, A., & Vasanthakumar, S. (2023). *ladybug-geometry.* Retrieved from https://github.com/ladybug-tools/ladybug-geometry

Mackey, C., Roudsari, M., Vasanthakumar, S., Peng, M., & Dao, A. (2023). *honeybee-schema.* Retrieved from https://github.com/ladybug-tools/honeybee-schema

McNeel and Associates. (2014). *Grasshopper. algorithmic modeling for rhino.* ([computer software])

McNeel R. and Associates. (2023a). *rhino3dm.* Retrieved from https://github.com/mcneel/rhino3dm

McNeel R. and Associates. (2023b). *rhinocommon.* Retrieved from https://github.com/mcneel/rhinocommon

McNeel R. and Associates. (2023c). *rhino.inside-cpython.* Retrieved from https://github.com/mcneel/rhino.inside-cpython

Poco-Aguilar, S. E. M., Wate, P. S., & Robinson, D. (2022, December). Multi-agent

learning of incremental housing development strategies for solar utilisation in peru. In *Proceedings of bso conference 2022: 6th conference of ibpsa-england* (Vol. 6). Bath, UK: IBPSA-England. Retrieved from https://publications.ibpsa.org/conference/paper/?id=bso2022_28

Roudsari, M., Subramaniam, S., Dao, A., Vasanthakumar, S., Mackey, C., Haider, A., & Armour, T. (2023). *honeybee.* Retrieved from https://github.com/ladybug-tools/honeybee

Roudsari, M. S., Dao, A., Mackey, C., Subramaniam, S., & Bachant, P. (2023). *Butterfly.* Retrieved from https://github.com/ladybug-tools/butterfly

Roudsari, M. S., & Mackey, C. (2017). *Ladybug tools.* ([computer software])

Roudsari, M. S., Pak, M., & Smith, A. (2013). Ladybug: a parametric environmental plugin for grasshopper to help designers create an environmentally-conscious design.. ([computer software])

The National Renewable Energy Laboratory. (2017, 9). *Energyplus™, version 00.* Retrieved from https://www.osti.gov//servlets/purl/1395882 ([computer software])

The National Renewable Energy Laboratory. (2023). Reopt energy integration optimization. Retrieved from https://reopt.nrel.gov/

TIOBE Software BV. (2023). *Tiobe index for september 2023.* Retrieved from https://www.tiobe.com/tiobe-index/ ([Webpage])

Trimble Inc. (2000, 8). *Sketchup.* ([computer software])

U.S. Department of Energy. (2021, 3). *Energyplus version 9.5.0 documentation. energyplus essentials.* Retrieved from https://energyplus.net/assets/nrel_custom/pdfs/pdfs_v9.5.0/EnergyPlusEssentials.pdf ([computer software])

Vasanthakumar, S., Mackey, C., & Dao, A. (2023). The urban weather generator. Retrieved from https://github.com/ladybug-tools/uwg

Ward, G. J. (1994). The radiance lighting simulation and rendering system. In *Proceedings of the 21st annual conference on computer graphics and interactive techniques* (pp. 459–472).

Chapter 5

Optimisation Selector

The current chapter presents and tests the method for the Selector component belonging to the optimisation loop of our methodology. As such, it begins by framing the optimisation task within the Architectural Design Optimization (ADO) field of study. After reviewing the most utilised methods in this field, the chapter arrives at the conclusion that none fits our task due to the dynamic nature of the incremental housing optimisation problem. As an alternative approach is needed, the field's theoretical foundations in simulation-based optimisation reveal control optimisation as a viable tool. By taking into account operations research and control literature, the chapter defines our optimisation task as a Markov Decision Problem (MDP), which could use reinforcement learning (RL) as an optimisation mechanism. To provide empirical ground for this choice, the chapter presents a test done to optimise a single building envelope to maximise its solar exposure.

As positive results are achieved, the chapter continues by presenting RL but now from the perspective of machine learning (ML) literature. This serves as a way to introduce multi-agent reinforcement learning (MARL) so the optimisation of multiple geometries is achieved. This leads to the adaptation of the previously presented algorithm for a successful scaling-up from single to multi-agent RL. An adapted version of naïve Q (λ) joint-action learning is thus presented and tested as the ideal method to be used for the purposes of this research. Finally, the chapter arrives at conclusions and recommendations on the implementation of the adapted algorithm on the posed workflow.

5.1 The location dilemma as an optimisation problem

The dweller involved in incremental housing development faces two consecutive dilemmas when confronted with the *need* to expand. First comes the financial dilemma (*can I afford it?*), which, with a positive response, leads to the location dilemma (*where will I place the extension?*). This research established that the earlier would be tackled by an investment decision model (Figure 1.1.1) that mimics the logic behind incremental housing development in the global south, essentially modelling the 'trigger' to incrementally expand. The location dilemma, meanwhile, as it has repercussions on the building envelope's geometry (the optimisation's objective function) and represents the cognitive function of agents in an ABM, was assigned to the optimisation loop. In this sense, the present chapter deals exclusively with the search for a method that finds satisficing solutions when dwellers face the second dilemma.

In this context, using optimisation methods to find satisficing solutions becomes a building form-finding problem. In building design, solving this problem is usually a matter reserved for architectural design, which directs us to the concept of architectural design optimisation (ADO) (Wortmann & Nannicini, 2017). This sub-field of study makes use of computational design methods to optimise quantifiable variables of building design problems. It has become increasingly popular in the last decades due to the acknowledgement by the design community that optimisation can be a powerful tool for structural design, form finding and compliance with increasingly demanded environmental design certifications. Although any quantifiable variable can be optimised, ADO is particularly interested in the interrelationship of those that architects value more: form, function and structure.

To understand better the optimisation problem at hand, we need to first list the objective function, the variables and the number of possible solutions:

- The objective function of the optimisation problem is aligned with the imputed agents' rationally bounded behaviour. This is complemented by additional information shared with the agents due to the universal policy choice mentioned in Chapter 1. This shared information differs according to the development

stage of the workflow. In the version shown in this chapter, the shared information is the total number of sunlight hours on the building envelope. The imputed behaviour meanwhile responds to a minimisation of the predicted operational energy costs coming from the agent's expansion location decision. In this chapter this is implicit, as agents try to maximise solar exposure during winter months.

- The variables implied in these optimisation problems are the building envelope geometry and the performance values shared with the agents. As such, a modification in the envelope's geometry (implicated in the expansion location choice of the agent) entails a variation in the performance metrics.

- The possible solutions in the search space are as many as there are intermediate modular arrangement possibilities before the buildable space (within lot and height limitations) becomes fully filled. In the experiments shown in this chapter, this is possibly in the order of thousands.

Many optimisation methods exist to make ADO possible. Nevertheless, the list of those frequently used in industry and academia is relatively short. The ADO community relies highly on model-free stochastic parametric optimisation methods (Wortmann, Cichocka, & Waibel, 2022). Stochastic methods, called simulation-based methods, allow the maximisation (or minimisation) of net rewards obtained from a random system (Gosavi, 2015c). Parametric refers to the fact that their goal is to find the values of parameters that maximise or minimise a function, and model-free refers to the fact that they do not require the analytical form of the objective function (Gosavi, 2015c). Given these categories, the preference for these methods in the ADO community is easy to guess, even more so considering the usual requisites within the industry: Stochastic optimisation has an advantage over deterministic as execution time is prioritised over finding absolute global optima; meanwhile, a model-free approach allows profiting from existing commercial (energy, lighting, etc.) simulation software, thus making the process more expedite.

The apparent relative advantage of parametric optimisation to solve most ADO problems is particularly relevant to the present chapter. This approach is used considering the simplifying assumption that the built environment is static in time.

Although this is not true, as we know that cities are dynamic systems in permanent change, this assumption allows getting close-to-optimal solutions useful for enough of the building's life cycle while using powerful optimisation algorithms that require relatively modest computer power. Although this assumption is enough to solve most ADO problems, it is not so for our case study. Incremental housing can only be thoroughly analysed by considering its highly dynamic nature. Thus, a static approach to its optimisation would provide optimal solutions only for a single state of the multiple that a dwelling visits before arriving at its conclusion, each state being influenced by its neighbouring context. An approach that considers moving a system along a desirable path of states instead of providing a solution for only one seems more suitable for our case study [1].

In simulation-based optimisation literature, the alternative to parametric (also called static) is control (also called dynamic) optimisation (Gosavi, 2015b). Under this other paradigm, we are not trying to optimise the parameters that maximise (or minimise) the rewards on a single state, but, as there are several states which change "dynamically", we try to optimise the "decisions" or "actions" (hence the name control) that we should take at each state to achieve a given goal (Gosavi, 2015d, 2015b). Thus, the outcome is an optimal sequence of states and actions rather than parameters that lead to a single-state solution. It seems clear that, if simulation-based optimisation is the way to follow, dynamic optimisation is the best choice for optimising the location dilemma in incremental housing.

To understand better how control optimisation can be applied to our case study, we must first understand it as a system. Roughly speaking, a system is "a collection of entities that interact with each other" (Gosavi, 2015d) whose behaviour, in the case of dynamic systems, can be described in terms of its number of states (also known as traits). Control theory, the field of study that gave birth to dynamic optimisation, is interested in how these traits change over time. We can quickly identify both entities and states in our case study. At the macro level (in a neighbourhood), each dwelling is an "entity"; at the micro level, each household member

[1] Besides this, we have to consider that optimisation (as used in this workflow) also serves as an agent-cognition tool in a multi-agent setting. As such at an aggregate scale, the method must allow for a decentralised system that enables communication among agents, thus leading to negotiation and compromise. This factor, related to the scalability of the system, is discussed in greater depth later on in this chapter. Here, the discussion is focused on the operational affinity between incremental residential development at the individual level and dynamic optimisation methods

is one. These entities interact with each other (one's dwelling expansion affects the solar exposure of their neighbours, while the income-generating status of members affects the resource availability of the household) and decide where they will extend a dwelling. States, meanwhile, are the different spatial configurations that dwellings visit until they are considered "finished". Another characteristic of dynamic systems is that their behaviour can be explained by "governing variables" (Gosavi, 2015d), which in our case are the subjective choices of each household to build on a determined region of space at a determined time within constraints (legal, financial and spatial). As this choice is usually driven by purely subjective preferences, it can be considered random. We are thus dealing with a dynamic stochastic system.

Stochastic systems are usually associated with a stochastic process. Roughly speaking, these processes have a property that changes randomly with time (Gosavi, 2015b), typically in response to some known underlying stimulus; in our case, the dwelling's geometry. One example of a discrete stochastic process is the Markov process, which is characterised by three essential properties (Gosavi, 2015b):

- The jumpy property, as they jump regularly between states.
- The memoryless property, as the probability that the process jumps from state i to state j, does not depend on the states visited by the system before arriving at state i.
- The unit time property because the process jumps between states after *unit* time.

To be sure that we are dealing with a stochastic process, we can check these properties individually in our case study:

- First, as dwellings constantly expand and change their geometrical configurations, we can be sure we comply with the jumpy property.
- Second, we comply with the memoryless property as at every time-step households decide where to expand driven by subjective preferences not reliant on their previous choices, and influenced by the evolving actions of their neighbours. We could argue that their decisions are affected by the current configuration of the dwelling, which derives from decisions taken in the past. However,

bear in mind that we are not directly evaluating those decisions but merely their consequences in the current state (or geometrical configuration).

- Finally, we know that the expansion process happens at any moment in which dwellers have the need and resources; nevertheless, for modelling purposes, we need to discretise this so geometries can only change once a year, thus complying with the third property.

Now that we have determined that we are dealing with a stochastic system associated with a (by simplification, discrete) stochastic process, we can define the location dilemma as a Markov Decision problem. As said before, households face a financial dilemma to determine if they can afford to build an extension in certain time steps. They have little choice here, as they depend on external uncontrollable variables (cost and financial means). As shown in the socio-economic model, probable future states of these variables can be predicted using a stochastic model that relies on Transition Probability Matrices (TPM). As no external agency can control the process, when Markov chains exist, these are "uncontrolled".

Nevertheless, this is not the case in the location dilemma. Once the decision on expansion has been taken, households have to decide where to locate this extension, and here, they have restrained control over their choice. Therefore, they can choose an action to be taken at each state. Because there is likely more than one option of location each time they are faced with the dilemma, and because we assume that the time elapsed between state evaluations is uniform, we have a Markov decision problem (MDP). This is a problem of control optimisation or, more precisely, a problem of finding the optimal action to be selected at each state (Gosavi, 2015d). In the following section, we will learn about the methods to tackle this type of problem using dynamic optimisation.

5.2 Dynamic optimisation methods

To tackle MDPs, control optimisation uses a framework that allows analysing any process of interest. This framework is made up of five elements (Gosavi, 2015b):

- A decision maker (also called agent)

- Policies
- Transition probability matrices
- Transition reward matrices
- A performance metric

At each time step, the agent has to select an action while in a given state. The sequence of actions and states is the so-called policy. As at each state, multiple actions can be chosen; each one is associated with a probability, thus, a probability transition matrix exists. We need to weigh "optimal" policies to achieve our objective function. Therefore, a transition reward matrix appears as a way to associate rewards with transitions, leading to optimal policies. Finally, we can derive a performance metric from the objective function to compare policies and select the optimal one.

The most straightforward method to tackle MDPs is exhaustive enumeration, which consists of enumerating every policy with the possibility of being selected [2], evaluating their performance metric, and declaring the policy that produces the best value to be the optimal one. This approach has its obvious limitations, as we can only afford to check each policy's performance within time and computational constraints in very small problems (Gosavi, 2015a). Dynamic programming (DP) (Bellman, 1966) is more efficient for larger problems as it has a considerably lower computational burden. DP requires the computation of a so-called "value function" for every state that rests on a simple linear system of equations where the number of equations equals the number of decision-making states in the Markov chain ($|S|$). This system is often called the Bellman equation (Equation 5.1).

$$h_{\hat{\mu}}(i) = \bar{r}(i, \mu(i)) - \rho_{\hat{\mu}} + \sum_{j=1}^{|S|} p(i, \mu(i), j) h_{\hat{\mu}}(j) \text{ for each } i \epsilon S \tag{5.1}$$

Where:

- The unknowns are the $h_{\hat{\mu}}$ terms, which are the elements of the value function vector associated with the policy $\hat{\mu}$.

[2] An alternative to this brute force approach is static optimisation at each time-step, but this is out of bounds of the dynamic optimisation discussion.

- $\mu(i)$ denotes the action selected in state i under the policy $\hat{\mu}$.
- $\bar{r}(i, \mu(i))$ denotes the expected immediate reward in state i under policy $\mu(i)$.
- $p(i, \mu(i), j)$ denotes the one-step transition probability of jumping from state i to state j under the policy $\hat{\mu}$.
- $\rho_{\hat{\mu}}$ denotes the average reward associated with the policy $\hat{\mu}$

To obtain an optimal policy, we use policy iteration; this is, we start with an arbitrary policy, evaluate it with the Bellman equation and, with that metric stored in a vector, we can move to a better policy at every iteration until no further improvement is possible.

Although DP guarantees to give optimal solutions to any MDP, it is doubly cursed by the curse of dimensionality and by the curse of modelling (Gosavi, 2015a), both related to DP's reliance on transition probabilities. Reinforcement learning (RL) is a good alternative. Based on DP, it avoids these two curses because (Gosavi, 2015a):

- As a model-free method, it does not need the transition probability matrices as it stores the value function as Q-factors.
- When the MDP has millions of states, RL does not store the Q-factors explicitly but uses function approximation methods to approximate the Q-factors of millions of states.

The most straightforward form of RL is Q-learning (Watkins & Dayan, 1992). This model-free algorithm "can be derived from the Bellman optimality equation for discounted reward MDPs" (Gosavi, 2015a), which allows for finding an optimal policy using a value iteration method. This method differs from policy iteration because it starts with arbitrary values for the value function vector and applies an updating mechanism derived from the Bellman equation to update these values. In vanilla Q-learning (not customised or modified from its original form), the Bellman equation is presented as follows (Watkins & Dayan, 1992):

Considering an initial state $s \epsilon S$, in which we select an action $a \epsilon A$,
resulting in a new state $s' \epsilon S$:

$$s \ \rightarrow \ a \ \rightarrow \ s'$$

Calculate a Q-value for (s, a) with the following formula:

$$Q_{\text{new}(s,a)} = (1–\alpha)\ Q(s,a) + \alpha(R_{t+1} + \gamma maxQ_{(s')})$$

Where:

- $Q_{(s,a)}$ is the existing Q-value to be replaced.
- α is the learning rate, which weights the existing Q-value to be replaced against rewards.
- R_{t+1} is the reward received from choosing a while on s.
- γ is the discount factor, which weights possible future rewards ($maxQ_{(s')}$) against the reward obtained from choosing a while on s (R_{t+1}).
- $maxQ_{(s')}$ is the maximum Q-value registered so far for the actions available on s'.

The value vector, called a Q-table, stores one value (called Q-value) for every state-action pair. Q-values are updated using the Bellman equation every time the same state-action pair is revisited. With every additional iteration, Q-values collect information about the reward from selecting a given action in a given state. With this, an optimal policy is formed from its parts. When the agent implements a reward-greediness mechanism, it will, at each state, prefer the action with the highest Q-values and avoid the ones with the lowest ones. As updated Q-values receive feedback from future ones ($maxQ_{(s')}$), the agent has information about rewards available in the subsequent stages of the policy chain, thus moving towards an optimal policy rather than focusing myopically on the nearest reward. The discount factor (γ) variable can control the importance of these future rewards.

The process to implement vanilla Q-learning is the following (Watkins & Dayan, 1992):

Step 1 Initialize the Q-value for each state-action pair:

$$Q_{(s,a)} = 0, \ \forall \ s \ \epsilon \ S \hat{} \forall \ a \ \epsilon \ A(s)$$

Step 2 Repeat the following for each iteration (also called episode):

- Choose an action a from the current state s $a = \text{argmax}Q_{(s,a)}$
- Take action a and observe the next state s' and the immediate reward r.
- Update the Q-value of the current state-action pair using the Bellman optimality equation for discounted reward MDPs (Equation 5.1).
- Set the current state to the next state: $s = s'$

Step 3 Until the Q-value function converges or a certain number of episodes have been completed, repeat steps 2-3.

Although in small search spaces, this process is enough to find an optimal policy in a short time, in larger ones, the agent might conform to local optima (a solution optimal only in regards to the small section of the solution space that a "lazy" reward-greedy agent explores), which could be sub optimal when looking at the full picture. To avoid this problem, an epsilon-greedy behaviour can replace reward-greedy behaviour. This consists of including an exploration factor ε, which allows the agent to select random actions sometimes instead of choosing the ones with the highest Q-values. With this, the agent can "explore" the search space and avoid the negative consequences of its reward-greediness. This divides the optimisation process into an exploration phase and an "exploitation" phase, where the agent reinforces a "positive" behaviour with the information it has previously collected. To implement this, we modify the first point of step two and introduce a probability factor:

$$a = \text{argmax}Q_{(s,a)} \text{ with probability } (1 - \varepsilon)$$

$$a = \text{random with probability } \varepsilon$$

By having a decaying ε value, we can ensure that the probability of having a random solution is high on the earlier iterations (exploration phase) and lower on the latest ones (exploitation phase).

In summary, the location dilemma is a MDP, which can be optimised using RL, a control optimisation method. Under this framework, we have an agent (a

household), a set of states (possible geometric configurations) and actions (location decisions) that form possible policies, and an objective function related to the shape of the building envelope. From the optimisation process, we should get a policy that minimises energy use while maintaining thermal comfort. Given these precepts, we can now test RL in our case study.

5.2.1 Single-Agent reinforcement learning

To test the implementation of the described algorithm in our case study, a much simpler objective function was used: maximise the building envelope solar irradiation hours in the coldest months for an urban location in Peru. The results of these tests were initially presented in Poco-Aguilar, Wate, and Robinson (2022).

The vanilla Q-learning algorithm was implemented in the Development Optimisation Environment described in Chapter 4. All custom components were written in Python and implemented using Grasshopper (McNeel, R and Associates, 2014). The geometry's performance was evaluated using Ladybug (Roudsari, Pak, & Smith, 2013), a Grasshopper component designed to analyse weather data and perform solar irradiation computations. The initial configuration of the dwelling consisted of a single cubic module with a length of three meters per side, able to extend one module of the same dimension per each time step within the lot's physical limits and legal height limits. The lot's dimensions were set to be equal to the dimensions of three cubic modules in depth (9m) and two modules in width (6m), while the legal limit was set to three modules in height (9m).

The agent's lot was surrounded by fixed-shape neighbours on three of its sides (Figure 5.2.1), with the particularity that, being in the southern hemisphere (-16.32 latitude, -71.55 longitude, 2520 m.a.s.l.) in wintertime (21/03 until 23/09), the agent had to find a policy that allowed it to surpass its shadowing neighbours. Finally, although the socioeconomic model was not tested yet, socioeconomic inputs had to be defined for the proper functioning of the algorithm. These were a fixed total budget of 15k monetary units (MU) and a fixed need for space of six occupied modules. Each square meter of newly built walls costs 100 MU, while roofs cost 150 MU. There is no further justification for these precise figures, as these serve just to test and refine the presented workflow. The only explainable variable is the relative

difference between them, which intends to reflect the increased level of difficulty (translated to costs) in the construction procedure of roofs relative to that of walls. The agent always has the need to build, but its financial capacity at each episode is limited by its total budget; thus, part of the challenge is building the needed modules without depleting the total budget.

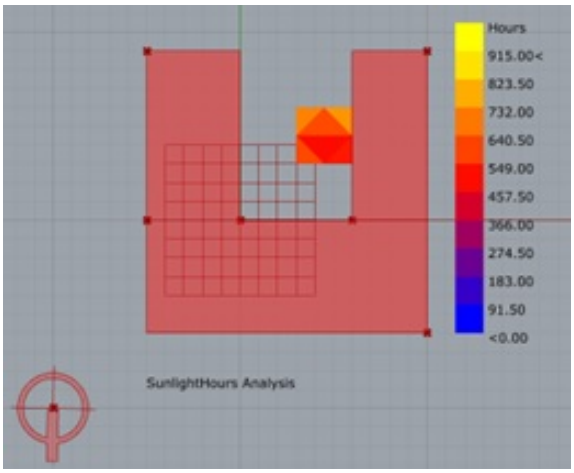


Figure 5.2.1: *Initial state of the housing core for experiment one. Surrounding buildings on red, core coloured with the analysis palette.*
Source: Poco-Aguilar et al. (2022)

To achieve step 1 of the process (initialise the Q-value for each state-action pair), the agent needed to know possible states and actions in advance. To do so, the agent calculates the states it can visit within the time-space constraints. As described in Chapter 4, states are defined as strings formed by "F"s and "T"s according to the status (within or outside the building envelope) of the central point of each of the three-dimensional modules in which the buildable space is divided. With this in mind, the algorithm generates all the possible combinations of these two letters in a string as long as the number of potentially occupiable number of modules that exist in a lot. To avoid wasting memory in generating and storing unnecessary vector entries, there is a pre-simulation process in which the algorithm gets the identification number of all the points that could be occupied within the maximum time-steps constraint, departing from the seed cubic module. While states have unique identifiers, actions across states have a standard identification number related to the available face of the envelope to be modified. As there are different actions per state, the algorithm calculates the maximum number of possible actions among all possible states. Thanks to the feature programmed on the geometry generation

algorithm that reduces the face identification number when this is beyond limits, higher action identification numbers are not a problem. This absolute maximum action number is calculated using the following empirical formula:

$$MAct = LongestDim + (((LongestDim * 2) - 1) * (ShortestDim - 1)) \quad (5.2)$$

Where:

- LongestDim is the longest dimension of the lot.
- ShortestDim is the shortest dimension of the lot.
- When the lot is a square, both positions are interchangeable.

At the beginning of the process, we do not know the maximum achievable performance (maximum possible sunlight hours within physical and legal constraints), so we must discover it from the search space. The method to achieve this is to record the maximum performances the geometry achieves at each episode. With this record in hand, we can reward the agent when the current performance is more or equal (within a tolerance, due to likely geometric imprecision) to the current record. This also means that initially, optimal policies would need to be replaced each time a better-performing one is discovered. This has consequences on the Q-learning settings, as in the initial exploration stages, the values must be replaced without leaving many traces. This also determines "winning" and "losing" policies. An agent "wins" the "game" when the selected policy leads to either breaking the currently established record or keeping it. It will "lose" if it performs well below that record. In this logic, the winning policies are "paths to victory". Because we can only judge the policy once the agent meets its need within constraints or cannot do it when the time is over, the major reward/penalisation is only granted on the last time step of the episode. To allow this, an episode ends when a performance record is broken or the maximum number of time steps is reached. In addition, to minimise the time steps to arrive at a solution, a minimal penalisation is assigned for each time step where the goal has not been achieved.

This maximum number of time steps per episode was set at ten. The algorithm was also given a maximum of five hundred episodes to converge to an optimal policy without any mechanism stopping the parameter if this was achieved earlier. As for the Bellman equation, the variables were set as follows:

- A learning rate (α) started at 0.99 with an empirically determined decay of 0.99976 per step. This ensures that replacement Q values get higher weights against old ones during the exploration phase, while the opposite happens in the exploitation phase. This strategy helps find new optimal policies with the "record-breaking" method.

- A fixed discount rate (γ) of 0.9 defines a non-myopic agent. This is because the "winning" or "losing" character of the episode is only defined on the latest time-step. As such, it is better to weigh more future rewards ($max(Q(s'))$ than immediate ones (R_{t+1}).

- An exploration rate (ε) starting at 0.99 with an empirically determined decay of 0.9982 per step. This allows for determining an early exploration phase, in which policies result from random action choices, and a late exploitation phase, in which positive behaviours are reinforced, stabilising an optimal policy.

- The reward assigned was +25 when the agent "wins the game" and -25 when it "loses". At each time step without an episode-killing result, a minimal penalisation of minus one is awarded to promote a solution with the minimum steps.

The test was run on Rhino 6, working on a Windows 10 PC with an Intel Core i5 vPro processor of 9th generation and an installed RAM of 8GB. The time of execution was about 45 minutes, after which the process automatically stopped when finishing simulating episode number five hundred. The accompanying figures show that the agent did manage to find and learn a satisficing policy. Figure 5.2.2, which shows the episode number on the X-axis, the accumulated rewards per episode on the first Y-axis and a scale for ε and α values on the second Y-axis, demonstrates how, as ε diminishes, a policy with high accumulated rewards stabilises. Only a few remaining random solutions in the last episodes interrupt the flat line on the highest

end. A completely flat line will be only possible if ε is an absolute zero, as just one incorrect (random) action taken in a previous time-step is enough to block the way to a "winning" solution. Figure 5.2.3, which shows the episode number on the X-axis and the maximum performance achieved at the final step of each episode on the Y-axis, demonstrates how the policy leading to the highest recorded performance is found in the exploration phase (higher ε) and it is later recovered in the exploitation phase when it stabilises.

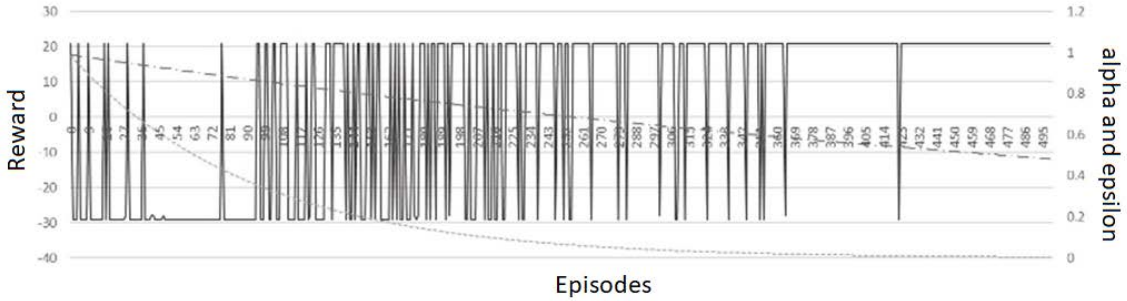


Figure 5.2.2: *Reward accumulated per episode as a dark continuous line, ε as dashed line and α as dashed-dotted.*
Source: Poco-Aguilar et al. (2022)

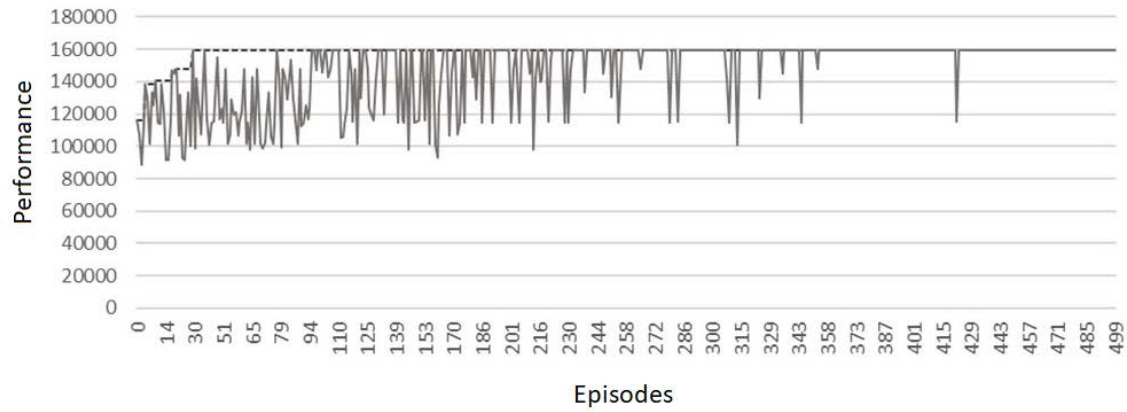


Figure 5.2.3: *Maximum performance per episode as a continuous line, maximum performance recorded as dashed.*
Source: Poco-Aguilar et al. (2022)

These results demonstrate that finding an optimal policy for our case study using RL with a simple objective function and predefined socio-economic variables is possible. They also show that the geometry generation algorithm in earlier chapters works well when integrated with an RL algorithm. Nevertheless, up to this point, we have only tackled the optimisation problem of a single individual, whilst this thesis

aims to deal with the optimisation of entire neighbourhoods. In the next section, we will jump from the unit to the multitude.

5.3 Multi-agent reinforcement learning

RL's simple but powerful approach to solving decision-making problems has popularised its uses beyond control optimisation. So much so that it has become one of the three machine-learning (ML) paradigms. From this perspective, we are not solely interested in optimising a dynamic process but more in how intelligent artificial agents could learn a desired behaviour with minimal human intervention. From the machine learning perspective, the repetitive episodes are experiences needed to "reinforce" a desired behaviour in an intelligent being by using "rewards". In this sense, we are resuming the previously introduced discussion of optimisation as an agent cognition strategy in rationally bounded agents.

The machine learning approach also adds something beneficial to this thesis: agent interaction. In the real world, intelligent beings do not exist in isolation; they interact with their environment and other intelligent agents. The relationship with these other agents adds a layer of complexity to RL, as each agent might have a different objective function. From there, we can imagine extreme scenarios of interaction among agents. In a fully cooperative scenario, for example, the objectives of all agents would be aligned, so they work together towards one goal. Whilst on a fully competitive one, objectives are entirely misaligned, so only one agent can find an optimal policy at the expense of the rest. There could also be team games, where objectives are shared among teammates but opposed between teams, and, in non-zero-sum games, the emergence of negotiation spaces, leading to compromise of parties. These scenarios simplify the intricate interactions that exist in reality, some of which include complex social dilemmas.

Going back to our optimisation problem, one could argue that, when uncontrolled and unoptimised, incremental development approaches more a competitive scenario than a cooperative one. As households cannot predict their neighbours' future states, or profit from those states, they would try to profit the most from the resources they believe they can access at a given time. For example, if I want to invest in

an extension that includes the capacity to implement PV panels on the rooftop, and I am the first one to extend a house in a block with only one-story houses, I could assume that it is better to build that extension on the first floor, as this minimises the risk that my neighbours obstruct my access to sunlight. The logic that accompanies me is competitive due to the likely imminent growth of neighbours. If, just after I finish building my extension, one of my neighbours suddenly builds three stories high on the most inconvenient side for my PV investment, we reach a conflict scenario. Given their individual goal, they will try to maximise their profit within constraints. As building regulations are not typically designed to indicate what can and cannot be done at the intermediate stages of incremental housing development, my neighbours are probably within their rights.

We end with a very unfair situation in which those with immediate access to capital to develop their dwellings at the early stages of a new neighbourhood's occupation could affect those needing more time for capital accumulation. The neighbourhood's full potential to minimise energy use could be held up due to varying access to capital and individualistic goals. An immediate solution to this is, of course, communication. If my neighbour and I had discussed our plans before their execution, we probably could have reached a fair agreement for both parties. Of course, this communication could only have been fruitful if we both had access to adequate information for our goals, such as some sun-path analysis, for example. To avoid future problems, this informed discussion should take place with all my immediate neighbours, and they should have it with their respective neighbours, and so on. In the real world, this sort of coordination is infrequent and hard to achieve. First, it is difficult for our target population to access the information needed for energy use minimisation with thermal comfort targets in mind. Second, even if they had access to this information, communication among all interested parties is complicated, time-consuming and requires technical knowledge. Finally and most importantly, we do not know if all will sacrifice their goals for the common good.

Approaching this problem in the virtual world via single-agent Q-learning is also problematic and shows the limitation of the algorithm for multi-agent scenarios. It also demonstrates that communication channels among agents are needed, even in the virtual world. For example, when using Q-learning as configured for the previ-

ously presented experiment but considering two agents (one next to another), we will encounter that they are unlikely to converge to a solution. This can be explained by their lack of communication and consequential coordination. The following example demonstrates this:

- Let us assume that, due to randomness (ε), *Agent One* does not expand on the first episode (selects action 0) while *Agent Two* does.
- On the last time-step of this first episode, *Agent Two* discovers that going from state "FFTFF" to state "FTTFF" via action "2" leads to record-breaking performance and, thus, to a maximum reward.
- As a consequence, at the end of episode one, the state-action pair (FFTFF, 2) stores a high value in the Q table of *Agent Two*.
- At the end of the next episode, *Agent One* builds two stories tall and occludes *Agent Two*'s access to the sun.
- Faced again with the state "FFTFF", *Agent Two* assumes that selecting action "2" will lead to a high reward.
- Nevertheless, now that its neighbour is affecting the performance of its geometry, that action selection leads to an underperforming result.
- Therefore, in the Q-table of *Agent Two*, a low value will replace the previous high value, and *Agent Two* will temporally forget that optimal policy.

When this happens several times with multiple agents, convergence becomes unlikely, limiting the usefulness of RL as a method. Nevertheless, these inconveniences can be avoided by making the agent "sense" the changes in its environment. If somehow *Agent Two* was aware that choosing the same action does not lead to the same reward, it would become reactive, and its probability of achieving convergence would increase. This leads us to the concept of joint-action learners (JAL), one of the two paradigms in Multi-agent reinforcement learning (MARL) distinguished by (Claus & Boutilier, 1998). On the opposite shore are Independent learners (ILs), who apply Q-learning in the classic sense, ignoring the existence of other agents. JALs, in contrast, learn the value of their actions in conjunction with those of other

agents via the integration of RL with equilibrium (or coordination) learning methods. This method implies having a more complex table to store the Q-values, one that includes the agent's state and tracks the synchronic joint action of the other agents in the environment. The complexity of this is that this table grows exponentially with the number of agents in the environment, complicating the optimisation of large sets of agents. A neighbourhood strategy can be used to avoid this so that only the neighbouring agents that affect the performance of another agent are used to build the Q-table.

These changes in the table also lead to some inconsistencies when applying step 2 from vanilla Q learning:

- We cannot choose directly a from s with the following formula, as there is a bi-dimensional matrix for every s that includes the agent's and neighbours' actions.

$$a = \text{argmax}Q_{(s,a)}$$

- For the same reason, we cannot get a max $Q(s')$ to apply to our usual Bellman optimality equation.

$$Q_{\text{new}(s,a)} = (1–\alpha)Q_{(s,a)} + \alpha(R_{t+1} + \gamma \text{max}Q(s'))$$

To solve this, we will use the Kleiner and Nebel (n.d.) approach. We initialise a complementary table that counts the times each combination of agent's state and their neighbours' joint action is visited ($C_{(s,a_{-i})}$) and another that counts the times every agent's state is visited ($n(s)$):

$$C_{(s,a_{-i})} \leftarrow 0 \forall s \, \epsilon \, S, \forall a_{-i} \, \epsilon \, A_{-i}$$

$$n(s) \leftarrow 0 \forall s \, \epsilon \, S$$

Where: $a_{–i}$ is a vector storing the actions of all the neighbours of agent i (Joint-action vector) but not the action of agent i (a_i) so that $a_i \cup a_{–i}$ contains all the actions selected during the current time-step.

Although the initial values are 0, this will be updated each time we visit the state joint-action combination. With this, every time we have to choose an action,

we can weigh its Q-values against the times the state-joint actions combination has been visited over the times the state has been visited so that the action selection happens by solving:

$$\arg\max_{a_i} \sum_{a_{-i}} \frac{C(s, a_{-i})}{n(s)} Q(s, \langle a_i, a_{-i} \rangle) \tag{5.3}$$

This new setting will also serve to solve the maximum future Q value problem in the Bellman optimisation equation so that the equation becomes:

$$Q(s, \langle a_i, a_{-i} \rangle) \leftarrow (1 - \alpha) Q(s, \langle a_i, a_{-i} \rangle) + \alpha(R(s, a_i + \gamma V(s'))) \tag{5.4}$$

Where:

$$V(s') = \max_{a_i} \sum_{a_{-i}} \frac{C(s', a_{-i})}{n(s')} Q(s', \langle a'_i, a_{-i} \rangle) \tag{5.5}$$

Now that we have the basic modifications that would make RL possible for JAL, we can test it in our case study to provide empirical evidence about its suitability.

5.3.1 An adapted version of Q joint-action learning

We will now test the MARL JAL algorithm and develop a fitted version for our case study. This new version is built empirically by improving the basic JAL algorithm presented in the previous section via a trial-and-error method. As such, this section offers four tests that go from the direct application of the basic JAL algorithm for the optimisation of only two agents and end up in a polished version of the algorithm applied to four agents but readily extendable to as many agents as needed. As each algorithm was built one on top of the previous one, and all were built on top of the single-agent RL one, the configuration is the same as the immediately previous one unless otherwise expressed.

On the first test, the algorithm was applied to optimise only two agents' behaviour. At this stage, the goal was to maximise the performance of only one of them, which means there was a selfish and a selfless agent. As multiple agents require more computational resources, the number of spaces needed was reduced to only four, and consequently, the available budget was reduced to 10.5k MUs. To

limit the time execution and force expansion on each time step, the maximum time steps per episode were the same as the number of spaces needed. An episode finished when the record performance for the selfish agent was either broken or kept within tolerance or when the maximum number of time steps per episode was reached. Rewards and penalisations were kept as in the single-agent case. The location and the performance evaluated were precisely the same as in the single-agent scenario, except for the neighbour's configuration, in which one of the static neighbours is replaced by the second agent (Figure 5.3.1). The Bellman equation had the same inputs as in single-agent optimisation.

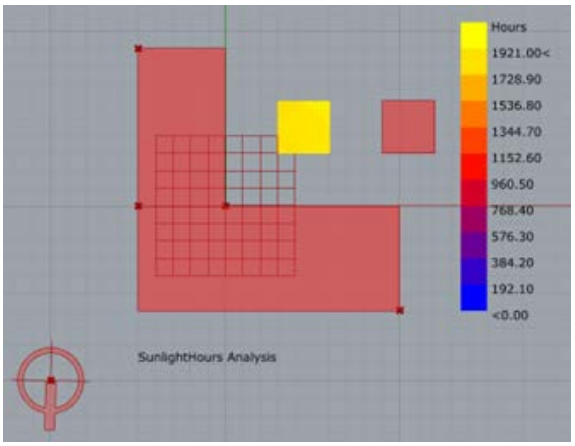


Figure 5.3.1: *Two agent optimisation initial setting.*
Dynamic agents shown as squares, with the selfish agent in false-colour

After 45 minutes of running in the same PC used for the single-agent reinforcement learning, the two agents seem to get close to stability on a high reward policy (Figure 5.3.2). Because the objective is unique and the reward for both depends on the selfless agent's performance, there is only one reward/episode line. First, we can notice that the agents take more time to stabilise a policy. The single-agent scenario reached stability on about the 90th episode, but the dual one did on about the 260th, even though the task is much more straightforward and equation variables remain the same. This might be due to the required coordination of action selections.

The same algorithm was used on a second test but now on four agents. In this test, the process could not be initiated because of the high memory burden of generating four Q-tables with the individual possible states of every agent and the possible combinations of their actions and their neighbours' actions. Taking into

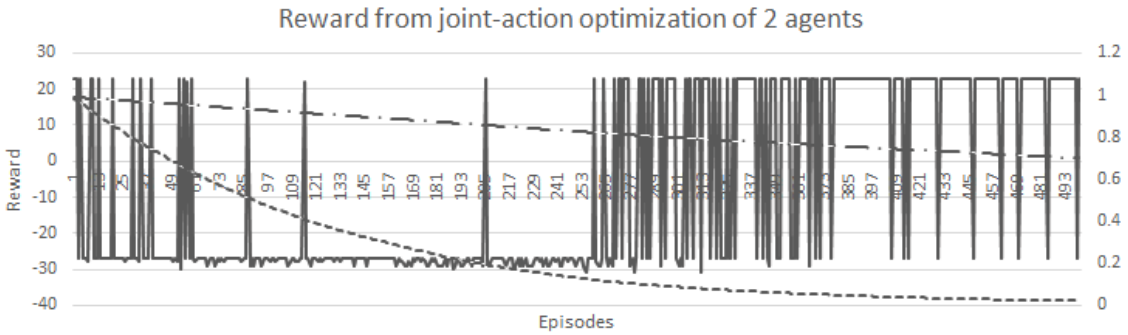


Figure 5.3.2: *Reward accumulated for two joint-action learners*

account that, given the size of the lot, the maximum height, and the maximum number of time steps, the geometry generation algorithm calculates 120 possible states and nine actions at each state on the first time step (t_0), the possible joint-action combinations for four neighbours is 87,480 (93 x 120). This is only for the first time step out of a maximum of four. Because the table was built in advance, an average personal computer would run out of memory at the first step of the RL process. A straightforward solution appeared once this problem was identified (Box 5.3.3). The table was not complete but was built incrementally as new states were visited. So that each time we reach a state, we check if it is already in the table; if not, we add it along with all the possible own actions (a_j) and neighbours' joint actions (a_{-j}). If the state exists, we proceed as usual.

1. Intial State s
2. Get Action a
3. Get New State s', Reward r
4. Check if s' on Q-table
    - **If not**, add state sub-table (dictionary) with possible combinations of future neighbour's joint actions A'_{-j} as keys and a list filled with 0s as long as A' (possible individual future actions). SET $\max Q_{(s',A)} = 0$
    - **If yes**, GET $\max Q_{(s',A)}$
5. With r and $\max Q_{(s',A)}$, SET $Q_{(s,a)}$
6. $s = s'$, until s terminal.

Figure 5.3.3: *Process for the incremental construction of the table*

With the modification added, the algorithm was tested once more. When applied to two agents, the computing time performance improved in relation to the previous test, with the same positive results for the optimisation. The process now took about half an hour. Nevertheless, when the algorithm was tested on four agents, problems appeared again. Unlike the previous run, the process started, but agents were not properly learning. After inspecting the historic Q-values, the problem became apparent. As the "winning" or "losing" status of a policy is only determined on the very last time step, we need to reach it at least twice.

On a first run, as the performance record is broken and a new optimal solution is found, the maximum reward/penalisation is saved for the "winning" state-action pair. On a second run, this is back propagated to the immediately previous couple via the Bellman equation's $\max Q_{(s)}$ variable. In our case, with four agents, just to transit from t_0 to t_1, there are 625 (5^4) possible action combinations, each resulting in a different environment setting. Moreover, in this experiment, the winning action selection is (at least) on t_3. As a result, we are unlikely to visit the same final state-joint-action combination twice.

The most straightforward strategy to overcome this challenge is to use eligibility traces (λ). Eligibility traces are "a kind of mathematical trick that improves the performance of Temporal Difference methods" (Salloum, 2019). In Q-learning, this is achieved using a "backward view", which communicates to the previous state-joint-action pairs that the policy they are following might be leading to a final "victory" state (one with maximum reward). This implies that we no longer update just one Q-value at a time, like in vanilla Q-learning, but all the previous values of the current policy in formation. This update means there is no need to re-visit a terminal state as the final reward will have a repercussion on all the policy's Q-values that lead to it. As it is possible to derive various other policies from the conforming states of a "winning" policy (and the more we go back, the more alternative branches appear), the weight of the reward has to be higher in the closer steps to the terminal and lower the more we go back. A supplementary vector ($e_{(s,a)}$) stores the state-action pairs visited in the forming policy and counts the time steps passed since each pair was last visited so their new Q-values can be adequately weighted.

The literature describes three forms of implementing λ on Q-learning (Sutton &

Barto, 1998). The first one is Watkin's Q(λ) (Watkins, 1989), in which the counting vector is set to zero every time a non-greedy action is taken (random selection), thus cutting the back-propagating effect. The second is Peng's Q (λ) (Peng, 1993; Peng & Williams, 1994), which does not differentiate exploratory from greedy action selections. Finally, "naïve" Q (λ), which looks just like Watkins's Q (λ), but traces are not set to zero on exploratory actions. Watkins's Q (λ) typical process (for single agent learning) is shown in Figure 5.3.4.

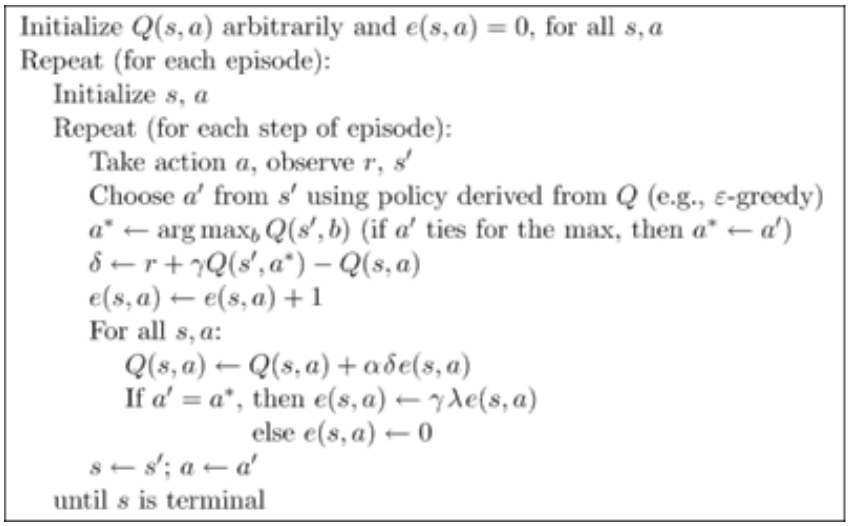


Figure 5.3.4: *Watkins's Q (λ) process for single-agent learning.*
Source: Sutton and Barto (1998)

With a close inspection of the process, it is easy to determine that this will not fit our case study. In our setup, the objective is to find record-breaking solutions. We depend on the exploratory phase to find these policies in this task. Add to this that our major reward/penalisation is located on the latest time step of each episode, and we can easily conclude that cutting trails on exploration is more inconvenient than beneficial. Due to the complexity of implementing Peng's Q (λ), naive Q(λ) seemed the ideal candidate to overcome this inconvenience. Nevertheless, this algorithm was to be adapted to the particularities of our optimisation problem (Box 5.3.5). As the "winning" or "losing" status of a policy is determined only at the latest time step, applying the λ strategy at that last stage is necessary while doing so on the previous ones is inconvenient. Additionally, as breaking a performance record is the most important task, only "winning" policies backpropagate, thus "illuminating" the path to rewarded terminal states only.

The algorithm was tested on a four-agent scenario (Figure 5.3.6), where the

Initialize $Q(s_i,(a_i,a_{-i}))$ arbitrarily and $e(s_i,a_i) = 0$ for all possible s_i, a_i, a_{-i} for T0 and T1 for all agents $i \in I$.

Repeat (for each episode, for each agent i):

Initialize s_i

Repeat (for each step of the episode)

Select a_i solving

$a_i \leftarrow \underset{a_i}{\operatorname{argmax}} \sum_{a_{-i}} \frac{C(s,a_{-i})}{n(s)} Q(s,\langle a_i, a_{-i}\rangle)$ with probability (1 - ε)

a_i = random with probability ε

$e(s_i,a_i) \leftarrow e(s_i,a_i) + 1$

Observe $R(s_i,(a_i,a_{-i}))$, s_i'

Check if s_i' exists in Q-table, if not, add it, along all possible $a_i \in A_i$ and $a_{-i} \in A_{-i}$. Do the same on e-table, but only with $a_i \in A_i$.

Determine V for future state with:

$$V(s') = \max_{a_i} \sum_{a_{-i}} \frac{C(s',a_{-i})}{n(s')} Q(s',\langle a_i', a_{-i}\rangle))$$

If $R(s_i, a_i)$ corresponds to "winning" policy

$\delta \leftarrow (R(s_i,(a_i,a_{-i})) + (\gamma\, V(s_i')) - Q(s_i,(a_i,a_{-i}))$

For all s_i,a_i in $e(s_i,a_i)$

$Q(s_i,(a_i,a_{-i})) \leftarrow Q(s_i,(a_i,a_{-i})) + \alpha\delta e(s_i,a_i)$

$e(s_i,a_i) \leftarrow 0$

If $R(s_i,(a_i,a_{-i}))$ does not correspond to a "winning" policy update current $Q(s_i,(a_i,a_{-i}))$ by solving:

$$Q(s,\langle a_i,a_{-i}\rangle) \leftarrow (1-\alpha)Q(s,\langle a_i,a_{-i}\rangle) + \alpha(R(s,a_i) + \gamma V(s'))$$

If $R(s_i,(a_i,a_{-i}))$ corresponds to a "losing" policy, for all s_i,a_i:

$e(s_i,a_i) \leftarrow 0$

$s \leftarrow s'$

Until s terminal

Figure 5.3.5: *Adapted naive Q(λ) learning for incremental JAL*

agents were isolated from any static neighbours. This was designed to allow faster convergence in a more complex stage. The objective here was to maximise the aggregated performance (sunlight hours during the time evaluated for all the geometries), which means that all agents are selfless and prioritise the common good over the individual. With the rest of the configuration remaining the same as the last test, and after an hour of processing on the same PC, the five hundred episodes were finalised. Because only one reward value is shared among all the agents, we see only one line of accumulated reward per episode. Figure 5.3.7 shows that the reward stabilises on

the highest end by about episode 300. There is a noticeable trend towards a delay in the convergence point as the number of interacting agents increases.

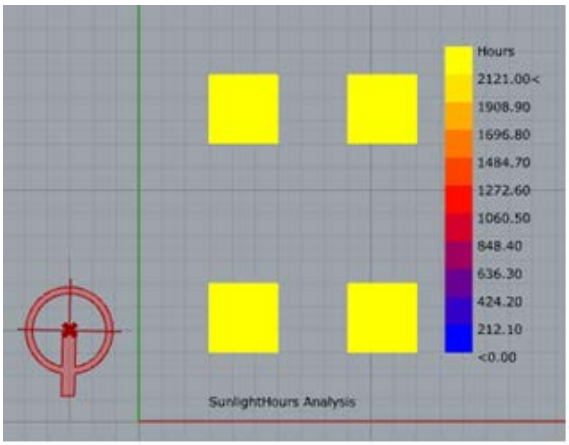


Figure 5.3.6: *Initial state of four interacting agents*

We can also note that, even though in the current test, the ε value starts at the same point and has the same decay as previously, there is more instability in the exploitation phase compared to previous ones. This is probably due to the coordination needed among agents and a magnifying effect on the epsilon value when applied in parallel to four agents. With four agents, we are using the same ε four times before decaying it, so it becomes more likely to have a random action selection even when this number is low. Additionally, arriving at a "winning" terminal state requires coordination on the actions selected by every agent at each time step, which means that a single diverging action from just one of the agents produces sub-optimal results.

Despite these effects, the agents seem to be learning a desired behaviour, and most importantly, they coordinate their actions to achieve their common goal. As can be noticed in Chapter 6, with minimum modification, this algorithm is able to handle tens of agents while interacting with an evaluator that deals with more complex simulations. Nevertheless, some inputs, particularly the epsilon value, needed to be adjusted to achieve convergence in a rational time and within a rational number of episodes.

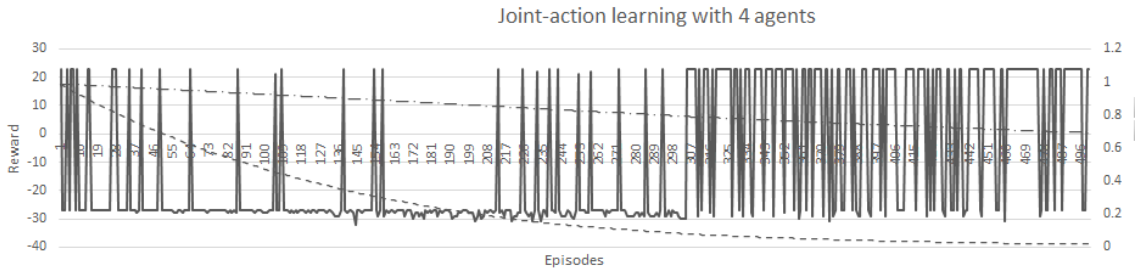


Figure 5.3.7: *Reward accumulation per episode for four interacting agents*

5.4 Conclusions

The objective of the current chapter was to find a Selector method for the optimisation loop described in the methodology. As such, it established our optimisation task at the individual building level as an architectural design optimisation (ADO) problem. Due to its nature, this problem seems to fit better dynamic optimisation methods rather than popular static optimisation ones. By relying on control optimisation literature, this chapter concludes that the most reliable dynamic method to tackle our problem at the individual building scale is Reinforcement Learning (RL). This model-free method avoids the curses of modelling and dimensionality implied in Dynamic Programming (DP). As a machine-learning paradigm, RL also serves as an agent cognition tool while offering scalability due to its straightforward adaptation to a Multi-agent setting. With minimal changes, it allows the interaction of multiple agents in a single environment, thus enabling scenarios of confrontation, cooperation and negotiation. When faced with these scenarios, the current chapter presented coordination as a necessary step to reach Multi-agent modelling. As such, it concludes that implementing the RL Q-learning algorithm with Joint-action learners (JAL) is the most suitable way to provide an optimisation mechanism for the thesis' workflow.

With this in mind, the chapter presents the results from testing several variants of the base algorithm proposed by the literature to a case study similar to the thesis' ultimate objectives. Using a trial-and-error method, the chapter presents a refined version of the algorithm that is adjusted to thesis needs, thus resulting in the likely candidate to be used by the final workflow deployment. The results from testing this algorithm are promising. Still, some adjustments must be made when applied to the final case study, mostly related to the number of interacting agents

in the environment. Particularly relevant is the epsilon value that determines the optimisation process's exploration and exploitation phases and the processing time implied in executing a workflow with tens of agents and several years of incremental development simulation.

References

Bellman, R. (1966, 7). Dynamic programming. *Science*, *153*, 34-37. doi: 10.1126/science.153.3731.34

Claus, C., & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. In *Aaai/iaai*. Retrieved from https://api.semanticscholar.org/CorpusID:871178

Gosavi, A. (2015a). Control optimization with reinforcement learning. In *Simulation-based optimization* (p. 197-268). doi: 10.1007/978-1-4899-7491-4_7

Gosavi, A. (2015b). Control optimization with stochastic dynamic programming. In *Simulation-based optimization* (p. 123-195). doi: 10.1007/978-1-4899-7491-4_6

Gosavi, A. (2015c). Simulation-based optimization: An overview. In *Simulation-based optimization* (p. 29-35). doi: 10.1007/978-1-4899-7491-4_3

Gosavi, A. (2015d). Simulation basics. In *Simulation-based optimization* (p. 13-27). doi: 10.1007/978-1-4899-7491-4_2

Kleiner, A., & Nebel, B. (n.d.). *Introduction to multi-agent programming. 11. learning in multi-agent systems (part b)*. Retrieved from https://gki.informatik.uni-freiburg.de/teaching/ws0809/map/mas_lect11b.pdf ([Presentation])

McNeel, R and Associates. (2014). *Grasshopper. algorithmic modeling for rhino.* ([computer software])

Peng, J. (1993). *Efficient dynamic programming-based learning for control* (Unpublished doctoral dissertation). Northeastern University Boston, MA.

Peng, J., & Williams, R. J. (1994). Incremental multi-step q-learning. In (p. 226-232). Elsevier. doi: 10.1016/B978-1-55860-335-6.50035-0

Poco-Aguilar, S. E. M., Wate, P. S., & Robinson, D. (2022, December). Multi-agent learning of incremental housing development strategies for solar utilisation in peru. In *Proceedings of bso conference 2022: 6th conference of ibpsa-england* (Vol. 6). Bath, UK: IBPSA-England. Retrieved from https://publications.ibpsa.org/conference/paper/?id=bso2022_28

Roudsari, M. S., Pak, M., & Smith, A. (2013). Ladybug: a parametric environmental

plugin for grasshopper to help designers create an environmentally-conscious design.. ([computer software])

Salloum, Z. (2019, 6). *Eligibility traces in reinforcement learning.* Retrieved from https://towardsdatascience.com/eligibility-traces-in-reinforcement-learning-a6b458c019d6

Sutton, R. S., & Barto, A. G. (1998). Eligibility traces. In *Reinforcement learning: An introduction* (1st ed.). MIT Press. Retrieved from http://incompleteideas.net/book/first/ebook/node72.html

Watkins, C. J. C. H. (1989, 5). Learning from delayed rewards.

Watkins, C. J. C. H., & Dayan, P. (1992, 5). Q-learning. *Machine Learning*, *8*, 279-292. doi: 10.1007/BF00992698

Wortmann, T., Cichocka, J., & Waibel, C. (2022, 3). Simulation-based optimization in architecture and building engineering—results from an international user survey in practice and research. *Energy and Buildings*, *259*, 111863. doi: 10.1016/j.enbuild.2022.111863

Wortmann, T., & Nannicini, G. (2017). Introduction to architectural design optimization. *City Networks: Collaboration and Planning for Health and Sustainability*, 259–278.

Chapter 6

Workflow Execution

The purpose of this chapter is to demonstrate the execution capacity of the workflow presented at the beginning of this thesis and to introduce the generated outcomes as proof of principle that passive building design, in general, and the optimisation of building envelope geometries, in particular, can be used in the context of incremental residential development in the global south to achieve increased thermal comfort with energy efficiency at individual and aggregate levels. To achieve this, the workflow components, so far tested as individual entities, needed to become integrated.

As such, this chapter begins by presenting the latest adjustments needed to integrate the independent components into one system. This includes adaptations to the Optimisation Environment presented in Chapter 5 and the digital implementation of the residential incremental development model presented in Chapter 1. These two allow reading the data generated by the Socio-economic model presented in Chapter 3 and transforming it into triggers that allow the optimisation of tens of interacting building envelopes in a residential incremental development setting. After bringing up these adaptations, the chapter continues by presenting the case studies from where EnergyPlus weather files (.epw) are taken and the scenarios posed to test the workflow. As the results are presented, the chapter is able to draw conclusions about the impact of applying optimisation methods to the decision-making process of our agents and on the possibility of implementing envelope geometry optimisation processes in the case of incremental residential development.

6.1 Latest adjustments

So far, this thesis has presented the theoretical arrangement of a workflow of models designed to simulate incremental development and optimise builder-dwellers' location decisions, bearing in mind energy efficiency and thermal comfort. Each of the components of the arrangement has been presented in its chapter, with only the necessary interactions among them to test their independent functionality. Even though the socio-economic model has been capable of delivering its outcomes so far, the optimisation loop has demonstrated a limited capacity to deal with multiple agents, remaining to test its theoretical capabilities beyond local neighbour interaction. As such, this section presents the latest updates to those components that make possible the functioning of the workflow with multiple interacting agents inside a neighbourhood. Additionally, in this chapter, the optimisation loop replaces the Building Energy Modelling (BEM) software used for the test in Chapter 5 (Ladybug, M. S. Roudsari, Pak, & Smith, 2013) with the one selected at the beginning of this thesis (Honeybee/EnergyPlus, M. Roudsari et al., 2023; The National Renewable Energy Laboratory, 2017).

The updates made to the workflow can be classified into three groups. The first one is concerned with the variations on the optimisation problem as presented in Chapter 5; the second one is concerned with the additions and modifications made to the Optimisation Environment presented in Chapter 4, while the last one is concerned with the integration of the model for residential incremental development presented in Chapter 1 (Figure 1.1.1) into a digital platform.

6.1.1 Optimisation Problem

The optimisation problem has some variations in regards to how it was presented in chapter 5:

- The objective function keeps aligned with the imputed agents' rationally bounded behaviour but now, the additional information shared with the agents due to the universal policy choice is the energy used to keep the enclosed space within a "healthy" indoor temperature threshold. Therefore, the imputed behaviour responds to an explicit minimisation of the predicted operational en-

ergy costs coming from the agent's expansion location decision as agents now aim to directly minimise operational energy use.

- The variables remain the same (the building envelope geometry and the performance values shared with the agents).
- Possible solutions in the search space remain related to the buildable space (within lot and height limitations) filling, but now they are in the order of tens of thousands, as we deal with tens of agents in a neighbourhood.

6.1.2 Optimisation Environment

The optimisation environment was altered in its two components: the Geometry Generation (GG) script and the development environment. The earlier required the creation of an additional class that encompasses multiple blocks within it so that a neighbourhood can be created and the learning process of all its components can be managed. The latter, meanwhile required to increase the time execution efficiency so that multiple interacting agents can have their learning processes within a feasible time period. This includes managing the cloud environment to allow parallel energy simulations and the integration of EnergyPlus, which has, so far, not been tested.

Creating neighbourhoods in the GG script.

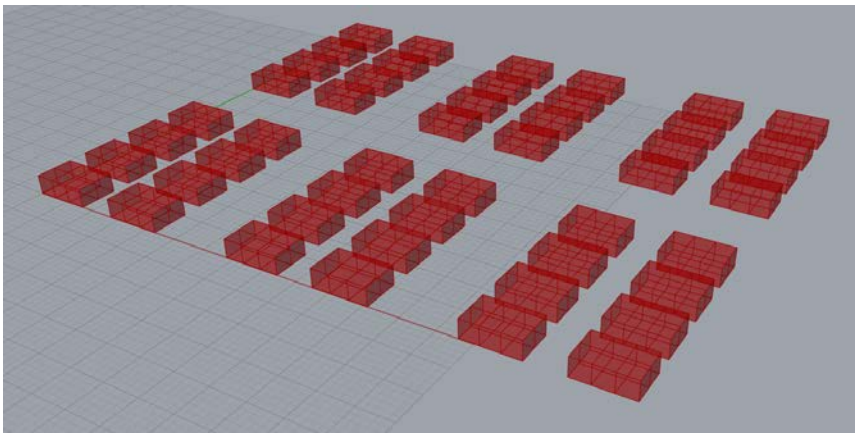


Figure 6.1.1: *Neighbourhood at its initial state created with settings from Appendix 13 seen in Rhino 3D*

Chapter 5 justifies the selection of reinforcement learning as an ideal dynamic optimisation method for the workflow and tests its capabilities for single and multi-

agent optimisation. Nevertheless, the tests on the optimisation loop were done on only up to four contiguous agents. Even when that was enough as proof of principle, further modifications were needed on the GG script and the learning algorithm to optimise several non-contiguous agents. This first section deals with the necessary changes made to the GG script to allow this.

The GG script was limited as it initially only considered two classes in a composition relationship: the *agent* and the *block*. Although generating a block with as many agents as required would be possible, that would not look anywhere close to a neighbourhood. As such, a new class was needed, one that allowed separating blocks by streets. This new class, named "*Neighbourhood*", enables the creation of multiple blocks separated by a given distance, representing the public space in between. The block class was also modified to get closer to the actual aspect of basic module developments in Peru. In Peru, most modern residential buildings have frontal and posterior setbacks. As such, blocks need to include those parameters as inputs. To make this possible, there is now a change of terminology; what an agent used to understand as the area of their "lot" now is just the "buildable area". Setbacks are, therefore, separations between "agent" objects within a block and thus can never be occupied by incremental development.

Additionally, the *Agent* class has been supplemented with reinforcement-learning-related attributes and methods. As such, the incremental construction of the Q-tables and supplementary matrices described in Chapter 5 can now be managed by using the attributes of the said class. This also makes it possible to manage the individual learning process at aggregate levels with the *Block* and *Neighbourhood* classes, whose added attributes and methods get and set those of their individual components related to the learning process.

Increased time processing efficiency in the optimisation algorithm

While these modifications allowed a more realistic approach to the geometry, the learning process remained a bottleneck. Although a realistic processing timeframe was achieved even in a fully cooperative approach in the test phase (Chapter 5), repeating the process in an entire neighbourhood with tens of agents demonstrated that just scaling the approach was insufficient. The first problem was that, for

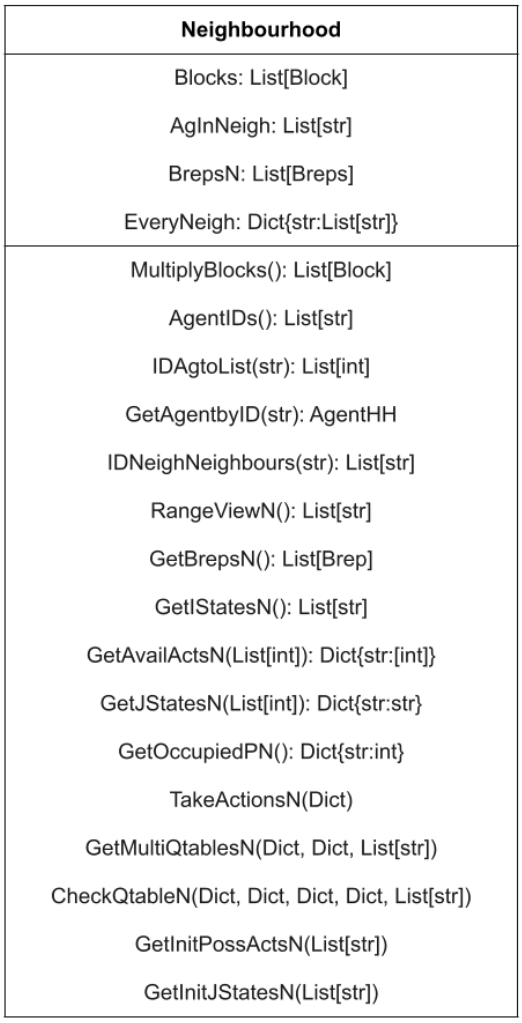


Figure 6.1.2: *Neighbourhood UML class diagram with its most important features*

every agent, every step of every episode, a complete model needed to be sent to the Building Energy Modelling (BEM) software. With each model initially taking up to 5 minutes to process on a local computer, the amount of time it would take to develop 25 years of socio-economic simulation for tens of agents and hundreds of episodes would have been enormous.

Two measures were then taken. On the one hand, it was necessary to find a way to parallelise energy simulations and, on the other, a way to diminish the complexity of the simulated model. To achieve the earlier, the development environment was left behind in favour of the execution environment (See Chapter 4). Pollination (Ladybug Tools LLC, 2023) was chosen as a cloud computing service to execute the energy simulations. This is a paid service offered by the same creators of Ladybug-Tools. It allows sending models in an HBJSON (Honeybee JSON) format to their

servers, processing them in EnergyPlus thanks to Honeybee "recipes". With their hundreds of CPUs, they can rapidly handle the tasks assigned and run several jobs in parallel. Nevertheless, we have to consider that even though this might diminish processing times, the complexity of the task is penalised by increased monetary costs. As such, efficient resource use demands to keep models as simple as possible. Consequently, a filtering process was applied to select only participating agents on every step, episode, and year and generate a Honeybee model only considering their immediate neighbours. Participating agents are those that modify their envelope on the evaluated time-step, while its shading neighbours are those encountered in its view range.

With these two modifications plus a feature to avoid multiple simulations of the same agent in the same state by storing its performance in a Python dictionary, it was possible to diminish processing times radically. Of course, energy simulation is just a part of the optimisation loop. A bottleneck remained in the learning algorithm itself. This was caused as memory ran out due to the high number of agent Q-tables being handled in parallel. Once identified, this issue was rapidly solved by applying the same filtering criteria to send models to the cloud. As such, only participating agents each year create Q-tables, while only the ones participating in a step modify it. As Q-tables include neighbours' current states and possible future actions, these variables must be retrieved from the agent object and its neighbours according to its view range.

Even when this improved processing times, the execution of an aggregate performance approach (as the one used in the last experiment of Chapter 5) within a reasonable timeframe remained unlikely in an entire neighbourhood. Two reasons are behind this:

- The testing phase used Ladybug, which allows sending all geometries, disregarding their location, to one Evaluator component so that an aggregated performance is automatically output. EnergyPlus does not allow the same, as only contiguous *Zones* are recognised as valid building inputs. This implies that to manage a similar objective function as in the latest test, individual results have to be aggregated.

- The filtering process presented above implies that not all agents output their

performances on every time step, and thus, both approaches are mutually exclusive.

As a consequence, new forms of testing cooperation between agents had to emerge.

6.1.3 Residential Incremental Development model

As shown in the general methodology (Figure 1.3.1), to communicate the results from the socio-economic model and trigger the optimisation loop, an intermediary component is needed. The introductory chapter established that this will be the digital implementation of the incremental development model presented at the beginning of this thesis. This section deals with that implementation and the mechanics considered to integrate this component with parts A and B of the methodology. This becomes explained in two sections, one which deals with the coordination of cycles and another that deals with the coordination of measuring units between the socio-economic model and the optimisation loop.

Coordination between socio-economic and optimisation cycles

A requirement to integrate the socio-economic model and the optimisation loop is a mechanism that coordinates their different cycles. While the socio-economic model operates in years, the optimisation loop occurs in episodes and steps. The most straightforward way to achieve this coordination is by considering years as an outer loop so that for every year, there is a learning process composed of episodes and steps. In a Multi-agent setting, within steps, there is an inner iteration of agents so that each one can act on its Q-table. This implies that the learning loop must find a satisficing solution at the end of each "socio-economic year," which must be reinstated the following one as the initial state for a new learning process. To achieve this, two adaptations were implemented. The first allows for expediting the learning process by implementing an auto-stop mechanism that acknowledges when the learning loop has found a stable solution. A second one selects and saves the satisficing solution from a given learning process and reinstates it at the beginning of the next.

The auto-stop mechanism (Appendix 8) works with a fraction of the total maximum episodes allowed per learning process as a user's input. This determines the number of latest episodes the algorithm must look at to count "winning" episodes. If, for example, the maximum allowed number of episodes is 200, and the users' input for the auto-stop mechanism is 0.1, the algorithm checks the latest 20 episodes. Then, it counts how many of these are "winning". As in our optimisation problem, the objective performance is not fixed, but moves as new record performances are found; the algorithm considers an episode as "winning" when its performance is within a tolerance value of the most recent objective. This tolerance has to be the same used to reward the solution within the learning process. Additionally, the auto-stop algorithm accepts a second input, the expected fraction of successful episodes, to determine a complete learning process. In the previous example, an input of 0.7 will mean that 70% or more of the latest 20 episodes will need to be "winning" episodes for the learning process to stop.

This simple scheme complicates when there are several agents in the learning process. In a competitive scenario, it is highly likely that on the same episode, some agents "win" and others "lose". Even more, the learning process may stabilise with a minority of agents having been able to find a reward-worthy solution. This is peculiar to those cases in which agents are adjacent and do not have much space for negotiation (tolerance from the goal, discussed later in the chapter), so a win-win situation is unlikely. To manage this, each evaluated episode receives an average success rate, considering the "winning" or "losing" states of all agents participating in that learning process. This figure is used to get an aggregated average success rate for all the episodes evaluated. The second input of the auto-stop algorithm serves as a goal so that the learning process stops when the aggregate success rate becomes equal to or more than the input. As nothing ensures that the latest episode executed before stopping is the best solution found, the satisficing set of states reinstated as initial the following year is taken from the evaluated episode with the best average performance. In such a way, it is possible to make sure that the initial states of every further year are the ones that benefit the most agents, even in a competitive mode.

As stabilisation in a non-compliant episode success rate is still possible, an al-

ternative exit strategy is taken. In this second option, the learning stops if all the evaluated episodes have the same average success rate. This indicates that the learning algorithm cannot achieve a better-performing solution; thus, processing further episodes would waste time and resources. This constraint also makes clear that we are dealing with a satisficing problem even when using an optimisation method.

Translation of socio-economic units to spatial units

From a certain perspective, the main task of the intermediary component in the methodological setup is to transform the units used in the socio-economic model (Part A) into the ones used in the optimisation loop (Part B). Additionally, considering that this intermediary component determines the timing of expansion, it could also be considered a micro-investment decision model. Micro as the monetary amounts are just enough to build an extension rather than to acquire a finished house. In this sense, this section will describe the implementation of the residential incremental development model exposed in Chapter 1 as both a unit translator and an investment decision model for its implementation in a digital setting.

Up to now, our socio-economic model has output the number of household members and the yearly financial balances of households, which are available for every simulation year. At the other end, the optimisation loop, through the geometry generation script, operates in terms of modular rooms, which compose dwellings. There, the incremental process consists of adding modular rooms each time this is required. Therefore, the challenge lies in turning people and financial means from the socio-economic model into time of requirement (or triggering moments) and modular rooms.

A key concept to allow this translation is the concept of "need". As presented in Chapter 3, the present research understands the "need" expressed in Figure 1.1.1 in terms of the statistical concept of residential "overcrowding". As that section only offered a general approach to that concept, this chapter will standardise it for modelling purposes. In Peru, overcrowding is defined as a situation in which the permanent inhabitants of a dwelling are three or more for every room that can be used as a bedroom (INEI, 2015). This includes all the rooms in a house except those used as kitchens, toilets, passageways or garages. This definition comes from

UN-HABITAT standards (United Nations Human Settlements Programme (UN-Habitat), 2021) and is widely applied in the global south.

Other region's definitions of this term vary substantially. In the US, for example (where their definition of potential bedroom excludes toilets, balconies, porches, foyers, corridors and half rooms), crowded housing units are those with more than one person per room. In contrast, severely crowded housing units are those with more than 1.5 people per room (United States Census Bureau, 2000). In the European Union, the definition of overcrowding depends on the living arrangement of the household. An overcrowded household does not have at its disposal a minimum number of rooms equal to one room for the household, one room per couple in the household; one room for each single person aged 18 or more; one room per pair of single people of the same gender between 12 and 17 years of age; one room for each single person between 12 and 17 years of age and not included in the previous category; and one room per pair of children under 12 years of age (EUROSTAT, 2021). In the UK, two different concepts of overcrowding exist, one used as a legal term and the other as a statistical definition. The latter, defined by the English Housing Survey, use the bedroom standard indicator, which calculates a standard number of bedrooms for each household by age/sex/marital status composition and the members' relationship. From there, Households are considered overcrowded if they have fewer bedrooms available than the notional number needed (Housing and Communities Department for Levelling Up, 2019).

In general, we notice that dwelling size is usually measured by the number of rooms usable as bedrooms to determine overcrowding. Wider differences arise on how inhabitants are related to this size, with some countries limiting the number of people per bedroom according to age, gender and kinship, and others simply aggregating the number of household members without further inquiring about their characteristics and relationships. Given this information, we could conclude that overcrowding within a household is a measure of comfort and that, in general terms, any relation of rooms/people below 0.33 implies an unacceptable standard of living. Ratios between 0.33 and 1 could be acceptable depending on the arrangement of people in bedrooms and their characteristics, and any ratio above 1 allows extra rooms for additional specialised activities. The present model uses these ratios as

determinants to define the need for expanding a dwelling. As such, and to push for a quality of housing that goes above the bare minimum, any dwelling with a ratio of rooms/people of less than 0.5 is considered in "urgent need" of expansion; one with a ratio between 0.5 and 1 is considered in "need", and any above 1 to have "no need" at all.

This concept allows sorting out the issue of "need" by relating people and modular rooms. Nevertheless, the financial capacity variable remains in the air. This input coming from the socio-economic model allows for determining who would actually build. The most straightforward way to introduce this variable is by finding a correspondence between the modular room units and the available financial means. This is done using the direct construction costs of a room unit with the same materials and technology typically used for incremental development in the base year of the social simulation (2015). This was achieved by projecting the cost of an academically documented construction budget for 2009 (Cardenas Vargas, 2013) in the region of Puno, in the Highlands of Peru (See Appendix 9). The micro-investment model updates this equivalence every year, considering the inflation registered by yearly Peruvian official construction values (See Appendix 10). With this transformation, agents in the "urgent need" and "need" categories are able to check whether they can afford the expansion and, if they do, expand. Agents in the "no need" category can build if their accumulated savings are well above the cost of building a room unit. This is determined by another input for the model, which indicates the fraction of the accumulated savings to be compared with the cost of a room unit. If this fraction of their savings equals or exceeds the room cost, they will build one additional room even when they do not need it. This aims to reflect construction activity to achieve increased comfort rather than solely a way to satisfy a need.

Finally, by implementing a loan mechanism, it is possible to increase the affordability of certain agents. In the case of the current model, only agents in "urgent need" can access a loan. If their accumulated savings are insufficient to cover the investment needed to satisfy their need, agents check if they can ask for a loan that covers the difference. For this, there is a "loan capacity" input, which indicates the maximum loan a household could take, expressed in a fraction or number of times their current yearly income. If the maximum loan covers the investment needed,

the agent takes only the required amount, and the figure becomes added to their accumulated debt. Every year that passes in the simulation, a fixed interest rate is added to this latter variable. Agents meanwhile try to pay their debts as soon as possible. If, in any further year, they encounter a positive balance, they use as much as they can to decrease their accumulated debts. The maximum amount they can invest in repaying their debts or on building an extension to their dwelling is set by an input that determines the fraction of their accumulated savings usable for these purposes.

6.2 Case study locations and comparative scenarios

The following section presents the locations selected as case studies and the formulated comparative scenarios. Location selection is based on relevant variables for the present research, including the availability of Energy Plus Weather files (EPW) for energy simulation, the existence of basic module developments in the area and climatic diversity between locations. The posed scenarios, meanwhile, respond to the possibility of testing diverse agent relationships in a multi-agent system and the functionality of the workflow as a tool to devise the relative impact of incremental extension decision-making on potential energy use and thermal comfort by using diverging objective functions.

6.2.1 Location selection

Three locations in Peru were selected as case studies to test the workflow. Their selection responds to climate diversity, TechoPropio's "New dwelling acquisition" (AVN) registered projects and the availability of EPW files to be used in the energy simulation (See Appendix 11). The cities of Piura and Juliaca and the region of San Martin were chosen. Each corresponds to one of the main natural regions of Peru (desertic coast, highlands and eastern rainforests) and is thus representative of their climatic conditions. In the case of San Martin, the entire region was chosen, as registered AVN projects are distributed among multiple cities of the area. Nevertheless, the region is still represented by a single EPW file, whose data was collected

at the airport of the most populated city in the area, the city of Tarapoto. The other two EPW files contain data taken from their corresponding cities' airports. In all the cases, the airports are located within the urban areas, so the EPW data is representative of the cities' local conditions. Hereunder, each of the locations are presented in more detail.

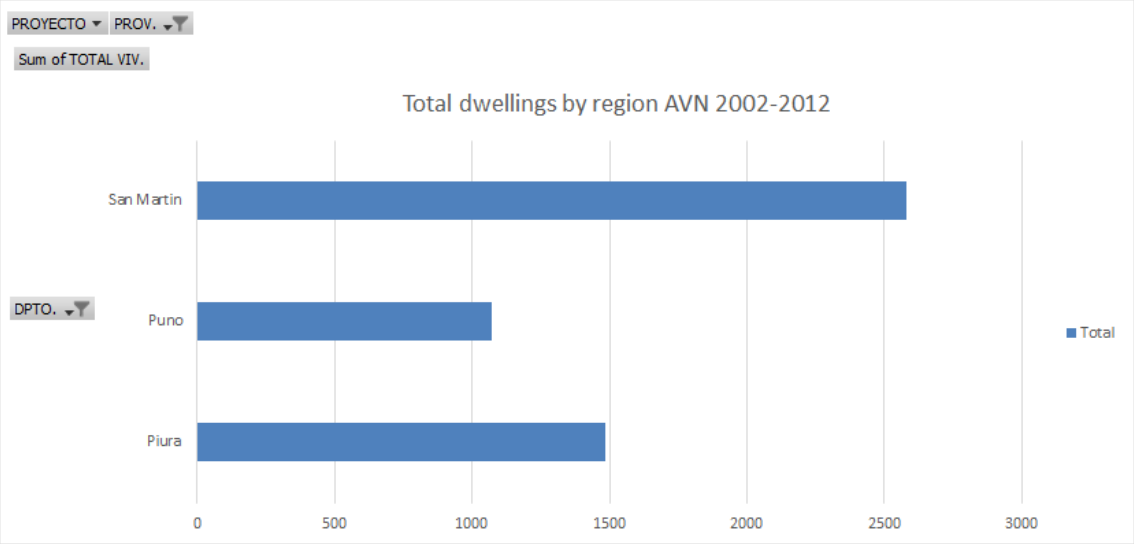


Figure 6.2.1: *Number of registered AVN dwellings in selected regions (2002-2012)*
Source: Mi Vivienda SA (by request)

Piura

The city of Piura, located in river Piura Valley, is the administrative capital of the region of the same name. Due to its location on the western side of the Andes, close to the equatorial line and at low altitude, it has a hot desert type of climate. Despite this, its location in the country's northern tip makes it vulnerable to heavy rains, particularly during El Niño events. Distanced 40km from the sea and surrounded by the Sechura desert plains, Piura usually registers temperatures above 30^{o}C and thus is known as the city of the eternal summer. Mild seasons are present in its location due to the influence of the northward-facing cold Humboldt current, which, while circulating on temperate latitudes, becomes influenced by higher seasonal thermal variations. Figure 6.2.3 shows daily dry bulb temperature variation from the EPW file corresponding to Piura. Figure 6.2.4 shows the total number of registered cooling and heating days in the dataset based on 18^{o}C as the heating setpoint and 24^{o}C as

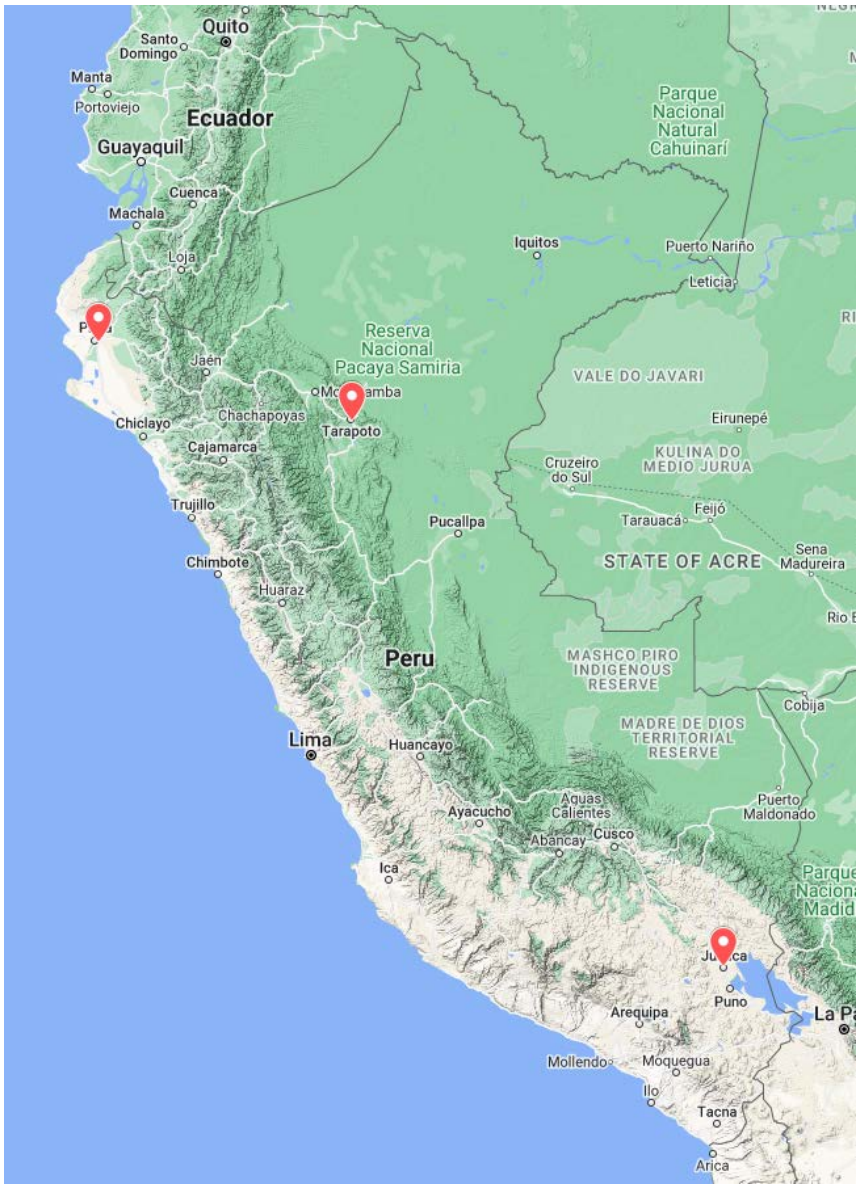


Figure 6.2.2: *Geographical location of selected EPWs*

the cooling setpoint [1].

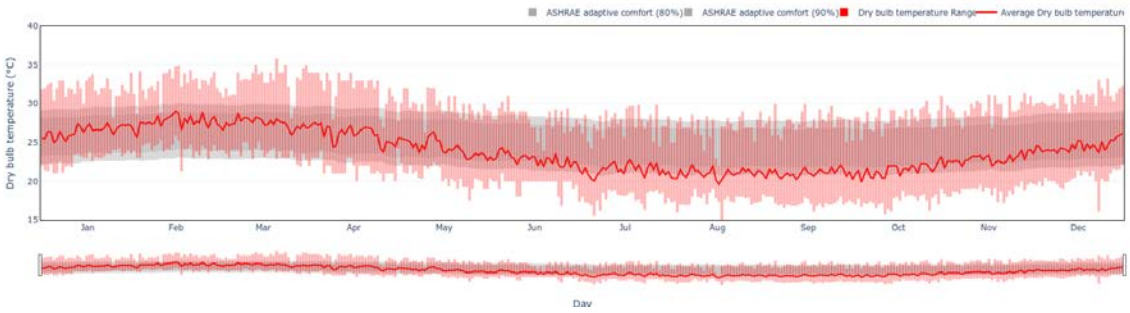


Figure 6.2.3: *Year Dry Bulb temperature from Piura's EPW*
Source: Betti G. (2022)

[1] Figures taken from WHO (1987) and justified later on this document.

Table 6.1: Climatic conditions according to EPW files by location

Location	**Piura Airport**	**Juliaca Airport**	**Tarapoto Airport**
Coordinates	80.616W 5.206S	70.158W 15.467S	76.373W 6.509S
Altitude	35.4 m	3825.8 m	274.3 m
Koppen-geiger Climate classification	**Subtropical desert (BWh)**	**Marine west coast (Cfb)**	**Tropical rainforest (Af)**
Average yearly temperature	23.9^{o}C	9.4^{o}C	25.6^{o}C
Hottest yearly temperature (99%)	33.2^{o}C	20.0^{o}C	35.0^{o}C
Coldest yearly temperature (1%)	17.0^{o}C	-3.2^{o}C	18.6^{o}C
Annual cumulative horizontal solar radiation	**2160.59 kWh/m^2**	**2432.64 kWh/m^2**	**1799.14 kWh/m^2**
Percentage of diffuse horizontal solar radiation	25.90%	24.50%	34.80%

With more than half a million inhabitants, its metropolitan area is the fifth most populated of Peru (INEI, 2022) and an important administrative and commercial centre in the north of the country. Due to this and plenty of available developable land in its surroundings, Piura has been one of the cities most benefited by the AVN programme. Between 2002 and 2012, the province of Piura registered 1487 dwellings under the AVN programme (Figure 6.2.1). The most remarkable incremental developments in that period were "Urbanizacion Santa Margarita" and "Urbanizacion El Sol de Piura", with 789 and 642 registered dwellings, respectively.

San Martin

The region of San Martin is in the northeast of Peru, between the foothills of the Andes and the Amazon rainforest. Connected with both the Pacific coast and the Amazonian port of Yurimaguas through the northern trans-Andean highway, it has grown thanks to commercial, industrial, and service activities brought by this con-

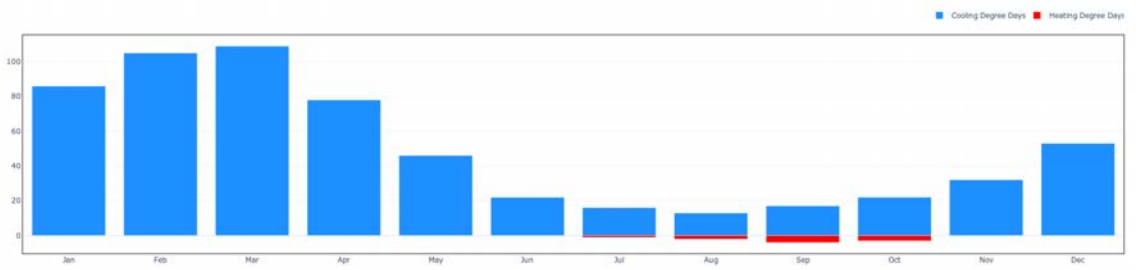


Figure 6.2.4: *Total heating and Cooling days in Piura's EPW*
Source: Betti G. (2022)

nection. Although the region's capital is the city of Moyobamba, the most significant urban agglomeration and the regional airport site is the city of Tarapoto. This city is in the east of San Martin region, in the lowest foothills of the Andes. With about 200 thousand inhabitants in its metropolitan area, it is considered the third largest city in the Peruvian Amazon (INEI, 2022). Tarapoto is an important commercial, educational and tourist centre of the northeast. Other important towns in the region include Juanjui, Lamas, Rioja and Tocache. Due to its local importance and rapid growth, this region has been one of the most benefited by the AVN programmes in the last years. Between 2002 and 2012, the region registered 2579 dwellings under the AVN programme (Figure 6.2.1). The most important basic module developments in this period were Los Sauces de Tarapoto in Tarapoto and Los Jardines de Juanjui in Juanjui, with 380 and 2199 registered dwellings, respectively.

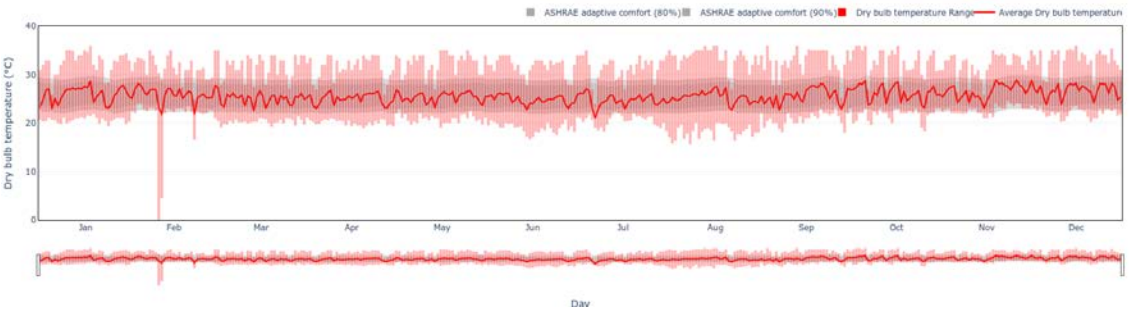


Figure 6.2.5: *Year Dry Bulb temperature from Tarapoto's EPW*
Source: Betti G. (2022)

Tarapoto, as a place with a tropical climate and close to the equator, does not present many thermal variations during the year. Figure 6.2.5 shows how maximum and minimum temperatures remain relatively stable throughout the year. Only a few days in February, the lowest temperature reached 0°C, but this seems to be more of an error in the data-gathering process than an actual event. The stability of

local temperatures can also be seen in Figure 6.2.6, which shows the total number of registered cooling and heating days classified in months based on 18^{o}C as the heating setpoint and 24^{o}C as the cooling setpoint.

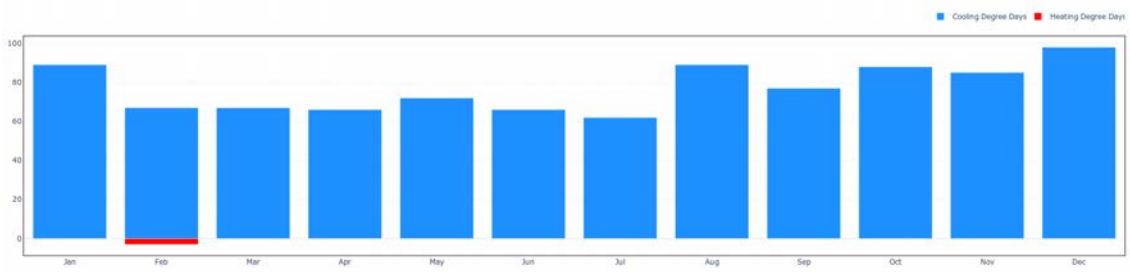


Figure 6.2.6: *Total heating and Cooling days in Tarapoto's EPW*
Source: Betti G. (2022)

Juliaca

Juliaca is a commercial city in southern Peru and the biggest city in the Puno Region. Located in the Peru-Bolivia plateau at more than 3,000 m.a.s.l, the city has more than 300 thousand inhabitants. Due to its location, it has a cold climate characterised by markedly dry and wet seasons. The dry season is linked to the astronomical austral winter. Clear skies and consequential significant daily temperature variations, aggravated by the altitude, characterise it. Higher relative humidities and cloud cover mitigate these effects during the wet season. Figure 6.2.7 shows how minimal temperatures consistently drop below 0^{o}C between May and September while maximum remain relatively stable throughout the year, with a slight but noticeable increase in the spring (September-November). Figure 6.2.8, which shows the total number of heating days considering a setpoint of 18^{o}C, confirms these statements.

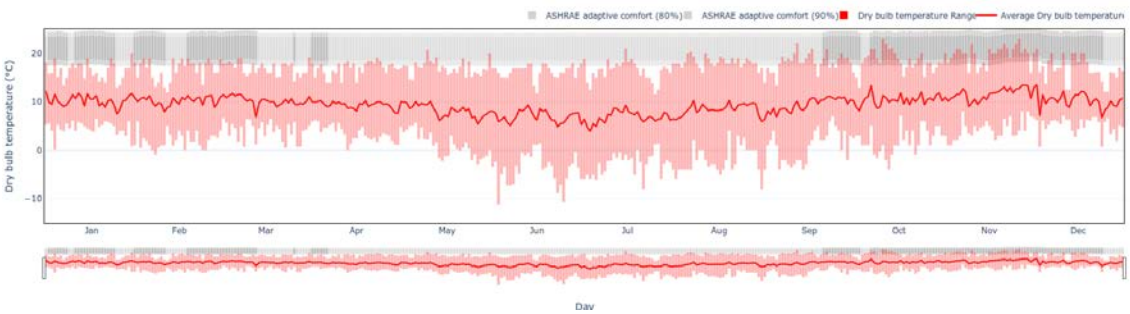


Figure 6.2.7: *Year Dry Bulb temperature from Juliaca's EPW*
Source: Betti G. (2022)

Due to its regional importance, Juliaca has been one of the cities in the highlands most benefited from the AVN programme. Between 2002 and 2012, 1071 dwellings were registered in the province (Figure 6.2.1), with the most significant basic-module developments being Residencial Aeropuerto, Urbanizacion Praderas del Inka and Urbanizacion Santa Monica, with 436, 186 and 214 dwellings respectively.

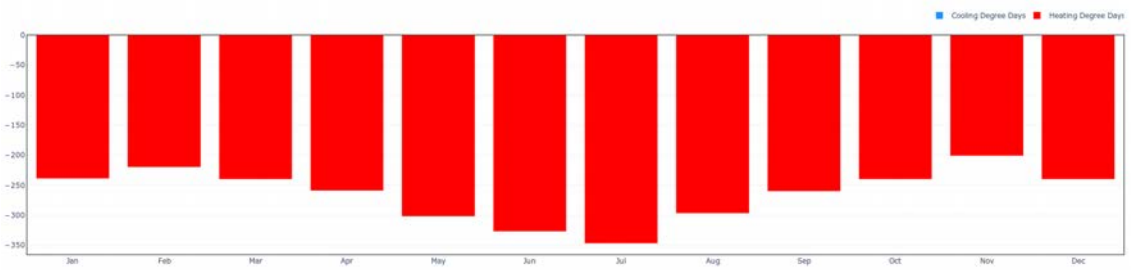


Figure 6.2.8: *Total heating and Cooling days in Juliaca's EPW*
Source: Betti G. (2022)

6.2.2 Scenario definition

Cooperation vs. Competition

Multi-agent systems allow the emergence of complex interrelations between agents. As such, the proposed workflow can also test different agent interaction scenarios. In general terms, these can be classified into three groups:

- Cooperation, when Agents act in a group to solve a common problem,
- Indifference, when Agents do not have any pre-set dispositions towards each other but interact because they use common resources.
- Adversary, when Agents have opposing interests.

How these scenarios were implemented in our workflow depended on the optimisation method selected. As explained in Chapter 5, multi-agent reinforcement learning problems are challenging "because they violate the stationary environment assumptions used by most machine learning systems." (Sen & Sekaran, 1998). Due to this, multiple attempts have been made in the past to enable agent interaction in RL. In the case of our workflow, the solution chosen is based on joint-action learners, as posed by Claus and Boutilier (1998). Joint-action learners can perceive other

agents' actions and thus can maintain a model of their strategies and choose their actions accordingly (Kapetanakis & Kudenko, 2002). Meanwhile, in the alternative approach based on Independent learners, these are unaware of the existence of other agents and can only perceive the reward associated with their joint actions.

In any case, be the agents independent or joint-action learners; they rely on "coordination" to reach individual or common goals. Coordination in this context has been defined as "choosing one's action based on the expectation of others' actions" (Sen & Sekaran, 1998) and "the ability of two or more agents to jointly reach a consensus over which actions to perform in an environment." (Kapetanakis & Kudenko, 2002). In this sense, the discussion between independent and joint-action learner approaches is about the origin of the coordination-enabling information from other agents' actions or rewards. Whatever the pathway, coordination enables cooperative, indifferent, and even adversarial agent-based scenarios. In the case of the latter, for example, agents could benefit from receiving information about what adversaries are doing or planning to do and thus act accordingly (Sen & Sekaran, 1998).

Although coordination enables agent interaction, some assumptions needed to implement it move the model away from depicting real-world situations. For example, in a real-world incremental neighbourhood, it is unlikely that people would find a way to agree on fully coordinated courses of action. Even if they were to find such a space, without the proper legal tools, enforcing agreements might be difficult.

As such, an initially posed cooperative scenario, tested in Chapter 5, reproduces the case of a Cooperative single-stage game (Fudenberg & Levine, 1998), a type of game in which agents have common interests and, as such, they are rewarded based on their joint action with the same reward for all. This initial approach suffered changes resulting from the resource optimisation strategies used to allow the processing of tens of agents (discussed in the first section of this chapter). As a result, agents in a cooperative scenario are no longer rewarded based on the entire neighbourhood's performance but on the performance of the section of the neighbourhood in their individual view range. In this scenario, cooperative and adversarial relations might emerge. The earlier would be when active agents are exclusive in each other's view ranges. When third active parties are included in one's view range but excluded from the other's, adversarial relationships might emerge. Thus, what we now re-

fer to as a cooperative scenario is more of a mixed scenario in which cooperative behaviour is likely but not exclusive.

The competitive scenario, on the other hand, is straightforward. In this case, the reward is based solely on individual performance. Despite this, we cannot consider it a purely competitive scenario, as agents' interests might not be completely antagonistic. For example, if actions 1 and 2 allow Agent A to reach its goal state, but only by selecting action 2 enables its neighbour to reach its own goal state, a possible solution satisfies both parties even if they have individualistic goals. In this case, the joint-action learner method's coordination strategy will privilege the convergence in action 2 rather than on 1. Therefore, if there is space to negotiate, agents with individualistic goals will cooperate to reach the common good.

Tolerance values control this negotiation space. Suppose agents are given more freedom to diverge from the absolute performance record on their history. In that case, they can find multiple goal states and negotiate with their neighbours to find a shared satisfactory solution. Lower tolerances, meanwhile, increase the likelihood of facing zero-sum games and reduce the probability of converging on stable solutions, as opposing agents might take turns to "win" at different episodes. To sum up, the coordination strategy prefers Pareto-dominated solutions even when goals are individualistic. Therefore, what we refer to from now on as a competitive scenario will imply, even if unintended, a cooperative component derived from coordination and tolerance values.

Energy use maximisation vs. minimisation

A second comparison puts together the best and worst possible incremental development scenarios. This is possible by setting diverging objective functions (maximisation vs. minimisation of energy use). This allows us to derive conclusions about the actual effect of the agent's extension choices on the potential energy use of their dwellings. Appendix 6 shows the objective function for a maximising scenario, while Appendix 7 does so for a minimising scenario. The cooperation and competition scenarios use the minimisation objective function, while maximisation and minimisation scenarios use a competitive approach towards agent interaction.

6.3 Results

6.3.1 Results from the micro-investment decision model

As stated earlier in this chapter, a micro-investment model is used to translate the units from the socio-economic model to the ones used in the optimisation loop. To allow the final execution of the workflow, we feed this intermediate model with the results presented in Chapter 3. Thus, the micro-investment modelling results follow a group of 48 agents classified in three socioeconomic levels (SEL) for 25 years. These arrive at the following variables for each agent and each year at the end of the simulation:

- Number of bedrooms per dwelling
- Occupancy ratio (rooms/people)
- Need category
- Accumulated debt
- Accumulated savings

The executed model was limited to adding only a maximum of one room per agent per year. This setting was determined to allow an expedited learning process. Other model settings can be seen in Appendix 12, the most remarkable being that all agents start with a seed dwelling composed of two potential bedrooms. The micro-investment model was implemented in R, and took just a few seconds running from R Studio on a Windows desktop computer implemented with an Intel Xeon CPU with a capacity of 3.50GHz and 16Gb of physical memory to get the desired results. The disaggregated version of these results can be seen in Appendix 5.

Regarding the number of bedrooms per dwelling, as expected, these increase as time passes. Nevertheless, some differences among SELs can be noted. As shown in Figure 6.3.1, agents classified on SEL D, related to the lowest income and larger household sizes, achieve larger dwelling sizes early in the simulation and continue to grow even when the other classes seem to have reached a plateau. As SEL does not differentiate fertility rates, this could be explained by larger initial household

sizes and a consequential higher probability of new household members, which together push for faster and increased incremental development. An apparent relative slowdown in the SEL D figures between years 6 and 16 could be interpreted as a generational change, not occurring (or delayed) in other SELs due to the ages of household members and the household sizes in simulation year 0. It is also important to point out the disparity in the number of agents in each class, which could affect the aggregate averages.

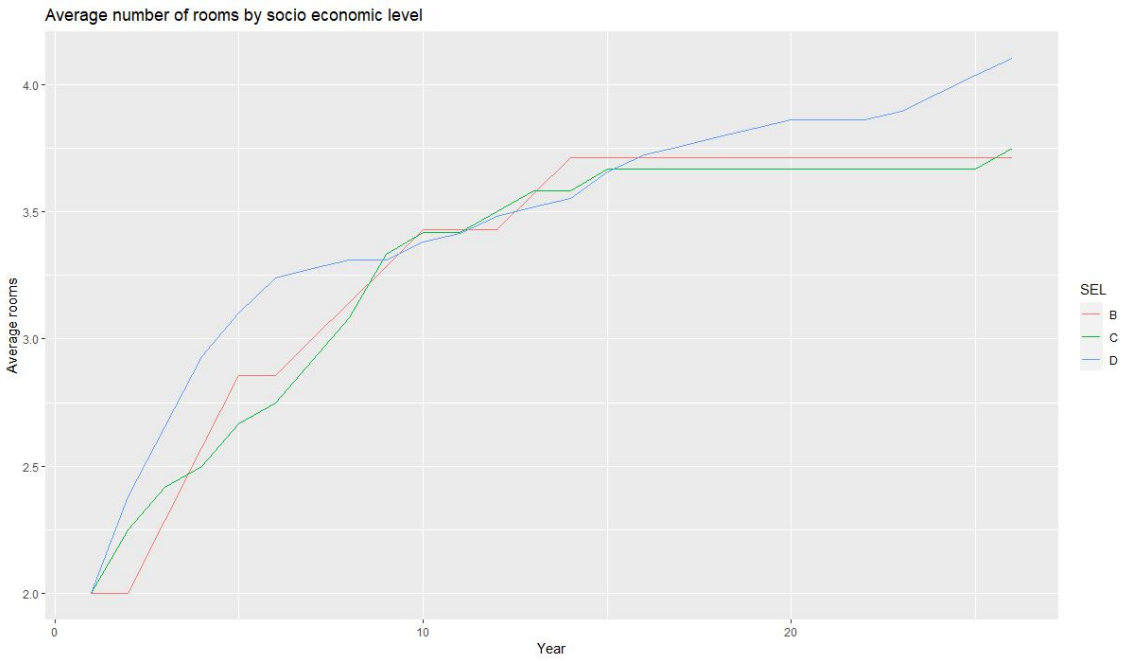


Figure 6.3.1: *Average rooms per SEL for every simulation year*

Figures on the average occupancy ratio per SEL tell us that dwelling growth appears to satisfy the "need" for space in households early in the simulation. Figure 6.3.2 shows how all SEL aggregate values leave the "need" upper threshold by simulation year 6, implying that beyond that point, dwellers are building because of an excess of means rather than due to lack of space. Thus, incrementalism helps to increase comfort rather than just satisfying essential needs. A factor pushing this result might be the initial number of rooms (two), which diminishes the "need" at simulation year 0 and thus provides space to invest in comfort in the remaining years. Another factor might be the optimistic scenario derived from the external macro trendline model used for alignment, which implies a stable and permanent increase in income during the 25 years of the simulation. A remarkable case is that of SEL B, which, despite having the highest income, takes more time to escape the

need area. This could result from smaller household sizes, deriving in the no-need category in early years. Additionally, this category depends solely on their savings for comfort growth as they cannot take a loan, so longer times are needed for expansion. A contributing factor could be that compared to the other SELs, B has higher expenses per capita relative to their income. Thus, yearly balances are lower and accumulated savings take more time to grow.

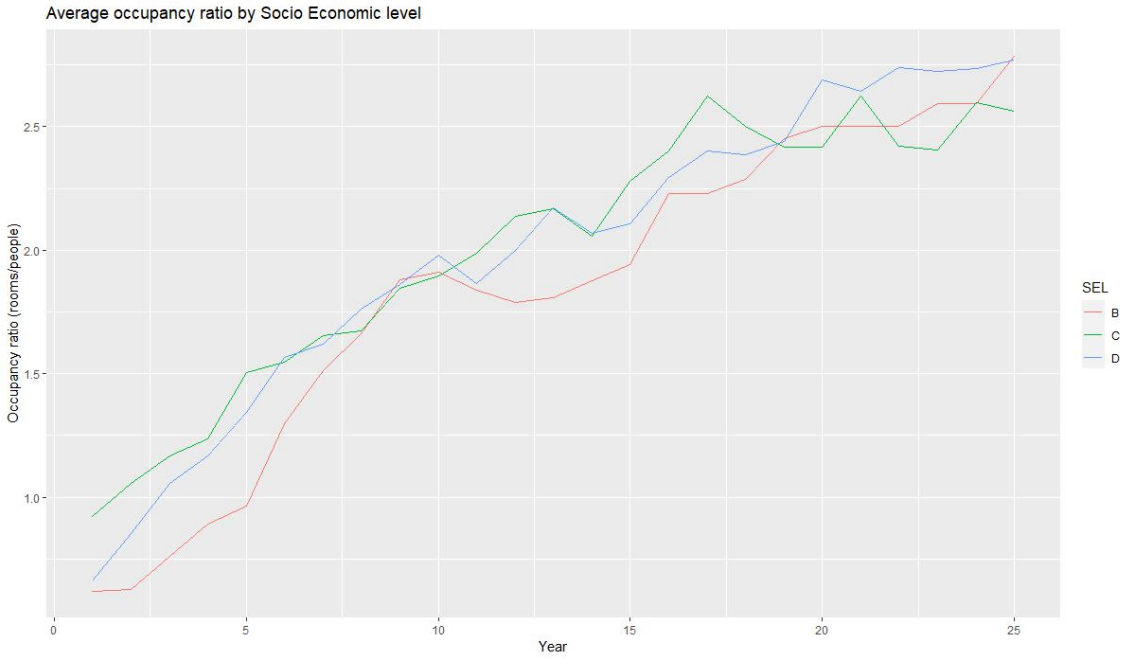


Figure 6.3.2: *Average occupancy ratio (rooms/people) per SEL for every simulation year*

Even if, at an aggregate level, overcrowding seems to be solved, figures at an individual scale show that not every agent in the simulation achieves its goal. Figure 6.3.3 shows how, even if the "No Need" category becomes popular early in the simulation, in the last years, there are still several agents on the "Need" and "Urgent" categories. It is also noticeable that for two years (19 and 20), there have been no agents in the "urgent" category. Considering that there is no active immigration mechanism, this seems to confirm the emergence of a second generation of births.

Regarding debt accumulation, Figure 6.3.4 shows that, as expected, the lowest-earning category (D) relies on loans more than the others. It also demonstrates that repayment is possible despite a relatively high annual interest rate (15%). The optimistic assumption about income evolution on the socio-economic model might play a role here. The sudden fall in the aggregate average for SEL D between

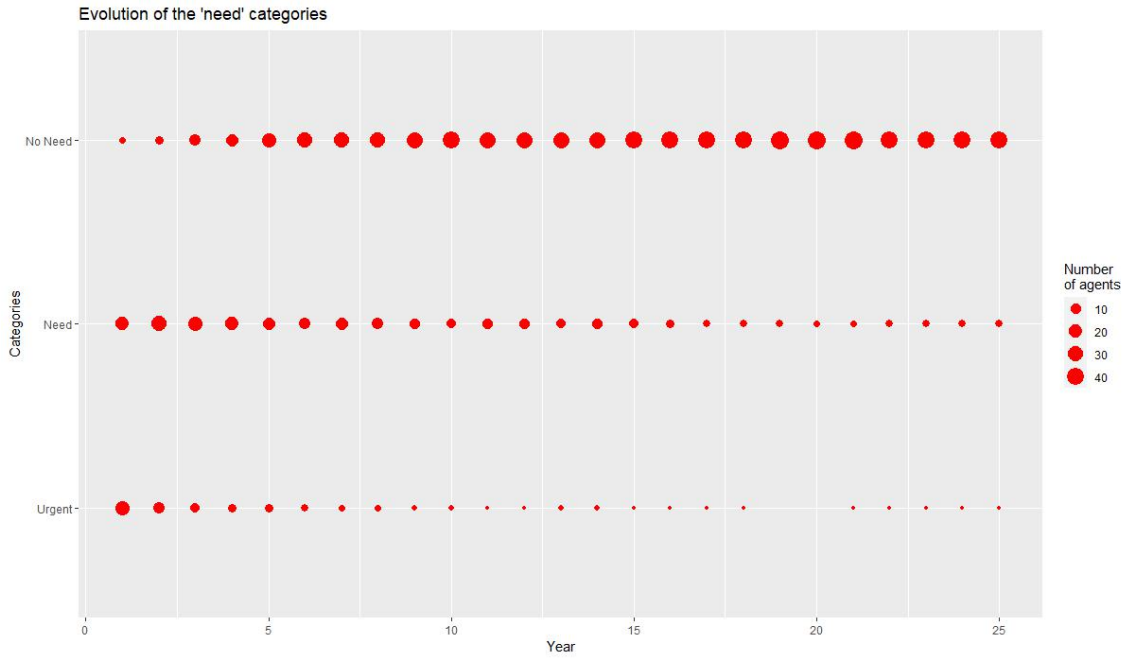


Figure 6.3.3: *Evolution of the "need" categorization of agents*

years 16 and 17 might be explained by the accumulation of debt on one agent, who suddenly and providentially increases their annual savings to repay the accumulated debt from years in one (Agent 39, identified in Appendix 5). SEL B lays on the opposite side of D, as no debt is ever acquired. This could be due to a combination of factors, including higher incomes and smaller household sizes.

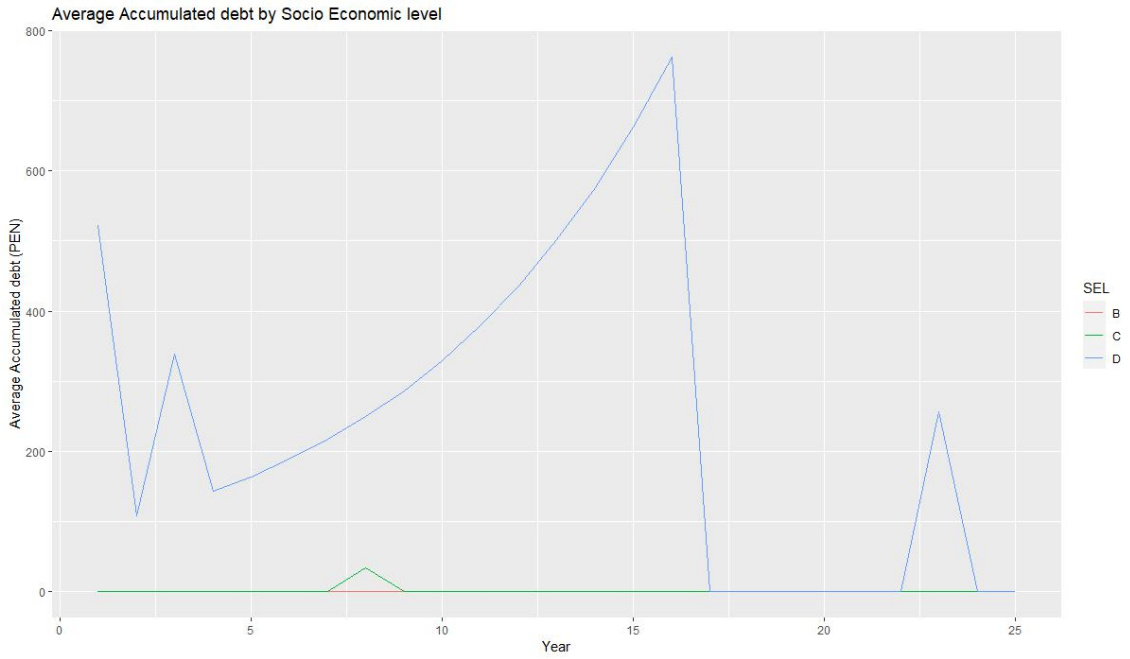


Figure 6.3.4: *Average accumulated debt per SEL for every simulation year*

Figures on accumulated savings by SEL seem to confirm some of the previously posed hypotheses. However, the highest earning category (SEL B) accumulates less savings than the other two. This could be, on the one hand, due to the highest expenses per capita identified in the socioeconomic model and, on the other, due to their reliance on their own resources to carry on with the incremental process. The SEL whose savings grow more is C, a logical consequence of not relying so much on credit to carry out incremental development and the increasingly diverging path between their income per income generator and expenses per capita, identified in the socioeconomic model.

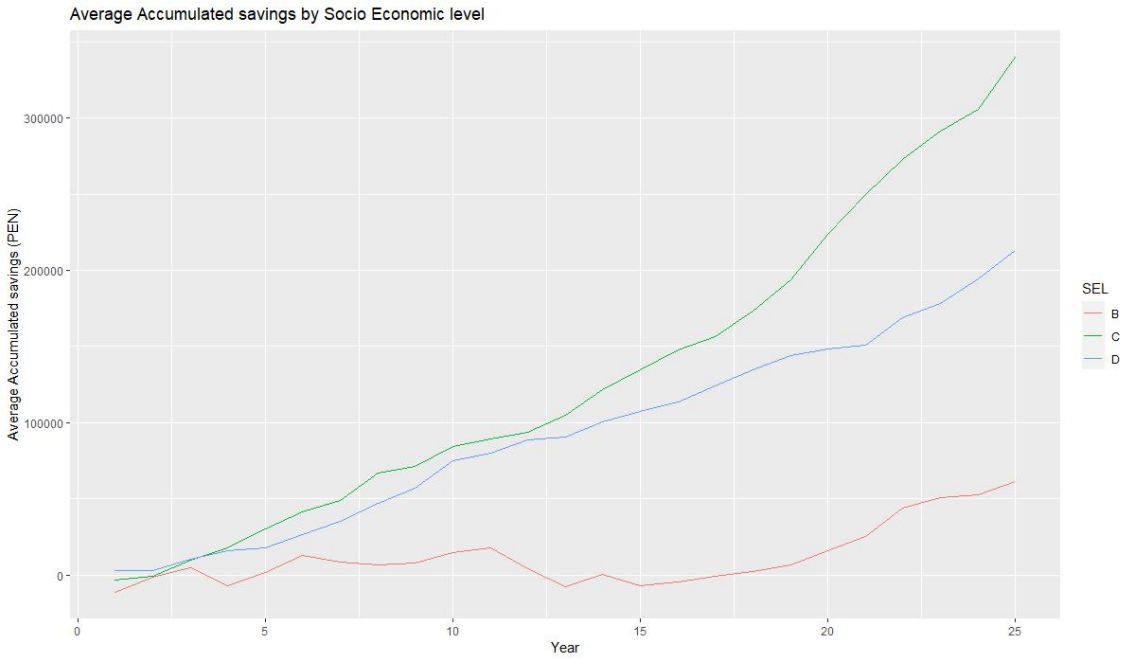


Figure 6.3.5: *Average accumulated savings per SEL for every simulation year*

Now that the output variables of the micro-investment model have been presented, it is possible to arrange these in time and space. Figure 6.3.6 shows the number of active agents each year of the simulation. Since agents are only allowed to build one bedroom per year, it also reflects the number of bedrooms built yearly. There is an evident construction boom in the early years of the simulation, which recedes as years pass. This complements the observations made in Figure 6.3.3, which shows how all SELs leave behind the "urgent need" category in the first six years. Additionally, two years of the simulation (19 and 20) show no activity. This agrees with Figure 6.3.3, which shows that, precisely in those two years, no single agent was classified in the "urgent need" category. The resurgence of construction

activity in the last four years seems to confirm a second generation of dwellers. The implications of these figures for the optimisation loop are that there will be several interacting agents in the early years, which implies longer processing times. As time progresses, the task will be much easier and thus, processing time figures will be reduced.

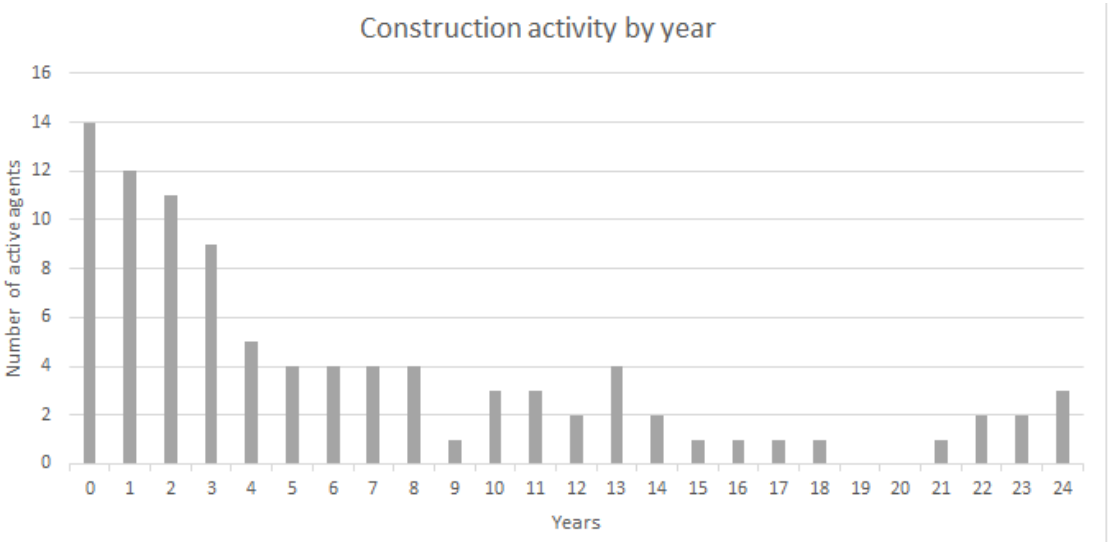
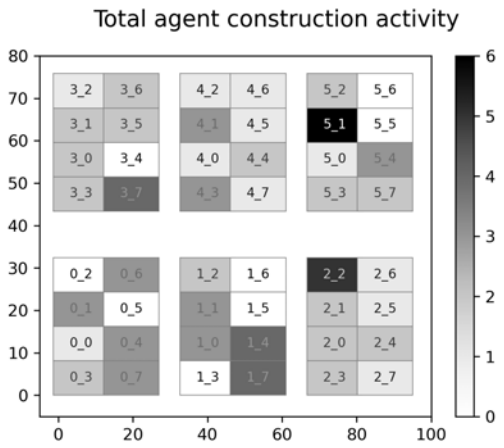


Figure 6.3.6: *Active agents per year*

Placing agents in space gives us more information on the implications of this development pattern. After a "Neighbourhood" object becomes created using the settings shown on Appendix 13, agents simulated on the loan/investment decision model can be assigned to a "lot" in this spatial setting. A simple, orderly process was used to assign locations to abstract agents so that the first agent corresponds to the first lot object created on the procedural geometry generation process. Figure 6.3.7(a) shows the construction activity (years in which expansions happen) of each agent distributed in space for the 25 years of the simulation; Figure 6.3.7(b) shows the construction activity of each agent by identification distributed in time, and Appendix 15 shows the construction activity for each year distributed in space.

Finally, an initial geometrical size was determined to set initial states. This comes in the form of a sequence of actions for each agent that produces an initial spatial setup (Figure 6.3.8). Although the micro-investment model settings determined that the initial size would be two bedrooms, the geometrical model needs to include other spaces in the dwelling which are not bedrooms; as such, the initial geometrical state has 6 modular cubes in size (mimicking the dimensions of real-world seed dwellings

(a) Activity of agents by location



(b) Agents activity by year

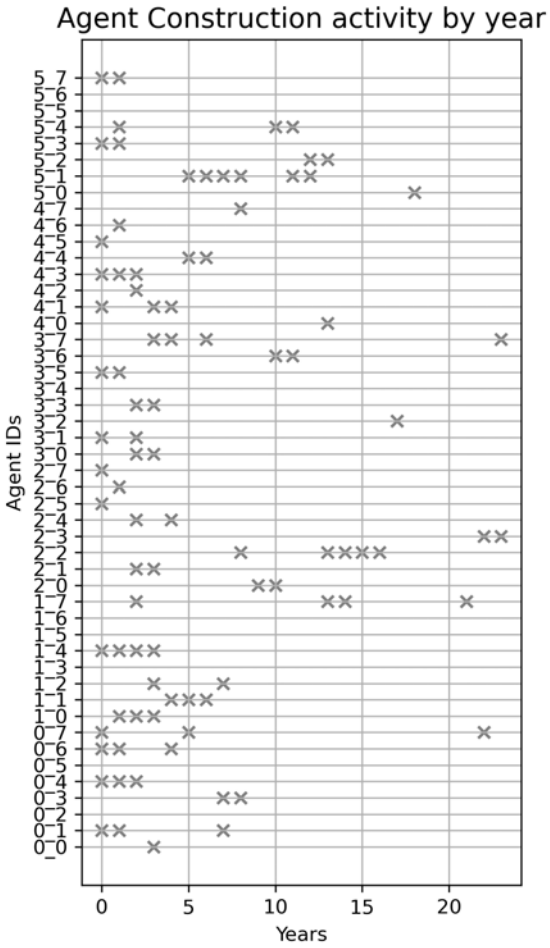


Figure 6.3.7: Agent activity in time and space

and including non-dormitory rooms). Despite this initial consideration, all further extensions are considered likely bedrooms, while other functional spaces are, for now, disregarded for model simplification.

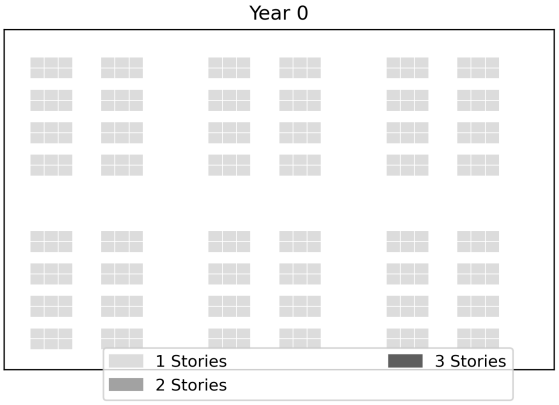


Figure 6.3.8: *Dwellings on their seed state before the learning process*

6.3.2 Optimisation results by location

Once the agents coming from the micro-investment model are in place, it finally becomes possible to execute the optimisation loop. Bearing in mind the Thesis' objectives, the present study uses potential Annual Energy use as a proxy for thermal comfort. This allows to optimise the geometries for the two established goals (energy efficiency and thermal comfort) while keeping a single performance figure as an objective function. As such, Honeybee's "Rooms" are conditioned and use heating and cooling set points (18^oC-24^oC) based on healthy indoor thermal standards (WHO, 1987). Under this approach, lower annual energy use figures imply fewer hours outside of the "healthy" zone and, at the same time, less demand for heating/cooling appliances to achieve the goal temperature interval. The selection of these set point figures responds to an inclusive approach towards building design that bears in mind the well-being of vulnerable groups while keeping simplified universally applicable categories (healthy vs. unhealthy indoor spaces). In any case, this consideration becomes secondary, as in our optimisation exercise, relative differences between solutions are more important than absolute performances. Absolute values will, therefore, remain solely referential, as they do not express the actual energy use of households with commonly used appliances in the real world.

This section reports the results from testing the workflow on the three locations introduced earlier in this chapter. As such, except for the EPW files, all the settings to run the BEM and the Multi-Agent reinforcement learning (MARL) algorithm were kept the same among locations. In the three cases, the objective function was to minimise the operational energy use of the dwellings. The agents, meanwhile, were rewarded based on their individual performance (competitive mode).

The BEM considers dwellings as mono-zone buildings. This is because no internal divisions exist; therefore, every dwelling is evaluated as one Honeybee "Room" or its equivalent in EnergyPlus, one "Zone". Despite this, the "zone" envelopes are formed by several quadrangular surfaces whose dimensions correspond to the module used in the GG script (Figure 6.3.9). Each surface is implemented with a Honeybee Aperture, controlled by an input determining the dimension relationship between the surface and its aperture. Other settings required by the BEM and the MARL algorithm can be found in Appendix 11 and Appendix 14, respectively.

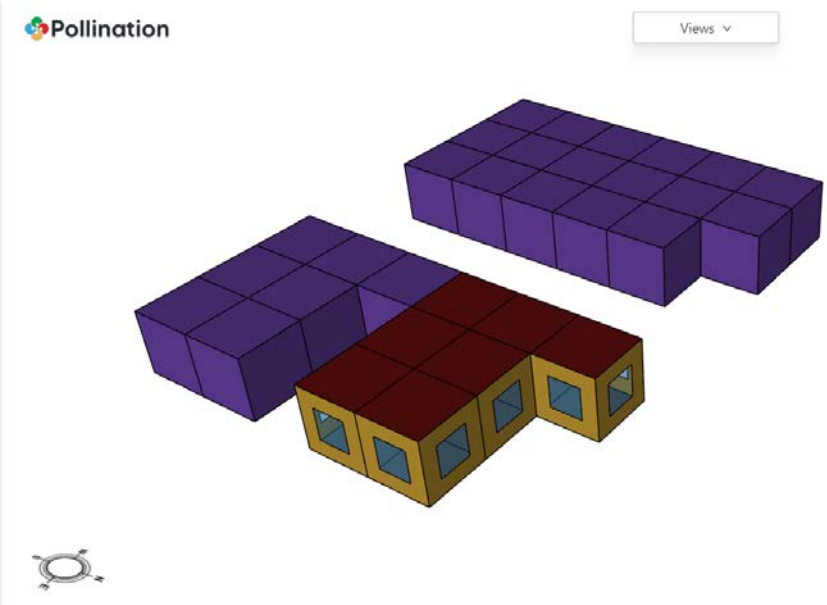


Figure 6.3.9: *Model of an agent and its shading neighbours, as seen from Pollination*
Source: Ladybug Tools LLC (2023)

From the three location scenarios presented, two (Piura and Tarapoto) were run locally, and one (Juliaca) was run on the cloud. This allowed the parallel processing of several locations at once and the comparison of execution performances among platforms. The local run was executed on a Windows desktop computer implemented with an Intel Xeon CPU with a capacity of 3.50GHz and 16Gb of physical memory. The cloud run, meanwhile, executed the energy simulation on the Pollination servers, with its learning process happening in a Windows laptop computer with an Intel Core i5-9400H CPU with 2.50GHz of capacity and an installed memory of 8Gb. Due to the differences in the computing capacity, these executions are compared in terms of the number of episodes taken per year to reach convergence and the average time taken by episode on each simulation year.

Figure 6.3.10 shows the number of episodes it took for each location to reach convergence. Due to the auto-stop algorithm inputs, no scenario will stop before 20 episodes (10% of the maximum allowed episodes). Besides this, it is noticeable that, in all locations in the early years of simulation, there is a higher number of episodes until convergence. This figure eventually stabilises on lower values as time passes. This is a natural consequence of more active agents in the early years, making it more difficult to reach an agreement on a joint-action solution. As all inputs except EPW files remain the same, the only plausible explanation for location differences is

that these agreements are more difficult to reach under certain climatic conditions and intermediate spatial arrangements. The case of Tarapoto is remarkable, which in years 0 and 3 takes considerably more episodes to reach convergence than other locations. In year 15, meanwhile, when there is only one active agent, all locations take the same number of episodes, corresponding to the minimum possible.

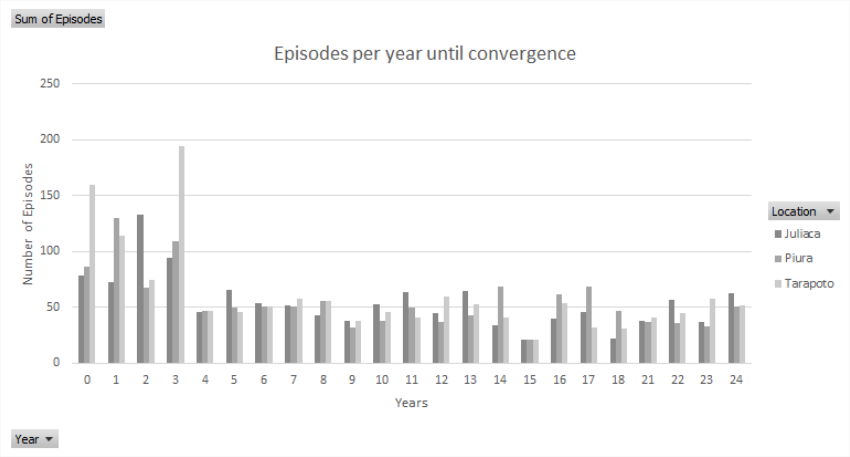


Figure 6.3.10: *Number of episodes until convergence by location*

Figure 6.3.11 shows the average time per episode for the three locations. In the case of Tarapoto, it is noticeable how consistent lower times per episode are. Energy modelling seems to take considerably less time when processing simulations that use that EPW file. When comparing the other two locations, Juliaca tends to have a better time performance in the earliest years. Nevertheless, its advantage is lost as time passes. An explanation for this is that the comparative benefit of processing models on the cloud, derived from its capacity to handle parallel tasks, is lost when there are fewer models to process. This is mainly because time is lost in uploading and downloading data, so with fewer models to process, local execution is more efficient than using the cloud.

As a result, in the current study, overall processing times are higher in the cloud environment. This is because most activity is carried out in the first five years, while in the remaining 20, fewer energy models need to run in parallel. This difference can be noted in Figure 6.3.12, where Juliaca, executed in the cloud, is the worst-performing location regarding execution time. Despite the higher number of episodes in selected years, Tarapoto benefits from lower energy modelling processing times and thus delivers results briefly.

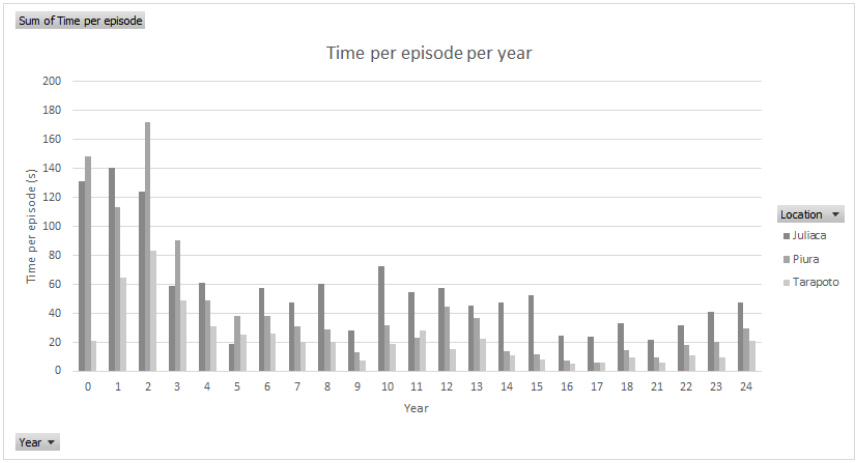


Figure 6.3.11: *Average time per episodes until convergence by location*

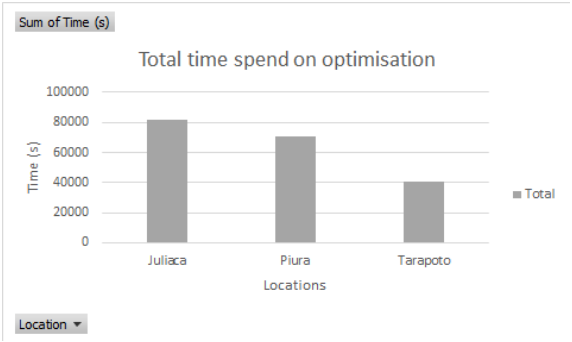


Figure 6.3.12: *Total processing time by location*

At the agent scale, each learning process has its characteristics. Appendix 16, Appendix 20 and Appendix 19, corresponding to Piura, Tarapoto and Juliaca, show the agents' graphs for every learning process of reward accumulation and registered performances. While these are kept for the reader's consideration, the current chapter will discuss the aggregate results in terms of performance, measured in aggregate Energy Use intensity (EUI). As in these cases, the learning process uses individual annual energy use as a performance measure; aggregate EUI is obtained by combining individual energy performances of all agents and dividing that figure by the neighbourhood's total built area. These two figures are captured from the final state of agents at the end of every yearly learning process. To get the total built area, the number of occupied modules per agent is aggregated to a global figure, and then this is multiplied by the floor area of a single module.

Although this is more than enough to compare scenarios and derive plausible conclusions, absolute values are referential; units used as input for lighting and equipment in the energy model are in Watts over the unit of built area (m^2), but

dwellings are mono-zone and do not have internal divisions, so it is expected that the energy model considers solely building footprints as total built areas, ignoring increased areas in upper floors. Although this might affect aggregate EUI values, it does not affect the individual learning process, which depends on relative differences. Additionally, the objective function (Appendix 7) uses total annual energy use as a performance measure, classified according to the number of occupied modules at the evaluated state, and never refers to area units.

Figure 6.3.13 shows relevant differences between locations, with Tarapoto using considerably more energy per square meter than the other two cases. This might result from that location maintaining consistently high temperatures all year round (Figure 6.2.5), unlike the other hot zone, Piura, which shows a temperature pattern that depicts mild seasons (Figure 6.2.3). Juliaca, meanwhile, the only location in a cold area, shows the best performance. This might result from higher solar radiation during colder and drier months, which allows solar heat gains during the day and diminishes the need for heating appliances. In terms of common trends, we can notice that in the three locations, energy use intensity seems to diminish as time passes. A plausible cause could be that the energy model settings remain unaltered despite increased areas. There is no feedback loop to alter, for example, the people density input for Honeybee. Another likely cause is the discrepancy between the built areas considered by the BEM and the ones considered to get the aggregate EUI values.

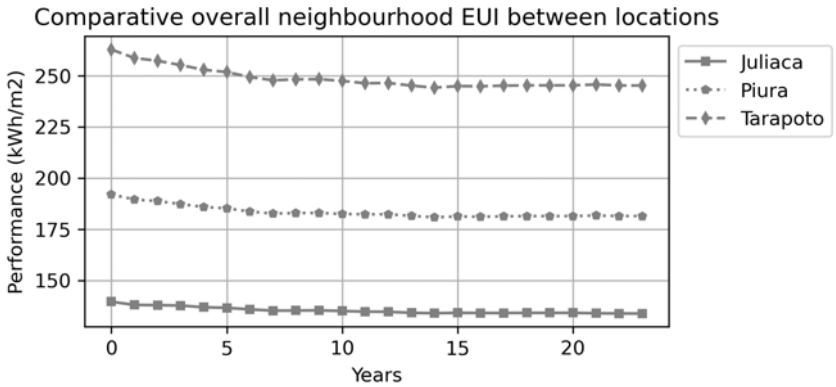


Figure 6.3.13: *Aggregate EUI by location*

Finally, the effect of the climatic conditions on the agents' decision-making process can also be measured by the spatial patterns that emerge. While Appendix 21

shows the spatial distribution of cubic modules for every simulation year in the three locations, this chapter will be limited to analysing the outcomes at the end of the simulation process (Figure 6.3.14). As the objective function is kept the same, there is not much difference among the patterns formed. In general terms, agents try to develop horizontally, with only a few growing two and only one three stories high. In this sense, a small difference exists between hot climates and the cold location (Juliaca). In the latter, there are more modules located on the upper levels compared to the other two. It is also the only location in which a three-storied building appears. This might have to do with the geometries' increased exposure to solar radiation, which may be diminishing the need for heating. Otherwise, developing vertically seems to be less profitable than doing so horizontally. This probably has to do with the lack of internal divisions within dwellings, which increases the energy demand when conditioning multi-storied buildings.

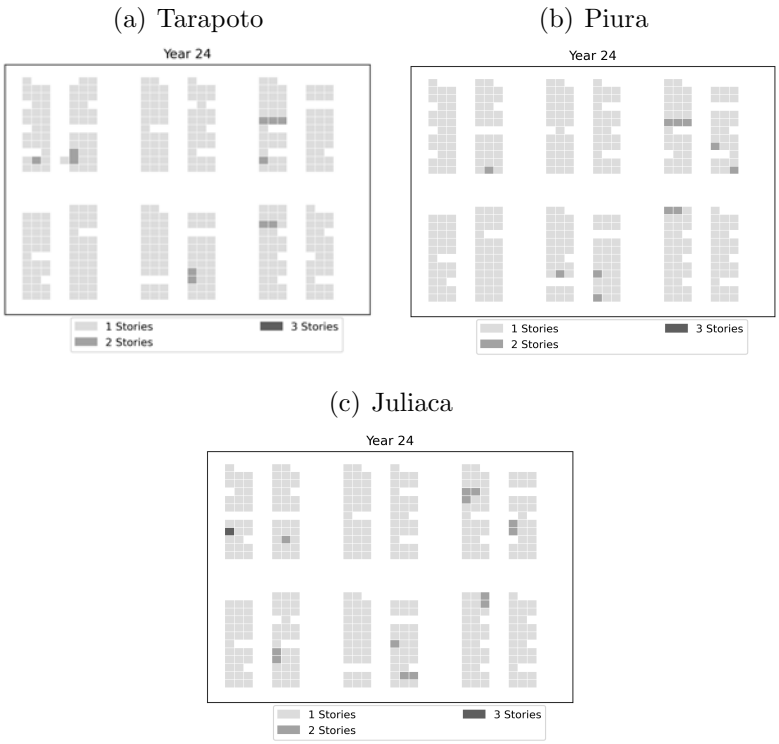


Figure 6.3.14: Spatial pattern comparison by location at final simulation year (competitive, minimisation)

6.3.3 Cooperative vs. competitive

This second scenario implies the addition of a simulation run not previously presented. In this new run, the EPW file of Juliaca is used in a cooperative approach and executed on the cloud. The location responds to the fact that this EPW file was already used in a cloud setting, and as the cooperative approach demands more resources and thus demands the cloud setting, it made sense to keep the site to allow comparison with minimal interferences. As such, the competitive run presented here is the one denominated as "Juliaca" in the previous section.

Figure 6.3.15 indicates the number of episodes it took to converge on each simulation year. Results here are mixed, as only in some years the cooperative learning process takes more time than the competitive one. As in the previous comparison, a common pattern among runs appears in year 15, in which agents take the least possible amount of episodes to reach convergence.

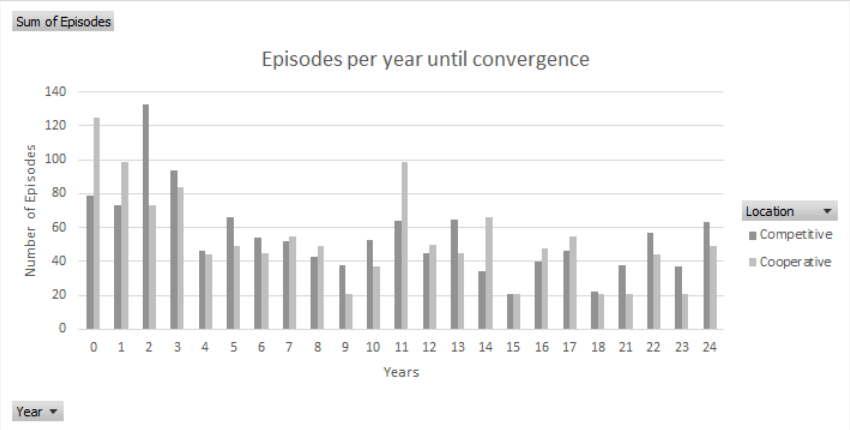


Figure 6.3.15: *Number of episodes until convergence in cooperative and competitive scenarios*

Figure 6.3.16 shows how, in the cooperative scenario, the average time per episode is, with few exceptions, consistently higher than in the competitive scenario. This is in line with the mechanics of the cooperative scenario, which requires sending more models per step (agents' model and its neighbours' models to aggregate performance figures) even when the number of participating agents remains the same as in the competitive one.

As expected, this impacts the overall processing time of the cooperative scenario, which is considerably higher in relation to the competitive one (Figure 6.3.17). As both were executed on the cloud, whether these results will hold if one or both were

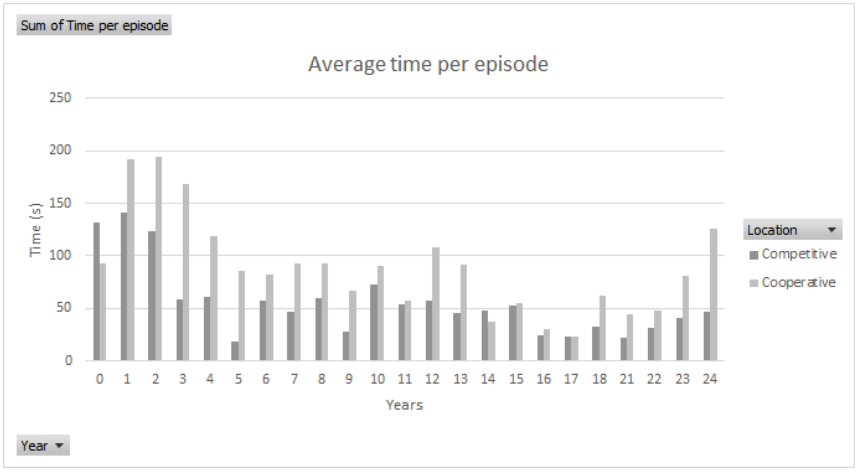


Figure 6.3.16: *Average time per episodes until convergence in cooperative and competitive scenarios*

executed locally remains an open question.

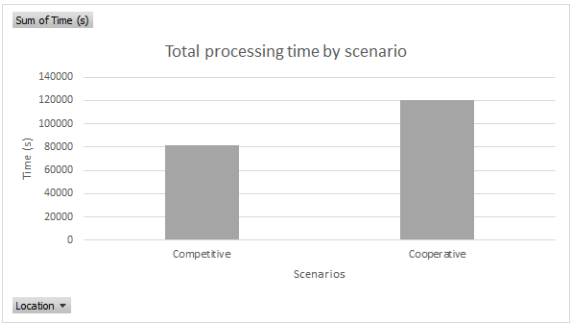


Figure 6.3.17: *Total processing times in cooperative and competitive scenarios*

Again, individual agent learning processes are left for the readers' consideration in Appendix 19 and Appendix 18, and the chapter will carry on by discussing the neighbourhood's aggregate Energy Use Intensity in kWh/m^2. In this sense, the competitive scenario seems to perform consistently better in relation to the cooperative one (Figure 6.3.18). A plausible explanation is that even in the competitive scenario, coordination exists. When negotiation spaces are present (because of tolerance values) and objectives are diverging but not opposed, Pareto-dominated solutions are preferred. As such, when several agents find a satisficing individual performance, they might be pushing further the aggregate performance of the neighbourhood with their individual efforts. Altought in the cooperative scenario might happen the same, there, individually satisficing solutions are diluted on locally aggregated performance values, and thus global aggregates are not pushed as much.

Finally, we carry on with the analysis of emergent spatial patterns. For the

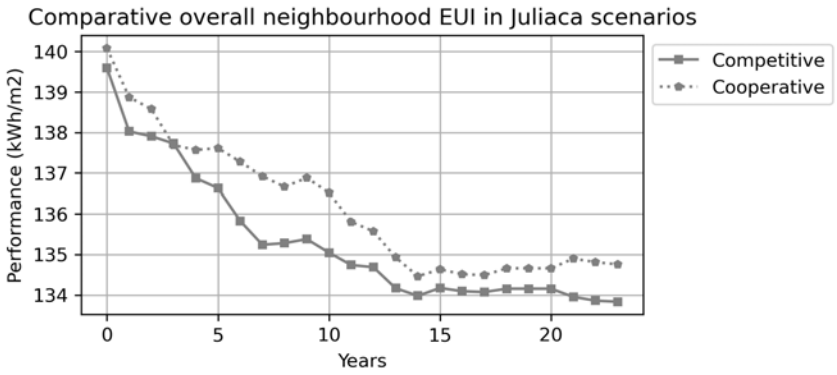


Figure 6.3.18: *Aggregate EUI in cooperative and competitive scenarios*

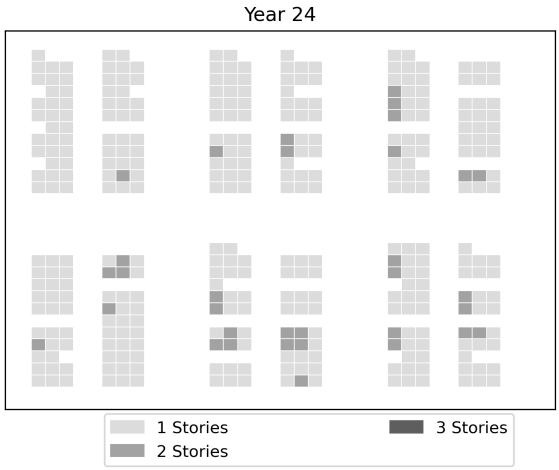
reader's consideration, Appendix 23 shows the spatial outcomes of every year for the two runs here compared. Figure 6.3.19 shows the spatial patterns at their last simulation year. According to these images, we can notice that when compared to agents on the competitive run, those on the cooperative one develop vertically with more frequency. As such, the corresponding graph has significantly more modules on upper stories. As hypothesised earlier, this might have to do with the increased possibility of collecting solar heat gains with vertical development. There must be an additional comparative benefit of taking this approach when aggregating the performances locally, which is otherwise ignored.

6.3.4 Maximisation vs. Minimisation

To carry on with this last scenario, the workflow ran once more to generate a fifth simulation. On this new run, the objective function is replaced, so the algorithm tries to maximise the energy use instead of minimising it. This run uses the EPW file of Piura and is executed locally. This choice was made as the location comparison put Piura's aggregated EUI at an intermediate level. Additionally, it demonstrated that for the task assigned, running locally was more convenient than doing it on the cloud. To minimise inconsistencies in the comparison, the run representing here the *minimisation* objective function corresponds to the location earlier denominated "Piura".

Figure 6.3.20 shows the number of episodes it took on each of these simulation runs to reach convergence according to the standards considered by the auto-stop mechanism. In most learning processes, the *minimisation* run seems to take fewer

(a) Cooperative spatial pattern at final simulation year (Juliaca, minimisation)



(b) Competitive spatial pattern at final simulation year (Juliaca, minimisation)

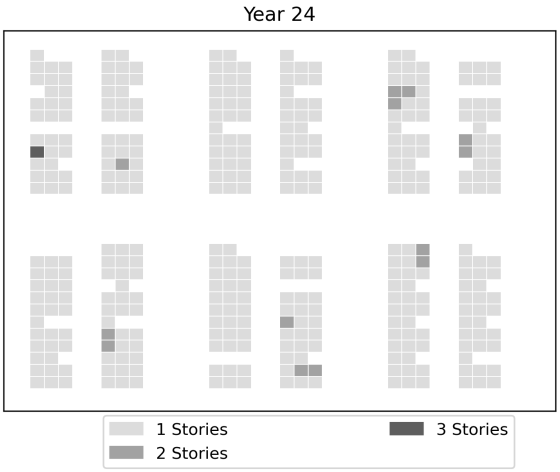


Figure 6.3.19: Spatial pattern comparison. Cooperative vs. Competitive scenarios

years to reach convergence. Interestingly, in year 15, where previously presented results showed that participating agents reached convergence in a minimal amount of episodes, the *Maximisation* process took considerably more episodes to converge. Some other such big relative differences exist, as in year 0 or year 2, which tell us that, depending on the objective function, the coordination task can be more difficult to achieve.

In terms of time per episode (Figure 6.3.21), both runs perform similarly with just a few exceptions. Particularly relevant is Year 0, where the *Maximisation* simulation took considerably more time per episode than the other scenario. This is unexpected as the energy simulation is the task that takes the most time of the optimisation

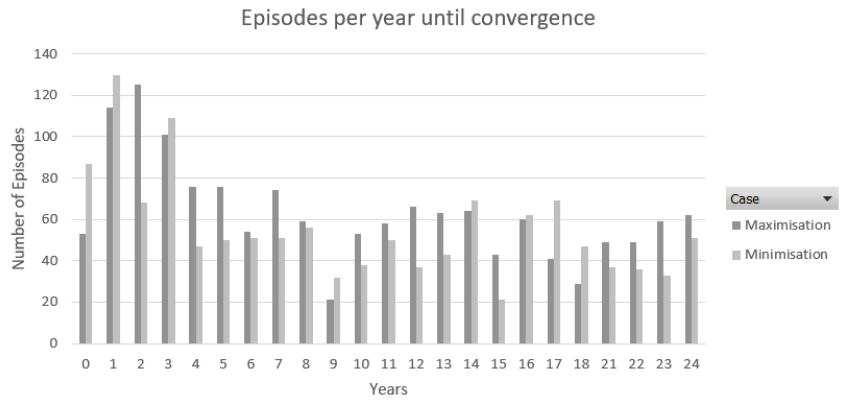


Figure 6.3.20: *Comparative number of episodes until convergence. Maximisation vs Minimisation*

process, and considering that both use the same settings, a similar execution time would be the most logical outcome. The difference is possibly derived from the fact that, as *maximisation* generates increasingly energy-dependent building envelopes, further efforts need to be made by the BEM to simulate these situations. The impact of this could be magnified when there are several active adjacent buildings, like in the case of Year 0.

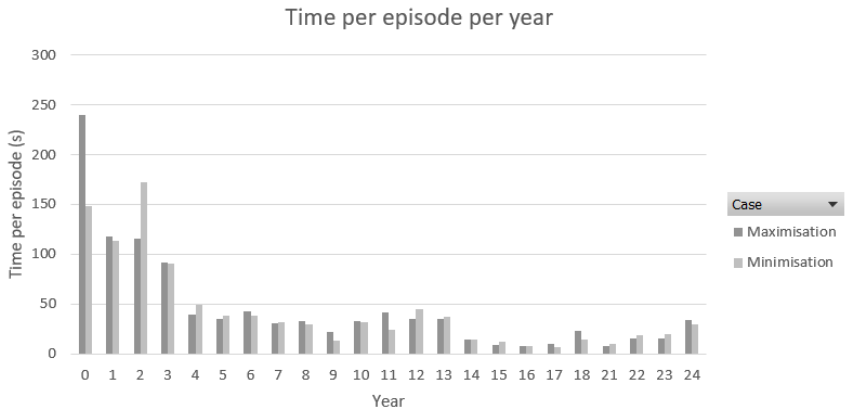


Figure 6.3.21: *Comparative average time per episode per year until convergence. Maximisation vs Minimisation*

This has consequences on the overall execution time of the *maximisation* run. Although the relative difference seems considerable (Figure 6.3.22), when compared to previous scenarios, it is actually not so much (roughly 40k seconds of difference between extreme cases in the two previous sections vs. 4k in the current one). This small difference can be explained largely by the longer processing time per episode in Year 0. Regarding aggregate energy use intensity at the neighbourhood scale Figure 6.3.23 shows clearly how values diverge. The most noticeable progression in

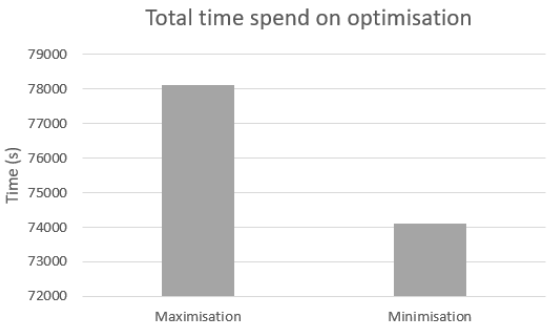


Figure 6.3.22: *Comparative total time until convergence. Maximisation vs Minimisation*

both cases happens in the earliest simulation years, precisely when there are more active agents. Even when this trend is almost symmetric, the maximisation run presents longer leaps in these early years. This might indicate that in the search space, higher energy-use solutions are easier to attain than lower ones.

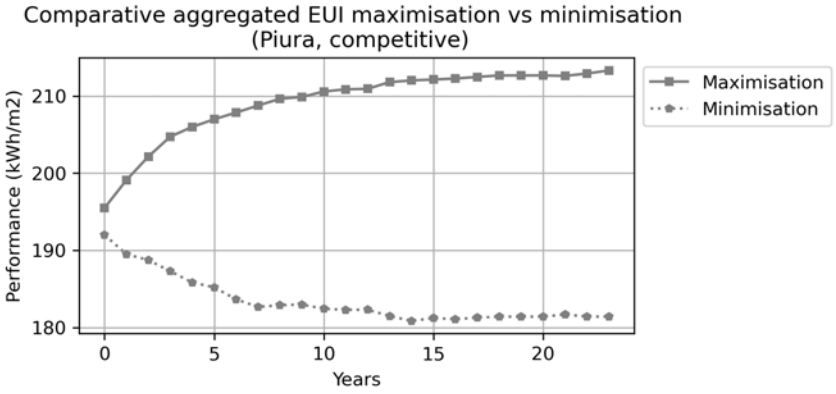


Figure 6.3.23: *Comparative aggregated EUI Maximisation vs Minimisation*

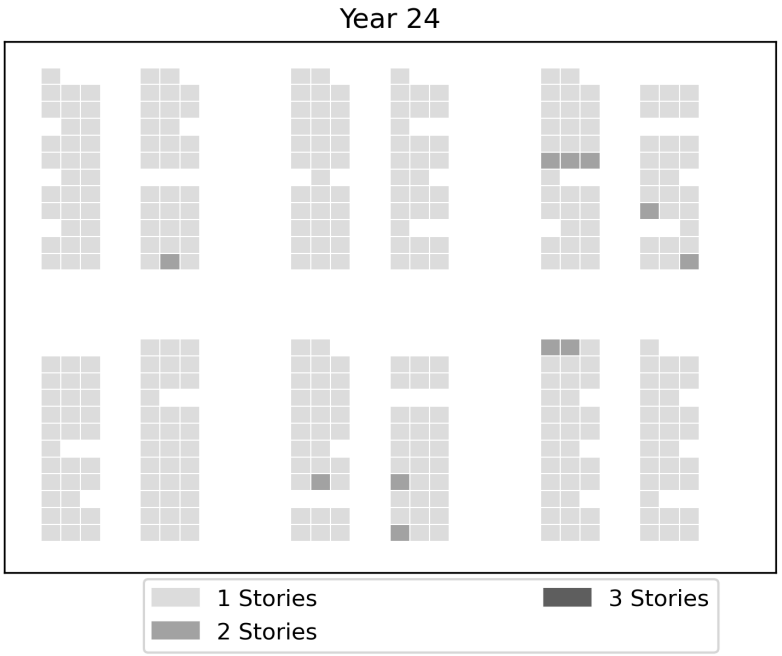
Finally, we can analyse the resulting patterns. As in previous cases, Appendix 23 shows the evolution of the neighbourhood every year and is left for the reader's consideration. The current section is dedicated to analysing the latest state of the neighbourhood. Figure 6.3.24 shows significant differences in the emerging patterns. On the *maximisation* run, there are evidently more multi-story dwellings. There is a clear preference for agents to choose to develop their houses vertically. This contrasts with the *minimisation* run, where agents seem to build vertically when no more horizontal space is available. The "towers" that agents on the *maximisation* run build seem to confirm the hypothesis posed earlier: vertical development is used as a way to increase solar heat gains. As such, it is avoided when minimising energy use in hot climates but is sometimes preferred in cold ones. The fact that on none

of the cold scenarios (Juliaca cooperative and competitive), the vertical patterns were so pronounced as in the *maximisation* run might indicate that another factor affects this choice: Due to the lack of vertical divisions, it is likely that conditioning a dwelling is more energy intensive when it has multiple stories. As such, agents in a cold climate find an intermediate solution, whilst agents with a maximisation objective function go to the extreme.

6.3.5 Scalability

The proposed methodology's scalability has been demonstrated. We have been able to jump from simulating the cognitive process of a single agent in the tests shown in Chapter 5 to tens of agents, as shown in the current chapter. This was done with minimal adaptations to the dynamic optimisation algorithm just to provide space for communication and negotiation among multiple agents. In terms of computational cost, even when these expectedly increased, that increase was manageable with simple fixes that optimised the functioning of the workflow.

(a) Minimisation spatial pattern at final simulation year (Piura, competitive)



(b) Maximisation spatial pattern at final simulation year (Piura, competitive)

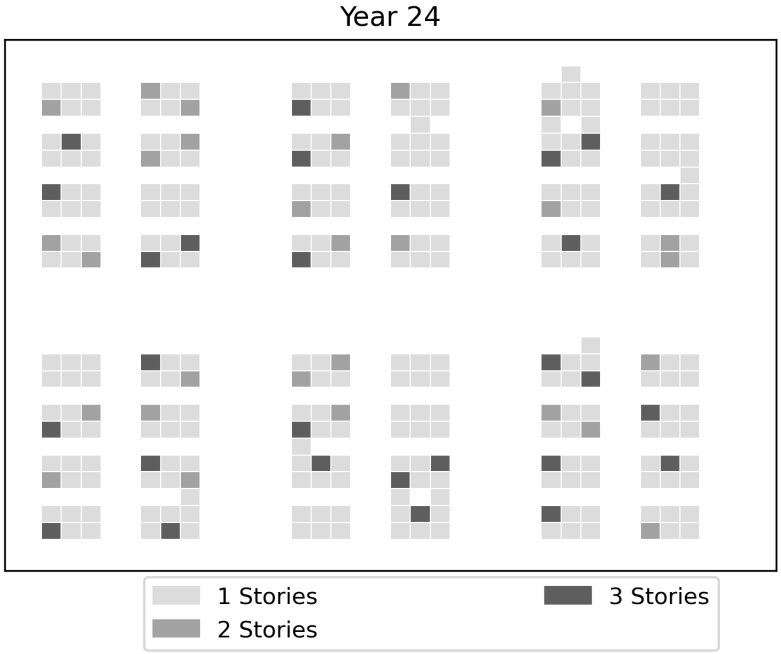


Figure 6.3.24: Spatial pattern comparison. Minimisation vs Maximisation scenarios

6.4 Conclusions

The present chapter has presented the execution of the proposed workflow. As such, its purpose was to demonstrate its functionality and to generate data that allows attaining the primary objective of this thesis. To reach these, it first presented a series of adjustments made to the workflow components to allow its integration. Consequently, it has presented the study locations from where Energy Plus weather files (EPW) are taken. This led to defining comparison scenarios used to test the system and obtain the desired outcomes. These had differing objective functions: A first one compared the system when it rewards individual performances versus when it does with locally aggregated ones, and the second one set diverging objective functions (maximise or minimise individual annual energy use). Given these scenarios and locations, the workflow was executed using the settings presented in the appendices and taking Annual Energy Use as the objective function. This allowed us to comply with the primary objective of the thesis, as Energy use considering conditioned spaces serves as a proxy for thermal comfort.

The results were presented and analysed, taking into account three variables: execution time, aggregate energy use intensity and the terminal spatial outcomes of the learning process. These serve to select suitable tools and methods for further development and to demonstrate the possibility and usefulness of applying passive building design digital tools in an incremental development context. As such, the results have been able to prove that: (1) For the proposed task, there is no comparative benefit of executing the energy simulation on the cloud, (2) our competitive setup achieves better performance at an aggregate level, and (3) there is an evident benefit of applying optimisation of building envelope geometries in an incremental setup translated in increased energy savings and thermal comfort. In addition, the simulation runs have permitted us to take a look at the emerging spatial patterns formed by intelligent agents with a given objective. Consequently, the present chapter has been able to attain its objective. It has not only demonstrated that a workflow simulating incremental residential development based on real-world data and an optimisation loop for building geometries optimisation is possible but has generated convincing data for interested parties to take this approach seriously.

Despite this, some issues remain for future development. For example, altought

we have successfully tested in previous stages the inclusion of extension costs in the decision-making process of agents (Chapter 5), this was not included in the final execution of the workflow. The micro-investment model generated data externally and determined ex-ante the moments in which agents become active. This approach has allowed for generating comparable scenarios due to their common starting point, but future developments could reintegrate this feature to include a financial variable in the learning process. Another feature tested previously but lost in this chapter is the capacity of agents to extend multiple modular rooms in a single learning process. The workflow is ready to test scenarios that include this feature, but to expedite the processing of tens of agents, it was taken out at the last minute. This latter feature could also serve to consider non-bedrooms as part of the extension, as these are vital to ensure the functionality of a residential unit. This, of course, would require classifying rooms within a geometry as potentially or non-potentially bedrooms and calculating the "need" according to this differentiation. Finally, regarding energy modelling, this could benefit from receiving geometries where each room or each floor is a "zone". This could put to the test the "build tall for heat gains" hypothesis. In this sense, Using Urban Building Energy Models (UBEM) could serve to minimise processing times in this scenario and allow the use of neighbourhood aggregate performances to test fully cooperative approaches.

References

Betti G., N. C. S. S., Tartarini F. (2022). *Cbe clima tool: a free and open-source web application for climate analysis tailored to sustainable building design. version: 0.8.10.* Retrieved from https://doi.org/10.48550/arxiv.2212.04609

Cardenas Vargas, V. (2013, 11). Planeamiento integral de la construcción de 142 viviendas unifamiliares en la ciudad de puno aplicando lineamientos de la guía del pmbok. Retrieved from http://hdl.handle.net/20.500.12404/4910

Claus, C., & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. In *Aaai/iaai.* Retrieved from https://api.semanticscholar.org/CorpusID:871178

EUROSTAT. (2021, 4). *Glossary:overcrowding rate.* Retrieved from https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Glossary:Overcrowding_rate

Fudenberg, D., & Levine, D. K. (1998). *The theory of learning in games* (Vol. 2). MIT press.

Housing and Communities Department for Levelling Up. (2019, 1). *Bedroom standard.* Retrieved from https://www.gov.uk/guidance/housing-statistics-and-england-housing-survey-glossary/a-to-z

INEI. (2015). Características socioeconómicas del hogar y de la vivienda donde residen las adolescentes. Instituto Nacional de Estadistica e Informatica. Retrieved from https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1199/cap03.pdf

INEI. (2022). *Peru: 50 años de cambios, desafios y oportunidades poblacionales* (D. Carhuavilca, A. Sánchez, C. Gutiérrez, D. Durand, & D. Mendoza, Eds.). Instituto Nacional de Estadistica e Informatica (INEI). Retrieved from https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1852/libro.pdf

Kapetanakis, S., & Kudenko, D. (2002). Improving on the reinforcement learning of coordination in cooperative multi-agent systems. In *Proceedings of the second symposium on adaptive agents and multi-agent systems (aisb02).*

Ladybug Tools LLC. (2023). *Pollination.* Retrieved from https://www.pollination.cloud/

Roudsari, M., Subramaniam, S., Dao, A., Vasanthakumar, S., Mackey, C., Haider, A., & Armour, T. (2023). *honeybee.* Retrieved from https://github.com/ladybug-tools/honeybee

Roudsari, M. S., Pak, M., & Smith, A. (2013). Ladybug: a parametric environmental plugin for grasshopper to help designers create an environmentally-conscious design.. ([computer software])

Sen, S., & Sekaran, M. (1998). Individual learning of coordination knowledge. *Journal of Experimental & Theoretical Artificial Intelligence*, *10*(3), 333–356.

The National Renewable Energy Laboratory. (2017, 9). *Energyplus™, version 00.* Retrieved from https://www.osti.gov//servlets/purl/1395882 ([computer software])

United Nations Human Settlements Programme (UN-Habitat). (2021, 12). *Sdg indicator metadata. (harmonized metadata template - format version 1.1).* Retrieved from https://unstats.un.org/sdgs/metadata/files/Metadata-11-01-01.pdf

United States Census Bureau. (2000). *Historical census of housing tables: Crowding.* Retrieved from https://www.census.gov/data/tables/time-series/dec/coh-crowding.html

WHO. (1987). *Health impact of low indoor temperatures: Report on a who meeting.* World Health Organization for Europe Copenhagen, Denmark.

Chapter 7

Conclusions and further research

7.1 Conclusions

The current research was developed to respond to the envisaged need for tools and methods to promote passive building design in aided incremental residential development. In this context, the hypothesis was that, with the implementation of passive building design, we could attain increased thermal comfort with energy efficiency at individual and aggregate levels, thus unlocking the capacity of millions of incremental developers to contribute to emissions reduction and enter a virtuous cycle of enhanced economic and physical well-being with environmental benefits. As such, the current thesis posed a single primary objective: To demonstrate the possibility of successfully implementing building envelope geometries' optimisation tools in the context of incremental residential development.

From the primary objective, three secondary ones were derived: (1) To build a computational workflow able to simulate the incremental development of dwellings, (2) to develop an optimisation methodology for building envelope geometries adapted for incremental development and (3) to develop a workflow linking these two to obtain data that demonstrates the possibility of applying passive building design in incremental residential development. These objectives led to a methodology that uses an Agent-based modelling (ABM) approach composed of two parts plus an intermediary component. While Part A generated data regarding "need" and "financial capacity" for all the requested periods, the intermediary component used these variables to feed a micro-investment decision model that sets the timing for incremental development. This timing information feeds an optimisation loop (Part B) that delivers satisficing geometrical solutions that minimise energy use with thermal comfort at hand while serving as an agent cognition tool in an ABM setting. At an aggregate level, these results should comply with the primary objective.

To test the posed workflow, the present thesis used Basic module developments in Peru as case studies. These are a form of aided self-help in which helping agencies provide an initial dwelling core that can be later developed as families' needs and financial capacity progress. This residential typology serves as testing grounds as it comprises top-down and bottom-up components, the latter plainly representing incremental development while the earlier gives trackable initial states that facilitate the simulation task. Additionally, as aided initiatives, public entities are usually

involved. In the immediate term, modelling results could support existing public policies and programmes. The location meanwhile responds to three factors: a tradition of these types of developments in the country, the availability of data to construct socio-economic models and the climate diversity of its territory. Finally, Peru is committed to reducing its residential sector emissions, as various national policies and programmes demonstrate. Nevertheless, low-income dwellers seem to be disregarded as possible contributors to these efforts. As such, applying the model to this case study could also motivate to broaden the policy approach in that country.

The bulk of this document has been devoted to justifying, describing and testing the methods selected for the workflow. As such, a socio-economic model belonging to Part A of the methodology was presented in Chapter 3. This model uses microsimulation as a method and comprises two sub-models: economic and demographic. Together, they can generate the number of household members and the joint financial capacity for each household each simulation year. The optimisation environment, meanwhile, was presented in Chapter 4 and comes in two versions used asynchronously at different stages of the workflow's development. It also includes a Geometry Generation (GG) script concerned with managing geometrical objects and sending them (along with geometry-related useful identifiers) to the other components of the optimisation process. To achieve its goal, this script complements its capabilities with those from third-party tools. Finally, due to the dynamic and multi-agent nature of our optimisation task, multi-agent reinforcement learning (MARL) was selected and tested as a Selector component of workflow's Part B. As a result of a trial-and-error experimental process, the present research implements an adapted version of Q-learning that uses Joint-action learners (JAL) and a customised naïve lambda strategy.

After selecting and testing the methods, it came time to run the entire workflow. Five simulation runs were made, varying by location, objective function or performance evaluation modality. In all of them, Annual Energy Use was used as an optimisation goal and proxy for thermal comfort. Results from these five runs were grouped and analysed according to three factors: execution time, aggregate energy use intensities and spatial patterns forming from intelligent agents with a given objective. As a result, we were able to conclude the possibility of integrating digital

tools to support passive building design in the context of aided incremental residential development. With this in mind, we can now evaluate if the research achieved its goals. Regarding Secondary Objective I, Chapters 3 and 6 of this document have presented the development and successful testing of a model that simulates residential incremental development based on plausible assumptions and real-world data. Regarding Secondary Objective II, Chapters 4 and 5 have presented the successful development and integration of an optimisation methodology for building envelope geometries adapted for incremental development. Regarding secondary objective III, Chapter 6 tests the general workflow and derives results demonstrating the possibility of applying passive building design in incremental residential development. As the workflow is run, we attain the primary objective of this thesis by showing the potential of successfully implementing building envelope geometries optimisation tools in the context of incremental residential development. This opens the gates for further research and development, particularly on the tasks pointed out at the beginning of this thesis: The implementation of this modelling approach in a participatory environment and its use to convince interested parties of its usefulness. Despite this apparent success, shortcomings remain for future adjustment. These are presented in the following section.

7.2 Further research

The present modelling workflow has left some tasks for future development. This section will put them forward by considering each workflow component individually and as interacting entities. The first of the individual approach is the socio-economic model. This could improve if it considers an immigration mechanism in its demographic sub-model, missing from the presented simulation runs due to the lack of a definition for an attractiveness factor. Another feature that could be added is the consideration of transitions between socioeconomic levels (SEL) categories or other similar categorical variables. This was not applied in the test runs as the base dataset does not register variations in these categories over the years. One variable considered in the base dataset but not in this component's modelling mechanics was the existence of multi-household dwellings. These could be included in future

developments, thus enhancing the plausibility of the modelling approach. Finally, this model would need to eventually jump from a plausible model to a statistically validated one. This implies, among other measures, implementing a more accurate macroeconomic simulation to be used as an alignment tool for the financial variables.

The optimisation environment, meanwhile, could implement geometry modules in Python that allow more freedom to execute the workflow in diverse settings. Due to the modules used in the present research, the execution was limited to Windows-based platforms. Thus, we have yet to be able to profit from available High-performance computing premises. As for the Selector component, the current research used JALs as the most straightforward solution for the coordination problem in multi-agent settings. Alternative scenarios could test Independent Learners (IL) and see if more realistic "competitive" scenarios emerge. The Energy modelling task, meanwhile, could benefit from using different thermal comfort standards. Although the learning process should not be affected (as it depends on relative differences rather than absolute values), this would allow additional data on the level of thermal comfort the diverse solutions attain. Regarding the interaction between energy modelling and the GG script, an additional workflow run could evaluate if the presence of internal divisions affects the emerging spatial patterns. Meanwhile, the interaction between the energy model and the learning process could benefit from considering whole neighbourhood aggregate performance values as an evaluation objective. Replacing the current Evaluator with an Urban Building Energy Modelling (UBEM) engine could make the optimisation possible within reasonable processing times.

Regarding the micro-investment decision model, this could benefit from being included within the learning process instead of working as an external agency. The capacity of the learning algorithm to incorporate this model was successfully tested in Chapter 5. Nevertheless, it was lost in the final runs due to magnitude. This was also the case of the capacity of dwellers to build more than one room per learning process. By reintegrating these features in a run with tens of agents, we could also profit and integrate the capacity of agents to build non-bedroom modules. This opens the door to include income-generating spaces that feed the socio-economic model. Another potential feature is the inclusion of a whole lifecycle incremental development

model that includes household agents moving in or out of the neighbourhood and a mechanism for demolition, subdivision, and re-development of buildings. Lastly, the loan mechanism could be refined to consider more informed assumptions about the micro-loan market and its conditions.

Finally, the overall functioning of the modelling setup could be improved by implementing further interaction between components. Except for the optimisation loop, the workflow presented in this thesis has a linear configuration and does not consider other feedback loops. As such, some improvements could be implemented.

- Feed the energy model with the results from the socio-economic model: The settings from the earlier could be modified dynamically by, for example, considering variations in the number of people inhabiting a dwelling and their age and work status. These variables affect occupancy ratios and usage schedules and would derive more precise results.

- Feed the economic model with the spatial states: Income-generating functions could be assigned to some rooms, thus impacting households' money availability with their presence.

- Feed the socio-economic model with Energy simulation outputs: A reduction of energy use could be translated to monetary terms and thus impact the financial capacity of dwellers.

- These loops could help optimise other geometrical and non-geometrical features, such as building materials and opening dimensions. This could broaden the objective function to include operational and embodied carbon emissions.

Besides this, as expressed in the introductory chapter, the workflow presented here intends to be an initial step in integrating passive building design tools and methods in an aided incremental design setting. As such, to carry on with this task, the results from this thesis must be discussed with the interested parties to promote a feedback process that propels further development. That should eventually lead to the transformation of the workflow into a participatory urban energy modelling tool.

Appendices

Appendix 1

Transition probabilities for the income generator status

Transition probabilities for the income generator status by age category for Male household chiefs.

Age Category	T_n	**Yes** (T_{n+1})	**No** (T_{n+1})
25-29	Yes	1.000	0.000
	No	1.000	0.000
30-34	Yes	0.980	0.020
	No	1.000	0.000
35-39	Yes	0.970	0.030
	No	0.667	0.333
40-44	Yes	0.970	0.030
	No	0.714	0.286
45-49	Yes	0.990	0.010
	No	0.667	0.333
50-54	Yes	0.956	0.044
	No	0.875	0.125
55-59	Yes	0.949	0.051
	No	0.833	0.167
60-64	Yes	0.969	0.031
	No	0.667	0.333
65-69	Yes	0.976	0.024
	No	0.750	0.250
70-74	Yes	0.974	0.026
	No	0.400	0.600
75-79	Yes	0.948	0.052
	No	0.333	0.667
79+	Yes	0.946	0.054
	No	1.000	0.000

Source: Based on data taken from INEI (2019a) Modulo-500 dataset

Transition probabilities for the income generator status by age category for Female household chiefs.

Age Category	T_n	**Yes** (T_{n+1})	**No** (T_{n+1})
25-29	Yes	0.885	0.115
	No	0.500	0.500
30-34	Yes	0.944	0.056
	No	0.800	0.200
35-39	Yes	0.961	0.039
	No	0.600	0.400
40-44	Yes	0.986	0.014
	No	0.667	0.333
45-49	Yes	0.958	0.042
	No	0.750	0.250
50-54	Yes	0.947	0.053
	No	0.833	0.167
55-59	Yes	0.940	0.060
	No	0.625	0.375
60-64	Yes	0.952	0.048
	No	0.333	0.667
65-69	Yes	0.978	0.022
	No	0.875	0.125
70-74	Yes	0.943	0.057
	No	0.800	0.200
75-79	Yes	1.000	0.000
	No	1.000	0.000
79+	Yes	0.895	0.105
	No	0.000	1.000

Source: Based on data taken from INEI (2019a) Modulo-500 dataset

Transition probabilities for the income generator status by age category for Male non-household chiefs.

Age Category	T_n	**Yes** (T_{n+1})	**No** (T_{n+1})
15-19	Yes	0.700	0.300
	No	0.480	0.520
20-24	Yes	0.837	0.163
	No	0.627	0.373
25-29	Yes	0.848	0.152
	No	0.595	0.405
30-34	Yes	0.904	0.096
	No	0.667	0.333
35-39	Yes	0.961	0.039
	No	0.667	0.333
40-44	Yes	0.958	0.042
	No	0.625	0.375
45-49	Yes	0.966	0.034
	No	0.400	0.600
50-54	Yes	0.929	0.071
	No	1.000	0.000
55-59	Yes	1.000	0.000
	No	0.400	0.600
60-64	Yes	1.000	0.000
	No	1.000	0.000
65-69	Yes	0.778	0.222
	No	0.667	0.333
70-74	Yes	0.875	0.125
	No	0.000	1.000
75-79	Yes	1.000	0.000
	No	1.000	0.000
79+	Yes	0.938	0.063
	No	0.143	0.857

Source: Based on data taken from INEI (2019a) Module-500 dataset

Transition probabilities for the income generator status by age category for Female non-household chiefs.

Age Category	T_n	**Yes** (T_{n+1})	**No** (T_{n+1})
15-19	Yes	0.627	0.373
	No	0.396	0.604
20-24	Yes	0.783	0.217
	No	0.441	0.559
25-29	Yes	0.856	0.144
	No	0.463	0.537
30-34	Yes	0.867	0.133
	No	0.534	0.466
35-39	Yes	0.894	0.106
	No	0.600	0.400
40-44	Yes	0.926	0.074
	No	0.492	0.508
45-49	Yes	0.933	0.067
	No	0.523	0.477
50-54	Yes	0.910	0.090
	No	0.520	0.480
55-59	Yes	0.868	0.132
	No	0.276	0.724
60-64	Yes	0.870	0.130
	No	0.538	0.462
65-69	Yes	0.923	0.077
	No	0.724	0.276
70-74	Yes	0.881	0.119
	No	0.083	0.917
75-79	Yes	0.760	0.240
	No	0.368	0.632
79+	Yes	0.921	0.079
	No	0.348	0.652

Source: Based on data taken from INEI (2019a) Modulo-500 dataset

Appendix 2

Age category probability distribution according to gender and chief role

Distribution probabilities for age categories by gender for household chiefs.

Age Category	Female	Male
0-4	0.000	0.000
5-9	0.000	0.000
10-14	0.000	0.000
15-19	0.004	0.003
20-24	0.024	0.018
25-29	0.047	0.051
30-34	0.061	0.086
35-39	0.096	0.099
40-44	0.114	0.118
45-49	0.123	0.135
50-54	0.097	0.115
55-59	0.116	0.100
60-64	0.090	0.085
65-69	0.075	0.071
70-74	0.057	0.050
75-79	0.049	0.036
79+	0.045	0.032

Source: Based on data taken from INEI (2016, 2017, 2018, 2019b, 2020) Modulo-200 dataset

Distribution probabilities for age categories by gender for non-household-chiefs.

Age Category	Female	Male
0-4	0.093	0.136
05-9	0.097	0.154
10-14	0.099	0.145
15-19	0.104	0.152
20-24	0.097	0.130
25-29	0.083	0.088
30-34	0.072	0.053
35-39	0.063	0.037
40-44	0.063	0.028
45-49	0.057	0.018
50-54	0.045	0.014
55-59	0.032	0.012
60-64	0.030	0.007
65-69	0.022	0.007
70-74	0.015	0.005
75-79	0.012	0.004
79+	0.017	0.009

Source: Based on data taken from INEI (2016, 2017, 2018, 2019b, 2020) Modulo-200 dataset

Appendix 3

Number of Household members per agent

		Simulation Years																								
ID	SEL	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	C	4	4	2	2	2	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	1
2	C	3	4	4	3	3	3	3	3	4	2	2	2	1	1	3	3	2	2	2	2	2	1	4	4	3	3
3	D	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	C	4	4	4	4	4	4	4	4	4	4	3	3	3	3	3	3	2	2	2	2	2	2	2	2	1	1
5	D	4	4	3	4	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3	2
6	D	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	2	2	4
7	D	4	4	3	3	3	4	4	4	4	4	4	4	4	3	2	2	2	2	2	2	2	2	2	1	1	1
8	D	3	4	3	4	3	3	3	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4	3	3
9	B	4	4	4	4	4	4	4	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	C	2	2	2	2	2	5	4	4	4	4	4	3	2	2	2	2	2	2	2	2	2	2	2	1	1	1
11	C	3	3	3	3	3	3	3	3	3	3	3	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1
12	D	6	6	6	6	6	6	5	5	4	4	4	3	3	3	3	3	3	3	2	1	1	1	1	1	1	1
13	D	5	5	5	5	5	4	4	4	4	4	4	5	4	4	4	4	4	3	3	3	2	1	1	1	1	1
14	B	4	4	3	3	3	4	3	3	3	3	3	3	3	3	2	2	2	2	2	1	1	1	1	1	1	1
15	B	4	4	4	3	3	3	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	D	4	3	2	2	2	2	1	1	1	1	1	1	1	1	4	4	4	4	4	4	4	4	5	3	3	3
17	D	2	2	2	2	2	1	1	1	1	1	2	3	2	2	2	2	1	1	1	1	1	1	1	1	1	1
18	D	2	2	2	2	3	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
19	D	4	4	4	3	3	3	3	3	2	2	2	2	1	1	7	6	6	6	6	6	5	5	5	5	5	6
20	D	4	4	3	3	2	2	2	2	2	2	3	3	2	2	2	2	2	2	1	1	1	6	7	7	6	5
21	D	6	6	6	5	5	4	4	3	3	3	2	2	2	2	2	3	2	2	2	2	2	2	2	2	2	2
22	D	2	2	2	2	2	1	1	1	1	1	1	5	5	4	4	3	3	2	5	5	5	5	5	5	5	4
23	B	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
24	C	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
25	B	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1
26	D	4	4	3	3	3	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
27	D	3	4	4	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
28	D	5	5	5	4	3	3	3	3	3	3	3	3	2	1	1	1	1	1	1	1	1	1	1	1	1	1
29	B	5	4	5	5	5	4	3	3	3	2	2	4	4	4	4	4	4	4	4	3	2	2	2	2	2	2
30	C	5	4	4	3	3	3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	3	3	3
31	C	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	2	2	2	2	2	2	2	2	2	2	2
32	D	4	4	4	4	4	3	3	4	4	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	5	4
33	C	6	5	4	3	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1
34	D	3	3	2	1	4	4	4	4	3	3	3	3	3	2	2	2	2	2	2	2	1	1	1	1	1	1
35	C	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	1	1	1	4
36	D	4	5	5	5	4	4	4	3	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
37	C	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	2	2	4	4	4	5	5	6	4
38	D	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
39	D	7	7	7	7	8	7	6	5	5	5	4	4	4	3	3	3	2	2	2	2	1	1	1	1	1	1
40	D	2	3	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1
41	D	5	4	4	4	4	5	5	4	4	4	4	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2
42	B	3	3	3	3	2	2	4	4	5	5	5	5	7	7	7	5	5	5	4	4	4	4	4	3	3	2
43	D	4	4	3	2	2	2	2	2	2	2	1	1	1	5	3	2	2	2	2	2	2	2	1	1	1	1
44	D	4	3	3	3	3	3	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
45	D	2	2	2	2	2	2	2	2	1	1	1	3	5	4	3	3	2	2	2	2	2	2	2	2	2	2
46	C	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
47	D	3	3	3	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
48	D	3	3	3	3	2	2	1	1	2	2	2	2	2	2	2	2	2	1	1	1	1	3	2	2	2	2

Number of household members for each agent and each year of simulation

Appendix 4

Yearly balance per agent

		Simulated Years																							
ID	SEL	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	C	15.81	-54.31	12.91	13.67	40.46	11.78	12.20	2.50	2.58	-2.71	13.80	19.92	0.34	12.41	12.62	0.31	-37.67	13.58	0.75	63.28	64.79	80.44	38.34	15.68	30.90	20.18
2	C	53.30	48.79	37.56	61.21	6.39	-7.49	21.96	22.58	46.92	5.44	5.49	5.66	14.62	27.01	58.48	-33.48	13.65	-12.18	13.93	0.50	35.47	39.79	96.06	53.08	4.13	36.34
3	D	-4.85	1.81	1.73	5.76	9.96	2.59	6.51	2.26	16.02	11.75	-11.33	13.70	-2.09	2.80	-2.16	8.98	9.25	-2.27	-2.31	-2.35	3.44	-2.42	-2.46	-2.49	11.44	11.71
4	C	-16.11	-0.61	-19.58	-0.07	17.07	-0.52	19.16	-20.55	41.59	64.39	41.59	-25.20	-25.69	-26.17	46.80	29.38	26.28	86.22	61.78	1.29	-13.17	0.59	37.83	7.65	-3.52	12.11
5	D	29.52	76.21	57.65	35.67	14.62	55.03	-16.60	34.53	43.59	20.96	1.63	1.79	1.96	23.15	2.28	24.72	25.50	26.29	7.02	7.23	-15.86	7.64	-14.01	6.21	-26.50	4.67
6	D	-12.97	0.17	-7.01	0.34	-14.61	-7.66	0.17	0.20	9.22	-12.76	-8.80	0.31	-8.60	1.06	7.31	5.97	9.25	15.85	-2.31	3.35	3.44	10.62	3.62	1.88	1.96	134.97
7	D	26.05	10.53	15.74	6.10	-5.04	-11.26	0.34	-11.87	-29.28	2.39	20.80	-35.57	0.67	3.16	-8.95	1.22	53.69	-10.41	-5.52	-17.04	34.43	28.96	77.58	38.41	11.44	11.71
8	D	1.69	14.27	1.63	0.68	-5.04	6.64	11.62	0.57	7.06	-6.38	12.15	2.62	12.95	-6.32	8.70	15.74	27.81	28.63	40.57	30.28	-20.36	30.68	32.75	49.46	79.89	105.64
9	B	18.92	-5.18	45.18	47.84	-8.36	-0.43	-1.53	-0.35	1.88	8.00	8.09	-3.94	8.28	1.37	-3.61	-11.99	8.65	8.74	0.86	0.76	-7.73	17.69	0.45	0.35	-6.08	9.47
10	C	-8.05	-0.87	-18.52	31.91	4.11	11.49	14.53	40.25	62.47	48.15	72.00	-25.20	0.34	66.18	0.61	57.26	0.89	26.97	61.78	49.24	36.15	-6.72	-14.13	-2.94	19.29	12.11
11	C	-2.03	-11.57	19.37	6.81	20.18	-7.51	-7.62	7.47	23.48	-29.43	-57.80	-11.13	11.82	0.43	-9.07	-3.08	-9.48	-9.69	30.89	24.38	18.08	33.15	4.57	19.34	35.41	51.33
12	D	-25.96	-6.71	-10.51	-25.08	-56.16	-30.99	-5.82	-1.67	-2.35	-12.52	-8.05	2.93	-6.26	-16.86	8.74	-10.92	-19.89	49.06	1.47	-8.52	11.38	-1.29	10.89	4.97	5.11	11.71
13	D	4.90	26.17	29.56	28.75	6.63	-11.31	-11.61	-2.46	-12.22	1.46	48.21	-12.29	59.38	-9.54	67.47	48.80	-14.58	-8.09	-8.28	2.20	-8.98	10.62	-1.27	-2.49	11.44	19.49
14	B	-22.43	-0.30	-5.94	-21.68	-10.02	19.89	1.51	1.27	-10.24	-12.04	-12.73	30.50	-61.42	-38.08	23.30	-7.66	-22.83	44.61	22.61	26.82	35.53	27.38	27.65	27.93	28.21	28.48
15	B	-17.85	-0.30	-13.70	-12.35	5.96	7.63	15.45	-12.06	15.82	1.78	31.72	24.61	6.94	1.37	21.27	21.61	1.06	-4.93	8.83	0.76	-5.42	17.69	9.20	0.35	0.25	-15.57
16	D	9.81	1.78	0.00	8.25	0.32	0.40	6.51	6.79	-1.51	7.33	27.38	33.01	3.36	-7.04	-1.64	39.91	-1.41	20.44	21.07	-1.05	-0.94	18.70	87.70	4.35	78.27	31.93
17	D	0.03	0.11	-0.01	7.78	-7.08	-1.56	10.54	11.29	21.22	22.04	10.40	-38.14	11.18	-8.78	11.25	-7.53	9.25	-2.27	-2.31	10.07	-7.09	3.53	10.89	-8.69	11.44	-2.57
18	D	-3.25	0.11	17.41	19.16	17.68	-17.11	6.55	7.35	14.12	14.67	15.21	15.76	-4.17	-14.08	-4.32	12.36	20.15	1.39	32.61	71.88	3.72	-22.09	-11.55	-10.52	17.78	3.90
19	D	-9.75	-54.37	-14.02	17.27	6.31	18.71	7.47	7.35	24.33	52.87	54.77	25.10	8.15	8.43	-12.14	-9.86	-27.90	4.16	40.66	-26.42	83.41	-3.57	64.88	-5.53	-5.55	-36.72
20	D	-12.95	-10.08	-36.74	-15.92	7.90	4.69	13.60	13.57	14.12	1.19	5.98	0.55	1.60	-8.13	1.14	11.59	-20.78	1.39	9.80	16.76	17.21	-30.51	-3.70	37.20	51.51	4.45
21	D	0.08	0.13	-21.03	28.75	25.58	8.30	-7.45	-18.15	-5.76	31.96	29.67	15.76	17.53	16.86	-4.32	1.37	25.16	19.05	19.60	44.89	-2.61	1.72	21.79	34.75	14.64	16.28
22	D	9.79	10.42	3.48	-3.41	3.98	6.24	2.49	2.63	2.35	2.44	2.53	-29.99	-1.71	-18.98	3.75	-23.26	-22.55	0.47	-5.39	-5.42	-53.58	-54.71	-9.25	34.57	147.73	192.48
23	B	9.46	-9.84	25.43	-9.97	-1.03	15.27	1.50	1.26	-1.17	10.16	-1.57	10.27	-3.29	-49.13	10.44	-2.57	-16.18	-34.13	-17.09	13.02	29.97	25.18	-21.47	10.93	-4.37	-4.57
24	C	7.91	10.19	8.50	-4.66	13.27	42.19	26.85	22.60	23.36	24.11	13.88	19.92	14.62	3.00	-2.98	9.59	29.38	-29.11	30.89	31.64	39.33	4.11	11.78	12.06	4.45	35.69
25	B	-37.82	11.43	38.92	41.06	34.05	16.46	9.12	8.91	-1.95	26.62	-2.62	-29.56	27.53	-17.67	42.53	10.49	48.83	6.73	-1.35	8.92	9.01	9.11	24.04	9.29	6.43	6.39
26	D	-6.51	10.53	26.18	18.36	18.99	1.24	47.16	30.88	23.49	33.22	7.61	2.62	-2.09	-2.12	-2.16	-7.37	9.25	9.52	17.27	16.76	10.35	4.68	37.51	12.36	34.39	13.02
27	D	-14.63	-3.08	-14.02	17.27	5.99	10.14	15.15	7.13	-19.24	-11.01	-15.88	23.02	18.36	35.17	31.19	8.98	-1.37	21.28	1.47	0.53	14.32	5.20	54.02	1.88	-9.46	2.04
28	D	-24.36	-11.58	8.52	56.50	18.99	31.63	21.06	61.20	56.44	31.96	46.78	-5.21	36.72	18.94	8.70	-2.20	14.56	-2.27	-2.31	-8.03	-2.38	10.62	10.89	18.58	17.78	5.26
29	B	-6.15	-0.30	-5.98	9.08	-75.26	-22.30	-14.72	-28.22	-10.24	-1.37	-13.43	-21.77	-58.48	42.67	14.56	-24.91	34.59	3.84	35.32	-22.30	-10.84	-11.17	18.40	18.58	-12.16	18.95
30	C	9.74	15.76	12.63	-6.88	6.66	78.12	51.01	24.91	10.34	37.52	16.76	-5.64	-17.28	6.37	43.19	69.74	84.16	16.82	-2.83	24.62	39.33	33.15	55.38	103.96	104.30	108.48
31	C	-0.05	-9.06	-9.26	-13.83	-0.25	14.14	24.39	-5.21	-5.31	-16.10	11.10	39.87	44.19	15.50	16.00	-43.27	-18.97	1.02	1.16	1.29	1.43	1.56	30.40	16.05	16.03	-13.80
32	D	-12.93	-23.62	13.88	16.50	35.57	31.14	45.18	3.47	21.01	2.23	54.77	-4.10	0.98	25.34	9.42	20.11	43.71	27.61	29.45	30.28	4.53	3.53	4.82	-2.49	-33.50	-17.38
33	C	-28.12	-24.68	-13.77	-2.30	-0.25	-9.88	-0.05	-10.24	-0.02	-10.68	16.66	39.87	11.86	30.27	31.11	15.98	16.18	10.09	10.64	17.66	18.08	25.49	11.36	-18.19	27.16	35.29
34	D	8.17	22.57	-0.01	17.31	62.52	-18.27	40.64	19.17	74.99	22.00	22.82	7.86	24.46	0.36	-4.32	5.97	-10.18	20.80	43.73	20.14	23.03	-2.42	3.62	3.71	3.80	-2.57
35	C	-8.05	2.64	12.91	22.78	8.95	2.33	-7.43	12.67	7.83	-2.67	2.83	19.93	20.52	3.18	9.34	15.98	22.71	3.33	-5.66	35.31	64.79	66.29	55.38	11.61	12.34	41.65
36	D	-19.41	26.17	47.18	70.98	72.01	2.48	51.23	22.13	10.03	44.09	72.93	23.69	8.15	4.19	13.76	8.98	9.25	9.52	9.80	16.76	4.53	4.68	4.82	4.97	25.37	19.49
37	C	-6.05	-2.18	6.46	20.45	11.05	11.78	2.36	7.60	-7.86	-45.67	8.36	-14.10	-8.49	3.30	3.91	28.49	16.99	-12.49	-13.09	-25.89	9.60	0.34	-45.10	-15.90	-85.73	74.43
38	D	9.76	17.34	-0.01	-5.32	9.78	-1.83	6.80	6.79	2.78	12.20	-5.99	-8.43	-8.60	10.91	-8.95	-9.95	0.44	-9.47	1.47	-9.82	2.69	17.03	29.72	-30.97	6.39	-16.13
39	D	4.91	-7.97	15.67	-18.53	-25.99	-7.78	-33.21	23.87	27.49	8.80	1.63	1.79	-18.52	3.16	-5.30	10.20	18.50	55.23	32.61	20.14	16.17	10.62	10.89	45.96	40.69	41.64
40	D	-6.49	-15.29	1.74	-19.67	-5.04	-11.44	-3.09	22.58	42.44	19.85	29.67	84.69	49.03	26.71	40.51	31.47	64.30	19.05	8.48	20.14	-1.30	-7.20	25.35	3.71	11.44	-9.67
41	D	4.94	-10.12	-10.54	-14.24	-30.06	-1.65	-5.05	-11.87	-4.02	-17.15	20.80	20.36	1.60	16.86	-2.81	1.22	-9.30	42.56	19.60	44.89	20.69	9.35	9.64	7.43	31.23	19.04
42	B	-6.52	-12.13	8.20	8.58	-1.72	20.26	79.43	14.67	5.58	-12.10	37.17	10.82	-0.28	-10.85	-50.22	-36.03	-37.11	0.83	-27.27	3.03	14.46	-22.34	71.90	-17.74	0.74	18.95
43	D	9.81	-13.32	-5.21	-3.41	0.32	-15.20	13.02	4.51	-12.43	2.23	27.38	37.67	24.51	8.81	59.89	3.71	-19.06	8.19	-13.75	-4.69	-34.39	7.06	-2.46	18.58	11.44	-2.57
44	D	32.56	15.63	-5.28	17.27	6.62	18.71	21.08	14.25	23.49	16.61	17.19	8.44	23.78	14.72	19.53	8.98	9.25	21.28	15.36	3.35	10.35	36.62	32.75	4.97	11.44	11.71
45	D	3.29	3.61	17.47	26.79	43.54	37.50	29.71	13.57	2.35	2.93	-11.09	2.93	25.44	-21.49	41.27	26.93	-26.49	-4.54	19.60	-10.87	16.53	-9.10	50.71	22.33	-5.06	-5.14
46	C	-6.09	5.10	6.27	2.27	-2.41	11.89	2.38	7.53	33.86	29.52	13.88	8.66	-2.87	-2.93	3.30	9.59	10.08	10.09	10.35	24.38	46.19	32.79	-2.92	18.90	4.45	20.18
47	D	-4.83	5.32	-19.27	4.70	11.98	-3.13	6.51	-14.53	7.06	2.93	12.66	-6.98	-16.77	-22.12	31.19	3.80	-2.23	-2.27	4.24	22.45	-14.53	-14.85	-9.09	-2.49	-2.53	11.71
48	D	14.72	67.39	15.72	10.13	0.32	0.40	-5.88	-1.90	-12.82	14.67	44.50	-8.43	0.33	-9.49	23.64	3.71	11.93	9.52	3.26	-2.35	-7.09	12.34	9.64	0.64	-10.69	-5.14

Yearly balance per agent in thousand of PEN

Appendix 5

Number of (potential) bedrooms per dwelling

		Simulation Years																									
ID	SEL	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	C	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	C	2	3	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
3	D	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
4	C	2	2	2	2	2	2	2	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	D	2	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	D	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3
7	D	2	3	4	4	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
8	D	2	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	5	5	5
9	B	2	2	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
10	C	2	2	2	2	2	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
11	C	2	2	2	2	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
12	D	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
13	D	2	3	4	5	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
14	B	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
15	B	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
16	D	2	2	2	3	3	3	3	3	3	3	3	3	3	3	4	5	5	5	5	5	5	5	6	6	6	6
17	D	2	2	2	2	2	2	2	2	2	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
18	D	2	2	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
19	D	2	2	2	2	2	2	2	2	2	3	3	3	3	3	4	5	6	7	7	7	7	7	7	7	7	7
20	D	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	4	4	
21	D	2	2	2	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
22	D	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4
23	B	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
24	C	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
25	B	2	2	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
26	D	2	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
27	D	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
28	D	2	2	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
29	B	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
30	C	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
31	C	2	2	2	2	2	2	2	2	2	2	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4
32	D	2	2	2	2	3	4	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	6	6
33	C	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3
34	D	2	3	3	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
35	C	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4
36	D	2	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
37	C	2	2	2	2	2	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
38	D	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
39	D	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
40	D	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
41	D	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3
42	B	2	2	2	2	2	2	3	4	5	6	6	6	7	8	8	8	8	8	8	8	8	8	8	8	8	8
43	D	2	2	2	2	2	2	2	2	2	2	2	2	2	3	4	4	4	4	4	4	4	4	4	4	4	4
44	D	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
45	D	2	2	3	3	3	3	3	3	3	3	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5
46	C	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
47	D	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
48	D	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

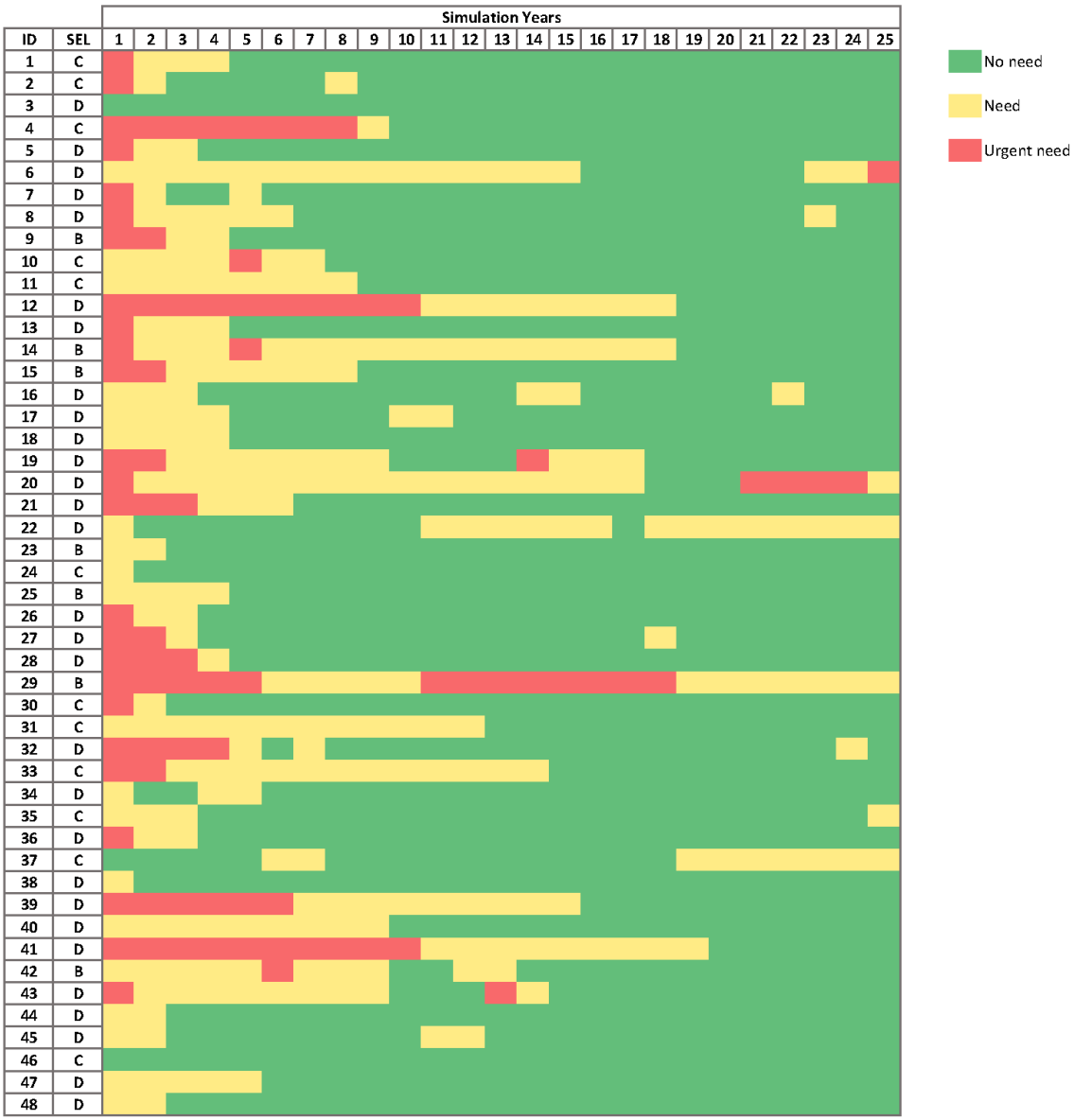
Number of (potential) bedrooms per dwelling for every simulated year

Occupancy ratio per dwelling

		Simulation Years																							
ID	SEL	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	C	0.5	1.0	1.0	1.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	3.0
2	C	0.5	0.8	1.3	1.3	1.3	1.3	1.3	1.0	2.5	2.5	2.5	5.0	5.0	1.7	1.7	2.5	2.5	2.5	2.5	2.5	5.0	1.3	1.3	1.7	1.7
3	D	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
4	C	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.8	1.3	1.3	1.3	1.3	1.3	1.3	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	4.0	4.0
5	D	0.5	1.0	1.0	1.7	1.7	1.7	1.7	1.7	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.7	1.7	1.7	2.5
6	D	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	1.0	1.0	0.5
7	D	0.5	1.0	1.3	1.3	1.0	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.7	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	5.0	5.0	5.0
8	D	0.5	1.0	0.8	1.0	1.0	1.0	2.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	1.0	1.7	1.7
9	B	0.5	0.5	0.8	1.0	1.3	1.3	2.5	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
10	C	1.0	1.0	1.0	1.0	0.4	0.8	1.0	1.3	1.3	1.3	1.7	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	5.0	5.0	5.0
11	C	0.7	0.7	0.7	0.7	1.0	1.0	1.0	1.0	1.3	1.3	2.0	2.0	2.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
12	D	0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.5	0.5	0.5	0.7	0.7	0.7	0.7	0.7	0.7	0.7	1.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
13	D	0.4	0.6	0.8	1.0	1.5	1.5	1.5	1.5	1.5	1.5	1.2	1.5	1.5	1.5	1.5	1.5	2.0	2.0	2.0	3.0	6.0	6.0	6.0	6.0	6.0
14	B	0.5	0.7	0.7	0.7	0.5	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	1.0	1.0	1.0	1.0	1.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
15	B	0.5	0.5	0.7	0.7	0.7	1.0	1.0	1.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
16	D	0.7	1.0	1.0	1.5	1.5	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	0.8	1.0	1.3	1.3	1.3	1.3	1.3	1.3	1.0	2.0	2.0	2.0
17	D	1.0	1.0	1.0	1.0	2.0	2.0	2.0	2.0	2.0	1.0	1.0	2.0	2.0	2.0	2.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
18	D	1.0	1.0	1.0	1.0	1.3	1.3	1.3	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	4.0	4.0
19	D	0.5	0.5	0.7	0.7	0.7	0.7	0.7	1.0	1.0	1.5	1.5	3.0	3.0	0.4	0.7	0.8	1.0	1.2	1.2	1.4	1.4	1.4	1.4	1.4	1.2
20	D	0.5	0.7	0.7	1.0	1.0	1.0	1.0	1.0	1.0	0.7	0.7	1.0	1.0	1.0	1.0	1.0	1.0	2.0	2.0	2.0	0.3	0.3	0.3	0.5	0.8
21	D	0.3	0.3	0.4	0.6	0.8	1.0	1.3	1.3	1.3	2.0	2.0	2.0	2.0	2.0	1.3	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
22	D	1.0	1.5	1.5	1.5	3.0	3.0	3.0	3.0	3.0	3.0	0.6	0.6	0.8	0.8	1.0	1.0	1.5	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.8
23	B	1.0	1.0	1.5	1.5	1.5	3.0	3.0	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
24	C	1.0	1.5	1.5	1.5	1.5	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0
25	B	0.7	0.7	0.7	1.0	1.3	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
26	D	0.5	1.0	1.0	1.3	2.0	2.0	2.0	2.0	2.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
27	D	0.5	0.5	0.7	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	1.0	1.5	1.5	1.5	1.5	1.5	1.5	1.5
28	D	0.4	0.4	0.5	1.0	1.3	1.3	1.3	1.3	1.3	1.3	1.3	2.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
29	B	0.5	0.4	0.4	0.4	0.5	0.7	0.7	0.7	1.0	1.0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.7	1.0	1.0	1.0	1.0	1.0	1.0
30	C	0.5	0.8	1.3	1.3	1.3	1.3	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	4.0	4.0	4.0	4.0	4.0	4.0	1.3	1.3	1.3
31	C	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.3	1.3	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
32	D	0.5	0.5	0.5	0.5	1.0	1.3	1.0	1.3	2.5	2.5	2.5	2.5	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	1.0	1.5
33	C	0.4	0.5	0.7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0
34	D	0.7	1.5	3.0	0.8	1.0	1.3	1.3	1.7	1.7	1.7	1.7	1.7	2.5	2.5	2.5	2.5	2.5	2.5	2.5	5.0	5.0	5.0	5.0	5.0	5.0
35	C	1.0	1.0	1.0	1.5	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	1.5	1.5	1.5	1.5	3.0	3.0	3.0	0.8
36	D	0.4	0.6	0.8	1.3	1.3	1.3	1.7	2.5	2.5	2.5	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
37	C	2.0	2.0	2.0	2.0	2.0	0.7	1.0	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	2.0	2.0	1.0	1.0	1.0	0.8	0.8	0.7	1.0
38	D	1.0	1.5	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	3.0	3.0
39	D	0.3	0.3	0.4	0.4	0.4	0.5	0.6	0.6	0.6	0.8	0.8	0.8	1.0	1.0	1.0	1.5	1.5	1.5	1.5	3.0	3.0	3.0	3.0	3.0	3.0
40	D	0.7	0.7	0.7	0.7	1.0	1.0	1.0	1.0	1.0	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	3.0	3.0	3.0	3.0	3.0	3.0
41	D	0.5	0.5	0.5	0.5	0.4	0.4	0.5	0.5	0.5	0.5	0.7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.5	1.5	1.5	1.5	1.5	1.5
42	B	0.7	0.7	0.7	1.0	1.0	0.5	0.8	0.8	1.0	1.2	1.2	0.9	1.0	1.1	1.6	1.6	1.6	2.0	2.0	2.0	2.0	2.0	2.7	2.7	4.0
43	D	0.5	0.7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	2.0	2.0	0.4	1.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	4.0	4.0	4.0	4.0
44	D	0.7	1.0	1.3	1.3	1.3	2.0	2.0	2.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
45	D	1.0	1.0	1.5	1.5	1.5	1.5	1.5	3.0	3.0	3.0	1.0	0.8	1.3	1.7	1.7	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5
46	C	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
47	D	0.7	0.7	1.0	1.0	1.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
48	D	0.7	1.0	1.3	2.0	2.0	4.0	4.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	4.0	4.0	4.0	4.0	1.3	2.0	2.0	2.0	2.0

Occupancy ratio in (potential) bedrooms per person for each simulation year and every dwelling

Need category evolution for every dwelling



Need category for every simulated year and dwelling

Accumulated debt per dwelling

		Simulation Years																							
ID	SEL	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	C																										
2	C																										
3	D																										
4	C									0.4																	
5	D																										
6	D																										
7	D																										
8	D		0.8																								
9	B																										
10	C																										
11	C																										
12	D																										
13	D																										
14	B																										
15	B																										
16	D																										
17	D																										
18	D																										
19	D																										
20	D																								7.4		
21	D				6.2																						
22	D																										
23	B																										
24	C																										
25	B																										
26	D		8.0																								
27	D																										
28	D																										
29	B																										
30	C																										
31	C																										
32	D																										
33	C																										
34	D																										
35	C																										
36	D		6.3																								
37	C																										
38	D																										
39	D			3.1	3.6	4.1	4.8	5.5	6.3	7.2	8.3	9.6	11.0	12.6	14.5	16.7	19.2	22.1									
40	D																										
41	D																										
42	B																										
43	D																										
44	D																										
45	D																										
46	C																										
47	D																										
48	D																										

Debt accumulated by each dwelling every simulation year in thousands of PEN

Accumulated savings per dwelling

		Simulation Years																							
ID	SEL	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	C	15.8	-38.5	-25.6	-11.9	17.2	29.0	41.2	43.7	46.3	43.6	57.4	77.3	77.6	90.0	102.7	103.0	65.3	78.9	79.6	142.9	207.7	288.2	326.5	342.2	373.1	393.2
2	C	53.3	91.7	118.5	179.8	186.1	178.7	200.6	223.2	257.5	263.0	268.4	274.1	288.7	315.7	374.2	340.7	354.4	342.2	356.1	356.6	392.1	431.9	527.9	581.0	585.1	621.5
3	D	-4.9	-3.0	-1.3	4.4	14.4	17.0	23.5	25.8	41.8	53.5	42.2	55.9	53.8	56.6	54.5	63.4	72.7	70.4	68.1	65.8	69.2	66.8	64.3	61.8	73.3	85.0
4	C	-16.1	-16.7	-36.3	-36.4	-19.3	-19.8	-0.7	-21.2	20.4	37.4	79.0	53.8	28.1	1.9	48.8	78.1	104.4	190.6	252.4	253.7	240.5	241.1	278.9	286.6	283.1	295.2
5	D	29.5	95.3	142.3	166.9	181.6	236.6	220.0	254.5	298.1	319.1	320.7	322.5	324.4	347.6	349.9	374.6	400.1	426.4	433.4	440.6	424.8	432.4	418.4	424.6	398.1	402.8
6	D	-13.0	-12.8	-19.8	-19.5	-34.1	-41.7	-41.6	-41.4	-32.2	-44.9	-53.7	-53.4	-62.0	-61.0	-53.6	-47.7	-38.4	-22.6	-24.9	-21.5	-18.1	-7.5	-3.9	-2.0	0.0	114.8
7	D	26.1	26.2	31.2	37.3	32.3	9.4	9.7	-2.1	-31.4	-29.0	-8.2	-43.8	-43.1	-40.0	-48.9	-47.7	6.0	-4.4	-9.9	-27.0	7.5	36.4	114.0	152.4	163.9	175.6
8	D	1.7	16.0	9.6	10.3	5.2	11.9	11.6	12.1	19.2	12.8	25.0	27.6	40.5	34.2	42.9	58.6	86.5	115.1	155.6	185.9	165.6	196.2	229.0	259.3	339.2	444.9
9	B	18.9	13.7	48.2	85.1	65.4	65.0	63.4	63.1	65.0	73.0	81.1	77.1	85.4	86.8	83.2	71.2	79.8	88.6	89.4	90.2	82.5	100.1	100.6	100.9	94.9	104.3
10	C	-8.0	-8.9	-27.4	4.5	8.6	8.4	11.0	39.0	101.5	149.6	221.6	196.4	196.8	262.9	263.6	320.8	321.7	348.7	410.4	459.7	495.8	489.1	475.0	472.0	491.3	503.4
11	C	-2.0	-13.6	5.8	12.6	21.5	14.0	6.3	13.8	24.7	-4.8	-62.6	-73.7	-61.9	-61.5	-70.5	-73.6	-83.1	-92.8	-61.9	-37.5	-19.5	13.7	18.3	37.6	73.0	124.3
12	D	-26.0	-32.7	-43.2	-68.3	-124.4	-155.4	-161.2	-162.9	-165.3	-177.8	-185.8	-182.9	-189.2	-206.0	-197.3	-208.2	-228.1	-179.0	-177.5	-186.1	-174.7	-176.0	-165.1	-160.1	-155.0	-143.3
13	D	4.9	20.7	39.5	57.3	52.6	41.3	29.7	27.2	15.0	16.5	64.7	52.4	111.8	102.2	169.7	218.5	203.9	195.8	187.6	189.8	180.8	191.4	190.1	187.6	199.1	218.6
14	B	-22.4	-22.7	-28.7	-50.3	-60.4	-40.5	-39.0	-37.7	-47.9	-60.0	-72.7	-42.2	-103.6	-141.7	-118.4	-126.1	-148.9	-104.3	-81.7	-54.9	-19.3	8.0	35.7	63.6	91.8	120.3
15	B	-17.9	-18.2	-31.9	-44.2	-38.3	-30.6	-15.2	-27.2	-11.4	-9.6	22.1	46.7	53.6	55.0	76.3	97.9	98.9	94.0	102.9	103.6	98.2	115.9	125.1	125.4	125.7	110.1
16	D	9.8	11.6	11.6	8.8	9.2	9.6	16.1	22.9	21.4	28.7	56.1	89.1	92.5	85.4	68.9	93.5	92.1	112.5	133.6	132.5	131.6	150.3	219.4	223.7	302.0	333.9
17	D	0.0	0.1	0.1	7.9	0.8	-0.7	9.8	21.1	42.3	64.4	61.4	9.6	20.7	12.0	23.2	15.7	24.9	22.7	20.4	30.4	23.3	26.9	37.8	29.1	40.5	37.9
18	D	-3.3	-3.1	14.3	22.4	28.8	11.7	18.2	25.6	39.7	54.4	69.6	85.4	81.2	67.1	62.8	75.1	95.3	96.7	129.3	201.2	204.9	182.8	171.2	160.7	178.5	182.4
19	D	-9.7	-64.1	-78.1	-60.9	-54.5	-35.8	-28.4	-21.0	3.3	43.2	98.0	123.1	131.2	139.7	112.6	87.4	43.8	31.8	72.4	46.0	129.4	125.9	190.7	185.2	179.7	142.9
20	D	-12.9	-23.0	-59.8	-75.7	-67.8	-63.1	-49.5	-35.9	-21.8	-20.6	-14.6	-14.1	-12.5	-20.6	-19.5	-7.9	-28.7	-27.3	-17.5	-0.7	16.5	-14.0	-17.7	19.5	14.4	18.9
21	D	0.1	0.2	-20.8	7.9	12.9	9.6	2.2	-16.0	-21.8	10.2	39.9	55.6	73.2	90.0	85.7	87.1	112.2	131.3	150.9	195.8	193.2	194.9	216.7	251.4	266.1	282.3
22	D	9.8	9.8	13.3	9.9	13.9	20.1	22.6	25.2	27.6	30.0	32.5	2.5	0.8	-18.1	-14.4	-37.7	-60.2	-59.7	-65.1	-70.5	-124.1	-178.8	-188.1	-153.5	-5.8	166.5
23	B	9.5	-0.4	14.3	4.4	3.4	18.6	20.1	21.4	20.2	30.4	28.8	39.1	35.8	-13.3	-2.9	-5.5	-21.6	-55.8	-72.8	-59.8	-29.9	-4.7	-26.1	-15.2	-19.6	-24.2
24	C	7.9	7.7	16.2	11.5	24.8	67.0	93.9	116.5	139.8	163.9	177.8	197.7	212.3	215.3	212.4	221.9	251.3	222.2	253.1	284.7	324.1	328.2	340.0	352.0	356.5	392.2
25	B	-37.8	-26.4	12.5	42.6	65.4	81.8	90.9	99.8	97.9	124.5	121.9	92.3	119.9	102.2	144.7	155.2	204.0	210.8	209.4	218.3	227.4	236.5	260.5	269.8	276.2	282.6
26	D	-6.5	4.0	8.9	16.3	35.3	36.5	83.7	114.6	138.1	171.3	178.9	181.5	179.4	177.3	175.1	167.8	177.0	186.5	203.8	220.6	230.9	235.6	273.1	285.5	319.9	332.9
27	D	-14.6	-17.7	-31.7	-14.5	-8.5	1.7	16.8	23.9	4.7	-6.3	-22.2	0.8	19.2	54.4	85.6	94.5	93.2	114.4	99.3	99.8	114.1	119.3	173.3	175.2	165.8	167.8
28	D	-24.4	-35.9	-27.4	18.1	25.8	57.4	78.5	139.7	196.1	228.1	274.8	269.6	306.3	325.3	334.0	331.8	346.4	344.1	341.8	333.7	331.4	342.0	352.9	371.5	389.2	394.5
29	B	-6.2	-6.5	-12.4	-3.4	-78.6	-100.9	-115.6	-143.9	-154.1	-155.5	-168.9	-190.7	-249.2	-206.5	-191.9	-216.8	-182.2	-178.4	-143.1	-165.4	-176.2	-187.4	-169.0	-150.4	-162.6	-143.6
30	C	9.7	15.1	17.0	10.2	16.8	94.9	145.9	170.8	181.2	218.7	235.5	229.8	212.5	218.9	262.1	331.8	416.0	432.8	430.0	454.6	493.9	527.1	582.5	686.4	790.7	899.2
31	C	-0.1	-9.1	-18.4	-32.2	-32.4	-18.3	6.1	0.9	-4.4	-20.5	-9.4	16.7	46.8	62.3	78.3	35.0	16.1	17.1	18.2	19.5	21.0	22.5	52.9	69.0	85.0	71.2
32	D	-12.9	-36.5	-22.7	-6.2	18.1	37.6	82.8	74.0	95.0	97.2	152.0	147.9	148.9	174.2	183.6	203.8	247.5	275.1	304.5	334.8	339.3	342.9	347.7	345.2	292.1	274.7
33	C	-28.1	-52.8	-66.6	-68.9	-69.1	-79.0	-79.1	-89.3	-89.3	-100.0	-83.3	-43.5	-31.6	-1.3	14.9	30.9	47.0	57.1	67.8	85.4	103.5	129.0	140.4	122.2	149.3	184.6
34	D	8.2	20.3	20.3	37.6	88.8	59.0	99.6	118.8	193.8	215.8	238.6	246.4	270.9	271.3	266.9	272.9	262.7	283.5	327.3	347.4	370.4	368.0	371.6	375.4	379.2	376.6
35	C	-8.1	-5.4	7.5	19.3	28.2	30.6	23.1	35.8	43.6	40.9	43.8	63.7	84.2	87.4	96.7	112.7	135.4	138.8	133.1	168.4	233.2	299.5	354.9	366.5	378.8	400.3
36	D	-19.4	6.8	14.4	74.4	146.4	148.8	200.1	222.2	232.2	276.3	349.3	373.0	381.1	385.3	399.1	408.0	417.3	426.8	436.6	453.4	457.9	462.6	467.4	472.4	497.7	517.2
37	C	-6.0	-8.2	-1.8	18.7	29.7	41.5	31.9	27.2	19.4	-26.3	-17.9	-32.0	-40.5	-37.2	-33.3	-4.8	12.2	-0.3	-13.4	-39.3	-29.7	-29.3	-74.5	-90.4	-176.1	-101.7
38	D	9.8	16.7	16.7	11.4	21.1	19.3	26.1	32.9	35.7	47.9	41.9	33.5	24.9	35.8	26.8	16.9	17.3	7.8	9.3	-0.5	2.2	19.2	48.9	18.0	24.3	8.2
39	D	4.9	-3.1	12.6	-5.9	-31.9	-39.7	-72.9	-49.0	-21.5	-12.7	-11.1	-9.3	-27.8	-24.7	-30.0	-19.8	-1.3	6.9	39.6	59.7	75.9	86.5	97.4	143.3	184.0	225.7
40	D	-6.5	-21.8	-20.0	-39.7	-44.8	-56.2	-59.3	-36.7	5.7	12.6	42.3	127.0	176.0	202.7	243.2	274.7	339.0	358.0	366.5	386.7	385.3	378.1	403.5	407.2	418.7	409.0
41	D	4.9	-5.2	-15.7	-30.0	-60.0	-61.7	-66.7	-78.6	-82.6	-99.8	-79.0	-58.6	-57.0	-40.2	-43.0	-41.7	-51.0	-8.5	11.1	38.9	59.6	68.9	78.6	86.0	117.2	136.3
42	B	-6.5	-18.6	-10.5	-1.9	-3.6	16.7	84.1	86.5	79.5	54.4	91.6	102.4	88.0	62.7	12.5	-23.6	-60.7	-59.9	-87.1	-84.1	-69.6	-92.0	-20.1	-37.8	-37.1	-18.1
43	D	9.8	-3.5	-8.7	-12.1	-11.8	-27.0	-14.0	-9.5	-21.9	-19.7	7.7	45.4	69.9	64.2	109.2	112.9	93.9	102.1	88.3	83.6	49.2	56.3	53.8	72.4	83.8	81.3
44	D	32.6	37.8	21.8	39.1	45.7	64.4	85.5	99.7	123.2	139.8	157.0	165.5	189.3	204.0	223.5	232.5	241.7	263.0	278.4	281.7	292.1	328.7	361.4	366.4	377.8	389.6
45	D	3.3	6.9	13.7	40.5	84.0	121.5	151.2	164.8	167.2	170.1	159.0	148.2	159.6	138.1	179.3	206.3	179.8	175.2	194.8	183.9	200.5	191.4	242.1	264.4	259.4	254.2
46	C	-6.1	-1.0	5.3	7.5	5.1	17.0	19.4	26.9	60.8	90.3	104.2	112.8	110.0	107.0	110.3	119.9	130.0	140.1	150.5	174.8	221.0	253.8	250.9	269.8	274.3	294.4
47	D	-4.8	0.5	-18.8	-14.1	-2.1	-5.2	1.3	-13.2	-6.2	-3.3	9.4	2.4	-14.4	-36.5	-5.3	-1.5	-3.7	-6.0	-1.8	20.7	6.2	-8.7	-17.8	-20.3	-22.8	-11.1
48	D	14.7	71.7	76.7	86.9	87.2	87.6	81.7	79.8	67.0	81.7	126.2	117.7	118.1	108.6	132.2	135.9	147.8	157.4	160.6	158.3	151.2	163.5	173.2	173.8	163.1	158.0

Accumulated savings for each agent every year of the simulation in thousands of PEN

Appendix 6

Algorithm 1 Objective function: Maximise

```
for all a ∈ A do
    procedure Determine if the geometry is Manifold and that needs are met
        if a.geometry is Manifold then
            m = true
        else
            m = false
        end if
        if Space need is satisfied with the new state then
            n = true
        else
            n = false
        end if
    end procedure
    procedure Determine if the new performance is worthy of reward and modify record accordingly(s, R, P, t)
        if s ∈ R[a] then                          ▷ This is the first time we visit a size
            R[a][s] ← P
            r ← P
        else
            if P > R[a][s] then                   ▷ Found performance better than record
                R[a][s] ← P
                r ← P
            else
                r ← R[a][s]
            end if
        end if
        u ← r − (r ∗ t)
        if P ≥ u ∧ m = true then                  ▷ Filter non-manifold solutions
            b ← true
        else
            b ← false
        end if
    end procedure
    procedure Assign reward to the current state(b, n, m)
        if b = true ∧ n = true then
            reward = 25
        else if b = false ∧ n = true then
            reward = −25
        else if m = false then
            reward = −25
        else
            if current step is last episode's step then
                reward = −25
            else
                reward = −1                       ▷ Minimise number of steps to reach solution
            end if
        end if
    end procedure
end for
```

- A is the set of participating agents
- s is the size of the current state (in occupied points)
- R is the record dictionary where past high-performing values are stored by agent identification and state size.
- P is the performance of the current state.
- t is the tolerance value to consider a performance worthy of a reward.

Appendix 7

Algorithm 2 Objective function: Minimise

```
for all a ∈ A do
    procedure Determine if the geometry is Manifold and that needs are met
        if a.geometry is Manifold then
            m = true
        else
            m = false
        end if
        if Space need is satisfied with the new state then
            n = true
        else
            n = false
        end if
    end procedure
    procedure Determine if the new performance is worthy of reward and modify record accordingly(s, R, P, t)
        if s ∈ R[a] then                          ▷ This is the first time we visit a size
            R[a][s] ← P
            r ← P
        else
            if P < R[a][s] then                   ▷ Found performance better than record
                R[a][s] ← P
                r ← P
            else
                r ← R[a][s]
            end if
        end if
        u ← r + (r ∗ t)
        if P ≤ u ∧ m = true then                  ▷ Filter non-manifold solutions
            b ← true
        else
            b ← false
        end if
    end procedure
    procedure Assign reward to the current state(b, n, m)
        if b = true ∧ n = true then
            reward = 25
        else if b = false ∧ n = true then
            reward = −25
        else if m = false then
            reward = −25
        else
            if current step is last episode's step then
                reward = −25
            else
                reward = −1                       ▷ Minimise number of steps to reach solution
            end if
        end if
    end procedure
end for
```

- A is the set of participating agents
- s is the size of the current state (in occupied points)
- R is the record dictionary where past high-performing values are stored by agent identification and state size.
- P is the performance of the current state.
- t is the tolerance value to consider a performance worthy of a reward.

Appendix 8

Algorithm 3 Autostop algorithm

procedure COLLECT THE PROPORTION OF "WIN" EPISODES PER AGENT(t, E, W, A, R, q, s)
  $E_i \subset E$
  $C \leftarrow NewDict()$
  if $E_i > E \cdot t$ **then**
    for all $e \in E_i$ **do**
      $c \leftarrow 0$
      $N \leftarrow NewList()$
      for all $a \in A$ **do**
        if $e \in W[a]$ **then**
          if $W[a][e] \neq$ "$lose$" $\wedge\, W[a][e] \neq -R$ **then**
            if $W[a][e] = nonmani$ **then**
              AddItem(N, true)
            else
              AddItem(N, false)
              if $W[a][e] \leq q + (q \cdot s)$ **then**
                $c = c + 1$
              end if
            end if
          end if
        end if
      end for
      if $\exists$ true $\in N$ **then**
        $C[e] \leftarrow 0$
      else
        $C[e] \leftarrow c \div size(A)$
      end if
    end for
  end if
end procedure
procedure DETERMINE IF THE LEARNING PROCESS STOPS IN THE CURRENT EPISODE(C, P)
  $stop \leftarrow false$
  if $size(C) > 0$ **then**
    $m \leftarrow x$ corresponding to $max(y) \forall (x, y) \subset C$
    $p \leftarrow (\sum y \forall (x, y) \subset C) \div size(C)$
    $u \leftarrow \exists!\ y\ \forall\ (x, y) \subset C$
    if $size(u) = 1 \wedge u[0] > 0$ **then**
      $stop \leftarrow true$
    end if
    if $p > P$ **then**
      $stop \leftarrow true$
    end if
  end if
end procedure

- Where E is the maximum number of episodes
- t is the fraction of E to be evaluated to determine the auto-stop.
- E_i are all elapsed episodes
- W is a three-dimensional array that stores agent IDs, episode IDs and the outcomes of those episodes for those agents.

- A is the set of participating agents.
- R is the maximum reward attainable (or penalisation when negative).
- q Maximum (or minimum, depending on the objective function) performance on record at the moment.
- s Tolerance value, a fraction of the current record performance that determines acceptable rewarded solutions.
- P The stop proportion input.

Appendix 9

Basic module Budget

The following budget example was taken from Cardenas Vargas (2013), which calculates the construction cost of a basic module of 40m2 of built area in the city of Puno in the southern Peruvian Andes (2009 prices). For the purpose of this thesis, the budget items have been re-arranged in sections that better help discriminate costs that are exclusive to an extension from those that are included in the dwelling core.

	Units	**Quant.**	**Unit Cost**	**Partial (PEN)**	**Total (PEN)**
General expenses (all construction works)					**179.31**
Guardian House and warehouse	GLB	1	93.89	93.89	
Worksite cleaning	GLB	1	85.42	85.42	
Expenses exclusive to ground floors					**5700.90**
Manual ground cleaning	m^2	48.52	1.79	86.8508	
Layout	m^2	38.75	0.88	34.1	
Level tracing and Layout Modifications	m^2	25.6	0.84	21.504	
Trenches for FoundationS	m^3	19.5	21.37	416.715	
Interior leveling and Compaction	m^2	54.32	2.72	147.7504	
Manual Filling with External Material	m^3	3.7	27.94	103.378	
Elimination of surplus material	m^3	18.98	4.99	94.7102	
Compacted Filling in Batches with own material	m^3	12.9	21.34	275.286	
FoundationS	m^3	12.65	148.75	1881.6875	

Concrete for FoundationS Protection	m^3	1.34	202.31	271.0954	
Formwork Installation and Removal for Foundation Protectors	m^2	17.85	23.14	413.049	
False Floor Concrete	m^2	33.73	23.79	802.4367	
Concrete in Reinforced Foundation Protectors	m^3	1.67	213.96	357.3132	
Formwork Installation and Removal for Foundation Protectors	m^2	20.25	23.14	468.585	
Structural Steel for Reinforced Foundation Protection	kg	79.62	4.1	326.442	
Walls (core or extension)					**10789.16**
Concrete in Columns	m^3	4.58	332.13	1521.1554	
Formwork Installation and Removal for Columns	m^2	22.94	41.91	961.4154	
Structural Steel for Columns	kg	328.62	4.17	1370.3454	
Stretcher bond KK Brick WallS	m^2	66.66	38.58	2571.7428	
Indoor Plastering	m^2	86.05	14.2	1221.91	
Outdoor Plastering	m^2	36.27	14.17	513.9459	
Plastering in Windows doors and other openings	m	41.5	8.7	361.05	
Framed wood doors	pce	1	280	280	
Framed plywood doors	pce	9	4.46	40.14	
Double glass (2) modular system (4 mm)	pce	3	250	750	

Double glass modular system (4 mm)	pce	1	65	65	
Vinyl Paint ON Exterior WallS	m^2	36.27	7.26	263.3202	
Vinyl Paint ON interior WallS	m^2	86.05	6.08	523.184	
Wall Power Outlets	pt	5	69.19	345.95	
Roofs/Ceiling (core or extension)					**6269.73**
Concrete in Beams	m^3	1.86	312.96	582.1056	
Formwork Installation and Removal in Beams	m^2	5.09	42.46	216.1214	
Structural Steel for Beams	kg	226.83	4.17	945.8811	
Concrete in Lightened Slabs	m^3	1.98	301.01	595.9998	
Formwork Installation and Removal in Lightened Slabs	m^2	30.93	30.51	943.6743	
Structural Beam for Lightened Slabs	kg	205.48	4.17	856.8516	
Hollow Clay Brick for Lightened Slabs	unit	248	1.91	473.68	
Ceilings	m^2	35.19	27.34	962.0946	
Vinyl Paint on Ceiling	m^2	35.19	5.19	182.6361	
Roofing	m^2	40.21	6.36	255.7356	
Ceiling Outlet	pt	5	50.99	254.95	
Floors (only finishings, core or extensions)					**1146.75**
Rubbed-colored Concrete FloorS	m^2	24.17	28.84	697.0628	
Light colored Venetian tile Floor	m^2	9.58	46.94	449.6852	

Basic appliances included in the core					**2319.93**
Kitchen Ceramic Wall	m^2	1.42	43.79	62.1818	
Framed Wood Doors	pce	2	380	760	
Main Door Lock	pce	2	35.23	70.46	
Stainless Steel Kitchen Sink	pce	1	161.2	161.2	
5. 3" PVC Rain Water Outlets	pt	2	31.85	63.7	
2" Bronze Cap Provision and Placement	unit	1	17.01	17.01	
Registry box 12" X 24" with Concrete Cap	pce	1	86.53	86.53	
Kitchen Sink Plastering	m^2	2.1	19.68	41.328	
Irrigation Wrench with 1/2" tap	pce	2	45.97	91.94	
Hydraulic Test for Residential Water Installation Piping	m	26.35	3	79.05	
Kitchen sink table	unit	1	77.24	77.24	
Mixed Power Outlet	pce	1	50.03	50.03	
Power Outlet for Kitchen	pt	1	65.58	65.58	
Phone Outlet	pt	1	62.91	62.91	
TV Antenna Outlet	pt	1	52.7	52.7	
Output for GONG Type Bell	pt	1	77.67	77.67	
Distribution Board	unit	1	109.58	109.58	
Electrical Network Conection	unit	1	159.59	159.59	
Concrete Wall for Electricity Metering	unit	1	231.23	231.23	

Appliances shared between Kitchens and bathrooms (included in core)					**1480.26**
2" PVC Sewage Outlets	pt	7	45.06	315.42	
2" PVC Outlets	m	17	16.18	275.06	
2" Bronze Sink Provision and Placement	unit	3	18.68	56.04	
Cold Water Outlet with 1/2" PVC-SAP Pipe	pt	6	49.87	299.22	
Distribution network 1/2" PVC-SAP pipe	m	25.5	10.43	265.965	
1/2" Bronze gate valves	pce	4	67.14	268.56	
Additional costs of a Bathroom (extension)					**1505.24**
Domestic Ceramic Wall	m^2	8.06	37.34	300.9604	
Concrete skirting	m	7	8.14	56.98	
Bathroom door lock	pce	1	26.37	26.37	
Low Tank Toilet	pce	1	265.83	265.83	
Wall Lavatory	pce	1	137.53	137.53	
Chomed Shower incl. accesories	unit	1	64.98	64.98	
4" PVC sewage Outlets	pt	3	59.03	177.09	
2" PVC Ventilation Outlets	pt	1	37.35	37.35	
4" PVC Outlets	m	20	20.67	413.4	
4" Bronze Cap Provision and Placement	pce	1	17.16	17.16	
2" PVC Ventilation Hat	pce	1	7.59	7.59	
				TOTAL	**29391.29**

Aggregated Unit costs

From the previous categories, it is possible to get aggregated unit costs:

Category	**Unit cost detail**	**Category budget**	**Quantity**	**Unit cost**
General expenses	Per work	179.31	1	179.31
Exclusive to ground floors	Per m^2 of built area	5700.90	40.00	142.52
Walls	Per m^2 of wall	10789.16	66.66	161.85
Roofs/Ceiling	Per m^2 of ceiling	6269.73	35.19	178.17
Floors	Per m^2 of finished floor	1146.75	33.75*	33.98
Shared between Kitchens and bathrooms	0.5 per bathroom	1480.26	0.50	740.13
Additional costs of a Bathroom	Per bathroom	1505.24	1	1505.24

* This is the sum in area of the two items belonging to this category

Extension Budget With these unitary costs, it is possible to generate approximate budgets for different extension situations using the prices for 2009. These tables aim to answer "How much did it cost to build an extension of 2.7*2.7*2.7 in Puno, Peru, in 2009?" (the size corresponds to one spatial module in the simulation).

Case A: Located on the ground floor, needs only ONE wall to enclose the perimeter.

Category	**Unitary cost detail**	**Unit**	**Unitary cost (PEN)**	**Total cost (PEN)**
Ground extras	per m^2 of built area	7.29	142.52	1038.99
General expenses	unit	1	179.31	179.31
Walls	per m^2 of built wall	7.29	161.85	1179.91
Ceiling	per m^2 of ceiling	7.29	178.17	1298.84
Floor	per m^2 of floor	7.29	33.98	247.70
			Partial total (PEN)	**3944.75**

Case B: Located on the ground floor, needs TWO walls to enclose the perimeter.

Category	**Unitary cost detail**	**Unit**	**Unitary cost (PEN)**	**Total cost (PEN)**
Ground extras	per m^2 of built area	7.29	142.52	1038.99
General expenses	unit	1	179.31	179.31
Walls	per m^2 of walls	14.58	161.85	2359.82
Ceiling	per m^2 of ceiling	7.29	178.17	1298.84
Floor	per m^2 of finished floor	7.29	33.98	247.67
			Partial total (PEN)	**5124.67**

Case C: Located on the ground floor, it needs THREE walls to enclose the perimeter.

Category	Unitary cost detail	Unit	Unitary cost (PEN)	Total cost (PEN)
Ground extras	per m^2 of built area	7.29	142.5226	1038.99
General expenses	unit	1	179.31	179.31
Walls	per m^2 of walls	21.87	161.85	3539.74
Ceiling	per m^2 of ceiling	7.29	178.1679	1298.84
Floor	per m^2 of finished floor	7.29	33.97772	247.70
			Partial total (PEN)	**6304.58**

Case D: Located above the ground floor, it needs ONE wall to enclose the perimeter.

Category	Unitary cost detail	Unit	Unitary cost (PEN)	Total cost (PEN)
General expenses	unit	1	179.31	179.31
Walls	per m^2 of walls	7.29	161.85	1179.913
Ceiling	per m^2 of ceiling	7.29	178.17	1298.844
Floor	per m^2 of finished floor	7.29	33.98	247.6976
			Partial total	**2905.76**

Case E: Located above the ground floor, needs TWO walls to enclose the perimeter

Category	Unitary cost detail	Unit	Unitary cost (PEN)	Total cost (PEN)
General expenses	unit	1	179.31	179.31
Walls	per m^2 of walls	14.58	161.85	2359.83
Ceiling	per m^2 of ceiling	7.29	178.17	1298.84
Floor	per m^2 of finished floor	7.29	33.98	247.70
			Partial total	**4085.68**

Case F: Located above the ground floor, needs THREE walls to enclose the perimeter

Category	Unitary cost detail	Unit	Unitary cost (PEN)	Total cost (PEN)
General expenses	unit	1	179.31	179.31
Walls	per m^2 of walls	21.87	161.85	3539.74
Ceiling	per m^2 of ceiling	7.29	178.17	1298.84
Floor	per m^2 of finished floor	7.29	33.98	247.70
			Partial total	**5265.59**

Case G: Located above the ground floor, needs FOUR walls to enclose the perimeter

Category	**Unitary cost detail**	**Unit**	**Unitary cost (PEN)**	**Total cost (PEN)**
General expenses	unit	1	179.31	179.31
Walls	per m^2 of walls	29.16	161.85	4719.65
Ceiling	per m^2 of ceiling	7.29	178.17	1298.84
Floor	per m^2 of finished floor	7.29	33.98	247.70
			Partial total	**6445.50**

Additional space needed

Extensions usually need areas besides their assigned function to provide access or complementary services. In the case of potential bedrooms, this is more evident, as they are usually fully enclosed and increase the demand for toilets.

The additional area for access to the extension is calculated with the most space-consuming solution, which is a corridor that runs adjacent to one of the sides of the module. This generates a corridor with an area of 2.43 m^2 (2.7*0.9 m) using a standard corridor width of 0.9 m. As this access could be potentially shared among at least two spaces, only half of that figure (2.215 m^2) is considered as the access area needed for each modular extension. This is equivalent to 0.15 of the module area, a fraction used in the budget to maintain the differences among cases. Toilets are sized using the same fraction of the extensions' built area, with additional costs referred to the need to implement additional appliances in this sort of space.

Case	**Room cost**	**Access**	**Bathroom**	**Bathroom exclusive appliances**	**Bathroom shared appliances**	**Total**
A	3944.75	591.71	591.71	1505.24	740.13	7373.55
B	5124.67	768.70	768.70	1505.24	740.13	8907.44
C	6304.58	945.69	945.69	1505.24	740.13	10441.33
D	2905.76	435.86	435.86	1505.24	740.13	6022.86
E	4085.68	612.85	612.85	1505.24	740.13	7556.76
F	5265.59	789.84	789.84	1505.24	740.13	9090.64
G	6445.5	966.83	966.83	1505.24	740.13	10624.52

Additional budget in PEN assigned to auxiliary spaces needed for functional extensions

Appendix 10

To get an average rate of inflation for the model, the most straightforward manner was to use the official building unit values published by the government to value buildings for tax purposes. Although these values are not commercial and do not aim to reflect the cost of building; they are updated yearly, taking into account inflation.

The following table shows the values of a building of an area of 40m2 (the same size as the budgeted core shown in Appendix 6) for every year between 2015 and 2019 in the "Sierra" region. The letters represent each of the categories included in the official building unit values for those years (Ministerio de Vivienda, 2015, Ministerio de Vivienda, 2016, Ministerio de Vivienda, 2017, Ministerio de Vivienda, 2018, Ministerio de Vivienda, 2019)

Year	**2015**	**2016**	**2017**	**2018**	**2019**
Built Area (m^2)	40	40	40	40	40
Walls	C	C	C	C	C
Roofs	C	C	C	C	C
Floors	G	G	G	G	G
Doors and windows	F	F	F	F	F
Coatings	F	F	F	F	F
Toilets	C	C	C	C	C
Electrical and sanitary fittings	E	E	E	E	E
Total value	23140.0	23834.8	24549.6	25285.6	25817.2
Yearly increase (%)		+3.00	+2.99	+3.00	+2.10

Valuation in PEN using the official unitary values for a 40 m^2 construction in the Sierra region of Peru

The average yearly increase in value for the selected years is 2.77 %

Appendix 11

Honeybee/EnergyPlus Settings

Construction sets

Exterior subset

Parameter	**Variable**	**Value**	**Source**
Wall	extw	7/8" Stucco + 0. 4" Normal weight concrete wall + 7/8" Stucco (R = 0.108 m^2 K/W).	Direct observation. Honeybee material library.
Roof	extr	Generic roof membrane + 0. 8" Normal-weight concrete floor + 7/8" Stucco (R = 0.182 m^2 K/W).	Direct observation. Honeybee material library.
Floor	expfl	7/8" Stucco + 0. 8" Normal-weight concrete floor + 7/8" Stucco (R = 0.152 m^2 K/W).	Direct observation. Honeybee material library.

Ground subset

Parameter	Variable	Value	Source
Wall	extw	Same as exterior wall (R = 0.108 m^2 K/W)	Direct observation.Honeybee material library.
Roof	extr	Same as exterior roof (R = 0.182 m^2 K/W)	Direct observation.Honeybee material library.
Floor	grdfl	Concrete pavement (R = 0.115 m^2 K/W)	Direct observation.Honeybee material library.

Sub-face subset

Parameter	Variable	Value	Source
Window	windows_	U 0.11 SHGC 0.34 Simple glazing (R = 1.42 m^2 K/W).	Direct observation. Honeybee material library.
Skylight	-	None	Direct observation.
Operable	-	None	Because ventilation is 0.
Exterior door	doors_	5/8" plywood + Generic wall air gap + 5/8" plywood (R = 0.417 m^2 K/W).	Direct observation. Honeybee material library.
Overhead door	-	None	Direct observation.
Glassdoor	-	None	Direct observation.

Schedules

People occupancy schedules

Hours	Weekdays (peop_wday)	Weekends (peop_wend)
0:00 – 6:00 hrs	1	1
06:00 – 08:00 hrs	0.5	0.75
08:00 – 17:00 hrs	0.25	0.75
17:00 – 21:00 hrs	0.5	0.75
21:00 – 24:00 hrs	1	1

Lighting schedule

Hours	Use
0:00 – 6:00 hrs	0
06:00 – 08:00 hrs	0.5
08:00 – 17:00 hrs	0
17:00 – 19:00 hrs	0.5
19:00 – 22:00 hrs	1
22:00 – 0:00 hrs	0

Electric equipment use schedule

Hours	Weekdays (eq_use)	Weekends (eq_sch)
0:00 – 6:00 hrs	0	0
06:00 – 08:00 hrs	0.5	0.5
08:00 – 17:00 hrs	0	0.5
17:00 – 19:00 hrs	0.5	0.5
19:00 – 22:00 hrs	0.5	0.5
22:00 – 0:00 hrs	0	0

Gas equipment use schedule

Hours	Use
0:00 – 6:00 hrs	0
06:00 – 07:00 hrs	1
07:00 – 11:00 hrs	0
11:00 – 13:00 hrs	1
13:00 – 17:00 hrs	0
17:00 – 18:00 hrs	1
18:00 – 24:00 hrs	0

Hot water use schedule

Hours	Use
0:00 – 6:00 hrs	0
06:00 – 07:00 hrs	0.5
07:00 – 18:00 hrs	0
18:00 – 19:00 hrs	0.5
19:00 – 24:00 hrs	0

Cooling and heating setpoints

	Setpoint schedule	
Parameter	**Variable**	**Value**
Heating	heat_spoint	18 ^oC
Cooling	cool_spoint	24 ^oC
Humidification	hum_spoint	None
Dehumidification	dhum_spoint	None

These setpoints do not reflect the presence of HVAC systems but are a reference to achieve a standard "healthy" indoor environment (WHO, 1987).

Programmes

People programme

Parameter	Variable	Value	Source
People	my_human _density	0.1 people/m^2	The mean figure of the base and the last years of the social simulation [a]
Activity schedule	None	120 W/person	The schedule is set to always on and uses Honeybee's default value, corresponding to awake adult humans who are seated.

[a] The average number of bedrooms/people from the social simulation's base year is 0.72. For that same timestep, dwellings have 2 bedrooms in 43.74 m^2, so that to each bedroom corresponds 21.87 m^2. Thus, on the simulation's base year, there is 0.033 people/m^2. For the last year of the simulation (average number of bedrooms 3.87; average size of the dwelling 60.78 m^2 and average number of bedrooms/people 2.65), we get an average of 0.17 people/m^2.

Lighting programme

Parameter	Variable	Value	Source
Watts per area	watts_light	3.15 W/m^2	Considering a 23 W energy saver bulb in each modular room.
Return fraction	return_fract_	0	Default value.
Radiant fraction	_radiant_fract_	0.32	Default value.
Visible fraction	_visible_fract_	0.25	Default value.

Electrical equipment programme

Parameter	Variable	Value	Source
Watts per area	watts_eq	6.56 W/m^2	Approximate based on relevant and situated literature [b]
Radiant fraction	radiant_fract_	0	Default value.
Latent fraction	latent_fract_	0	Default value.
Lost fraction	lost_fract_	0	Default value.

[b] Oraiopoulos et al. (2022) consider 12W per "zone floor area" for lighting and a bit more than twice that figure for electrical equipment (25W per "zone floor area"). The same correlation between lighting and electrical equipment is used here.

Gas equipment programme

Parameter	Variable	Value	Source
Watts per area	watts_gas	2.94 W/m^2	ERCUE survey 2019-2020 [c]
Radiant fraction	radiant_fract_	0	Default value.
Latent fraction	latent_fract_	0	Default value.
Lost fraction	lost_fract_	0	Default value.

[c] According to the ERCUE survey 2019-2020 (De La Cruz, Salazar, & Coello, 2021), in the urban areas of Peru, the average consumption of LPG, measured in 10-kg bottle units, was 1.06 per month, equivalent to 0.48 MMBTU, or 140 640 Wh per month. Considering 730hrs in a month, this is equivalent to 193W, then divided by the maximum ground floor area (65.61 m^2).

Hot water programme

Parameter	Variable	Value	Source
Flow per area	my_hotw_flow	0.21 l/h-m^2	Peru's building standards code taking into account 2 people per room [d]
Target temperature	_target_temp_	60 ^oC	Default value.
Sensible fraction	_sensible_fract_	0.2	Default value.
Latent fraction	_latent_fract_	0.05	Default value.

[d] Peru's building code (RNE) section IS 0.10 (Ministerio de Vivienda, 2006), referring to water and sewage provision, mentions in Article 11 a daily allocation of hot water in residential buildings according to the number of bedrooms. Considering two persons per bedroom, this ranges between 40-60 litres per person per day. This study considers the mean between these two extremes (50 l/person-day or 2.08 l/person-hour) and the density of dwellers (0.1 people/m^2).

Infiltration programme

Parameter	Variable	Value	Source
Flow per external area	my_flow	0.0003 m^3/s per m^2 of façade	Honeybee's recommended value for an "average" building. The schedule is set to always on.

Honeybee "Room" configuration

Honeybee room settings

Conditioned	myConditioned	TRUE	It does not reflect the presence of HVAC systems, but it is used as a reference to achieve a standard "healthy" indoor environment.[e]
Roof Angle	my_roofangle	60	Default

[e] The ERCUE survey 2019-2020 reveals that the number of Peruvian households using appliances to climatise interior spaces is negligible, between 0.3% using gas-powered systems (De La Cruz, Salazar, & Coello, 2021) and 2% using electric systems (De La Cruz, Salazar, & Santos, 2021).

Apertures

Aperture by ratio configuration

Parameter	Variable	Value	Source
Ratio	my_aper_ratio	0.25	Empirical observation.
Subdivide	my_subdivision	FALSE	Only one window per surface.
Windows height	my_win_hgt	None (Overridden by ratio)	-
Sill height	my_sill_hgt	None (Overridden by ratio)	-
Horizontal separation	my_h_sep	None (Overridden by ratio)	-
Vertical separation	my_v_sep	None (Overridden by ratio)	-
Operable	my_oper_apert	FALSE	Ventilation assumed from infiltration only.

Honeybee Simulation Parameters

Honeybee Simulation Parameters

Parameter	Variable	Value
North	my_north	180^o
Run period	my_run_period	Year (default)
Daylight saving	my_daylight_saving	False (default)
Holidays	my_holidays	None (default)
Start day of the week	my_startday	Sunday
Timesteps per hour	my_tstep	1
Terrain	my_terrain	Urban
Simulation control	my_simcontrol	Perform a sizing calculation but only run the simulation for the RunPeriod (default).
Shadow calculation	my_shadow_calc	"Full exterior with reflections, polygon clipping, periodic, for every day of the year (365)."
Sizing	my_sizing	None

Honeybee Simulation outputs

Honeybee Simulation outputs

Parameter	Variable	Value
Zone energy use	my_zone_eu	TRUE
HVAC Energy use	my_hvac_eu	FALSE
Gains and loses	my_gains_loses	FALSE
Comfort metrics	my_comfort_m	FALSE
Surface temperature	my_surf_temp	FALSE
Surface energy flow	my_surf_e_flow	FALSE
Load type	my_load_type	FALSE
Report frequency	my_freq_type	Annual

EnergyPlus weather files

EPW Files

City	Location	EPW file	Source
Juliaca	15.5° S 70.13° W 3824 m.a.s.l.	PER_PUN_Juliaca-Manco.AP.847350_TMYx.2007-2021.epw	climate.onebuilding.org
Piura	5.18° S 80.65° W 55 m.a.s.l.	PER_PIU_Piura-Iberico.Intl.AP.844010_TMYx.2007-2021.epw	climate.onebuilding.org
Tarapoto	6.48° S 76.37° W 356 m.a.s.l.	PER_SAM_Tarapoto-Paredes.AP.844550_TMYx. 2007-2021.epw	climate.onebuilding.org

Appendix 12

Socio-economic modelling settings

Socio-economic model settings

Parameter	Variable	Value
Seed for pseudorandom number generator.	set.seed()	229
Total number of synthetic agents in the simulation.	total_ag	48
Probability of agents belonging to the socio-economic level B.	sel_b	0.2
Probability of agents belonging to the socio-economic level C.	sel_c	0.3
Probability of agents belonging to the socio-economic level D.	sel_d	0.6
Number of years of the simulation.	nr_yrs	25

Micro-investment model settings

Parameter	Variable	Value	Source
Limits defining the categories of "need" in bedrooms/people.	need_thld	[-Inf, 0.5), [0.5, 1),[1, Inf)	International consensus on minimum adequate living conditions and proposed good standard [1]
Times the household's yearly financial balance (when positive) that can be taken as a loan.	loan_cap	4	In most places in the world, lenders use the rule of thumb of capping mortgages at 4.5 times the income and the loan-to-income ratio to 30%-40%.
The annual interest rate charged to loans.	interest	0.15	Average in 2015, according to Peru's banking and insurance superintendency [2].

The maximum fraction of a "no-need" household's accumulated savings that still triggers construction if it pays for the construction cost.	Noneed_thr	0.01	Empirical, avoids over-extension.
The maximum fraction of the household accumulated savings allowed to be used to pay for construction or for debt repayment.	invs_max	0.6	Empirical, avoids depletion of savings.
Initial size of the dwelling (in number of actual bedrooms).	dw_size_0	2	Minimum requirement for "Techo Propio" dwellings.
Building cost at year 0 (2015)	bld_cst_0	10120	Referential figure, calculated for 2015, see Appendix 9.
Annual inflation applied to construction costs.	inflt	0.028	Average calculated using Peru's Official Building Unit Values 2015-2019, see Appendix 10.

[1] The definition of overcrowding, according to UN-habitat, is to have more than three persons per potential bedroom. The international consensus to define adequate living conditions is two persons or less per potential bedroom. The gold standard would be to have at least one potential bedroom per person so there are remaining spaces to accommodate more specialized activities. Check Chapter 6, micro-investment model, for more details.

[2] The average interest rate for mortgages in Peru in June 2015 was 8.38%. The average for personal loans was 24.33% (Superintendencia de Banca Seguros y AFP, 2024).

Appendix 13

Geometry Generation script settings

GG script settings

Parameter	Variable	Value
Number of agents (dwellings) in the neighbourhood (data read from the socio-economic model).	nr_agents	48
Coordinates (in meters) of the origin of the neighbourhood in the X axis.	lotorigx	0
Coordinates (in meters) of the origin of the neighbourhood in the Y axis.	lotorigy	0
Number of blocks in the X axis.	my_nrblocksX	3
Number of blocks in the Y axis.	my_nrblocksY	2
Width of the streets between blocks (in meters).	my_streetwith	11
Size (in modules) of the buildable area within the lot in the X axis.[a]	lotsizex	3
Size (in modules) of the buildable area within the lot in the Y axis.[a]	lotsizey	3
Size of the module (in meters).	my_module	2.7
Coordinates (in modular units, within the lot) of the origin of the dwelling seed in the X axis.	coreorigx	0
Coordinates (in modular units, within the lot) of the origin of the dwelling seed in the Y axis.	coreorigy	0
Size (in modular units) of the dwelling seed in the X axis.	coresizex	1
Size (in modular units) of the dwelling seed in the Y axis.	coresizey	1
Maximum height allowed for vertical development (in modular units).	maxheight	3
The number of lots that make one side of a block.	NrlotsOnSide	(nr_agents/2) / n_blocks
If True, the blocks are formed by mirrored lots. If False, they are formed by a single line of lots.	doubleside	TRUE
If True, the side lots are generated on the X axis direction, if False, they are generated on the Y axis direction.	mainsideXorY	TRUE
Radius (in meters) of a circle drawn from the centre of lots and used to identify the neighbours of every dwelling.	radius_neighid	my_module * 4
Sequence of actions to set the initial module.	basic_mod_stps	[2, 1, 2, 5, 7]

[a] By default, the lots' buildable areas have a setback from the back boundary of the lot equivalent to 1 module and a setback from the street (front boundary of the lot) of 0.5 modules.

Appendix 14

Multi-agent Q-learning settings

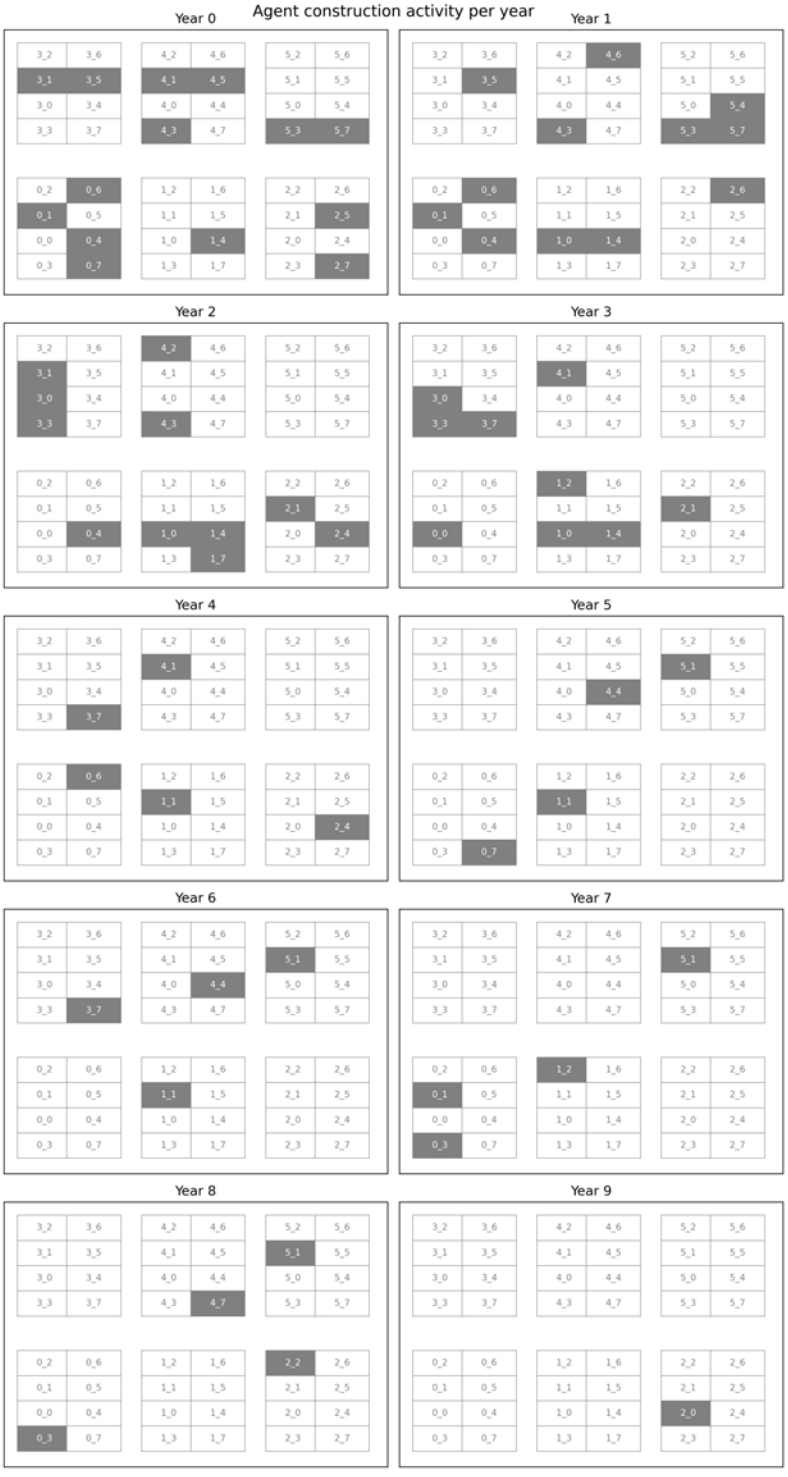
Multi-agent Q-learning settings

Parameter	Variable	Value
The tolerance within the absolute objective to still consider awarding the maximum reward to a solution.[a]	tolerance	0.01
The number of years in which the simulation will run (taken from the socio-economic model).	years	25
The maximum number of episodes allowed for each learning process.	episodes	200
The maximum number of steps within an episode. In this model, this is related to the number of rooms built in a year, currently capped at 1.	steps	1
This reward is assigned to an agent when they achieve the goal. The maximum penalisation (awarded on "deadly" states) is the negative of this value.	GOAL_REWARD	25
Penalisation for each step passed in which a "terminal" state is not encountered.	TIME_PENALTY	-1
"The discount factor, which determines the importance of future rewards."	DISCOUNT	0.9
The lowest learning rate achievable at the last episode allowed at the current learning process.	lowest_lr	0.1
The lowest epsilon value achievable at the last episode allowed at the current learning process.	lowest_ep	0.005
Fraction of the maximum allowed episodes in the learning task that have to be evaluated to determine a stop of the task.	lookback_f	0.1
Proportion of the latest episodes evaluated that need to be successful (maximum reward) so that the learning task stops.	stop_prop	0.75

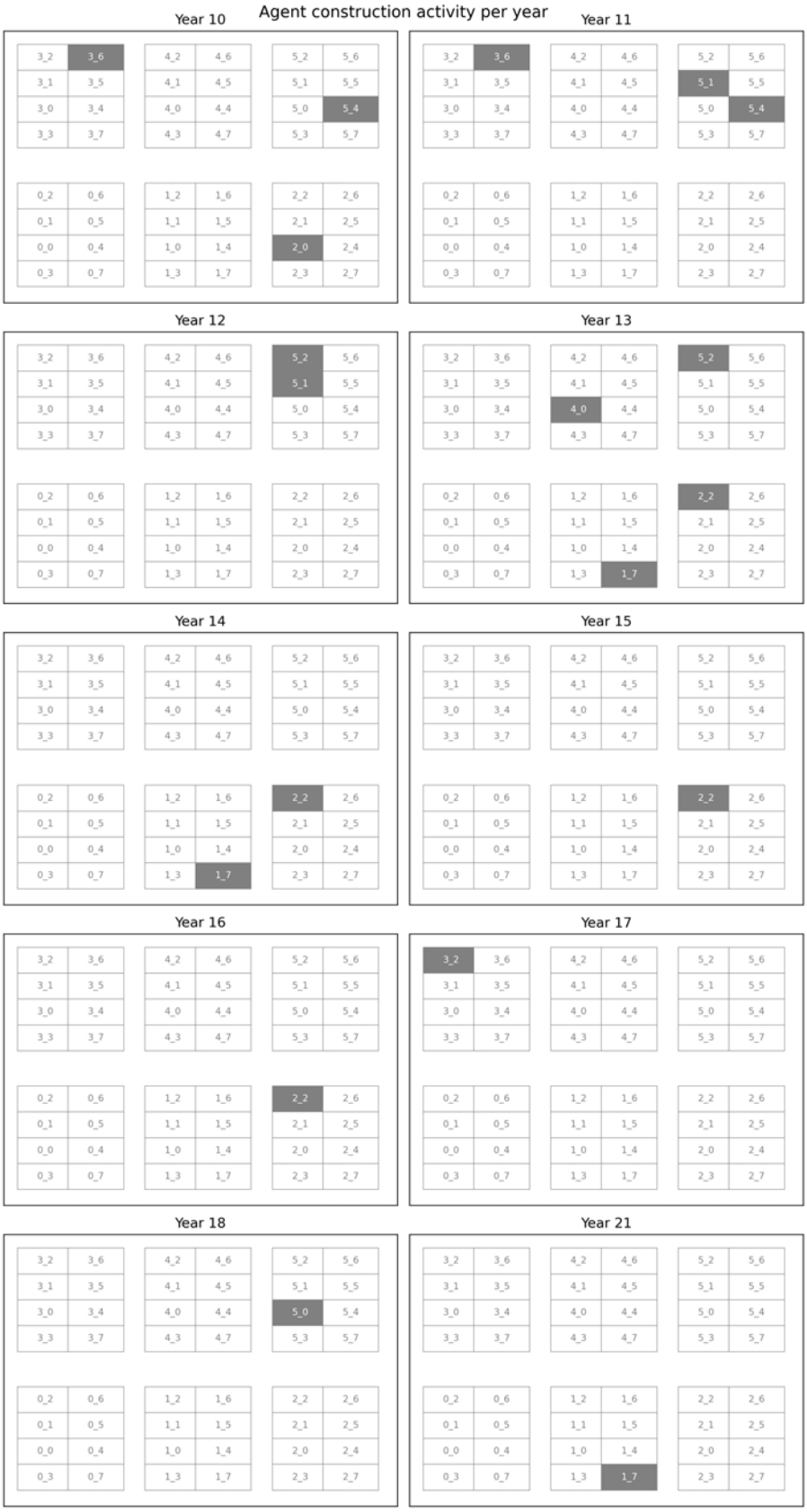
[a] It has to be noted that, as tolerance bounds a "negotiation space", it is possible to apply individualized tolerances to agents, so that individual attitudes towards negotiation are mimicked. Nevertheless, in this thesis, tolerances are simply used to facilitate the learning process and are thus universal.

Appendix 15

Agent activity on space by year



Construction activity on years 0-9



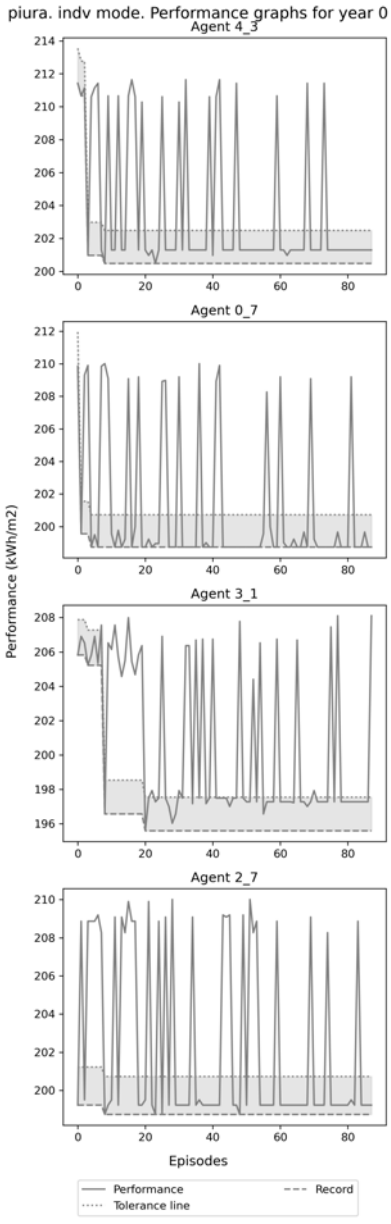
Construction activity on years 10-21

Construction activity on years 22-23

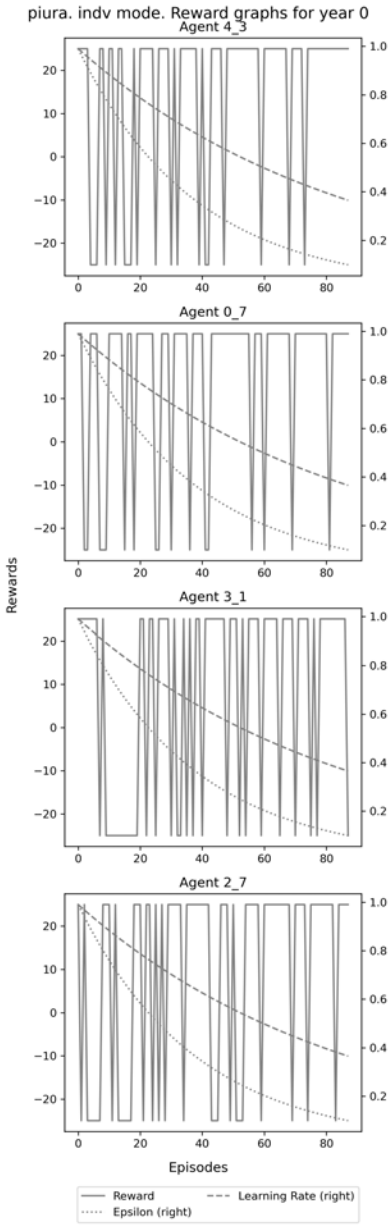
Appendix 16

Learning process by agent for each simulation year (Piura, minimisation, competitive)

(a) Performance



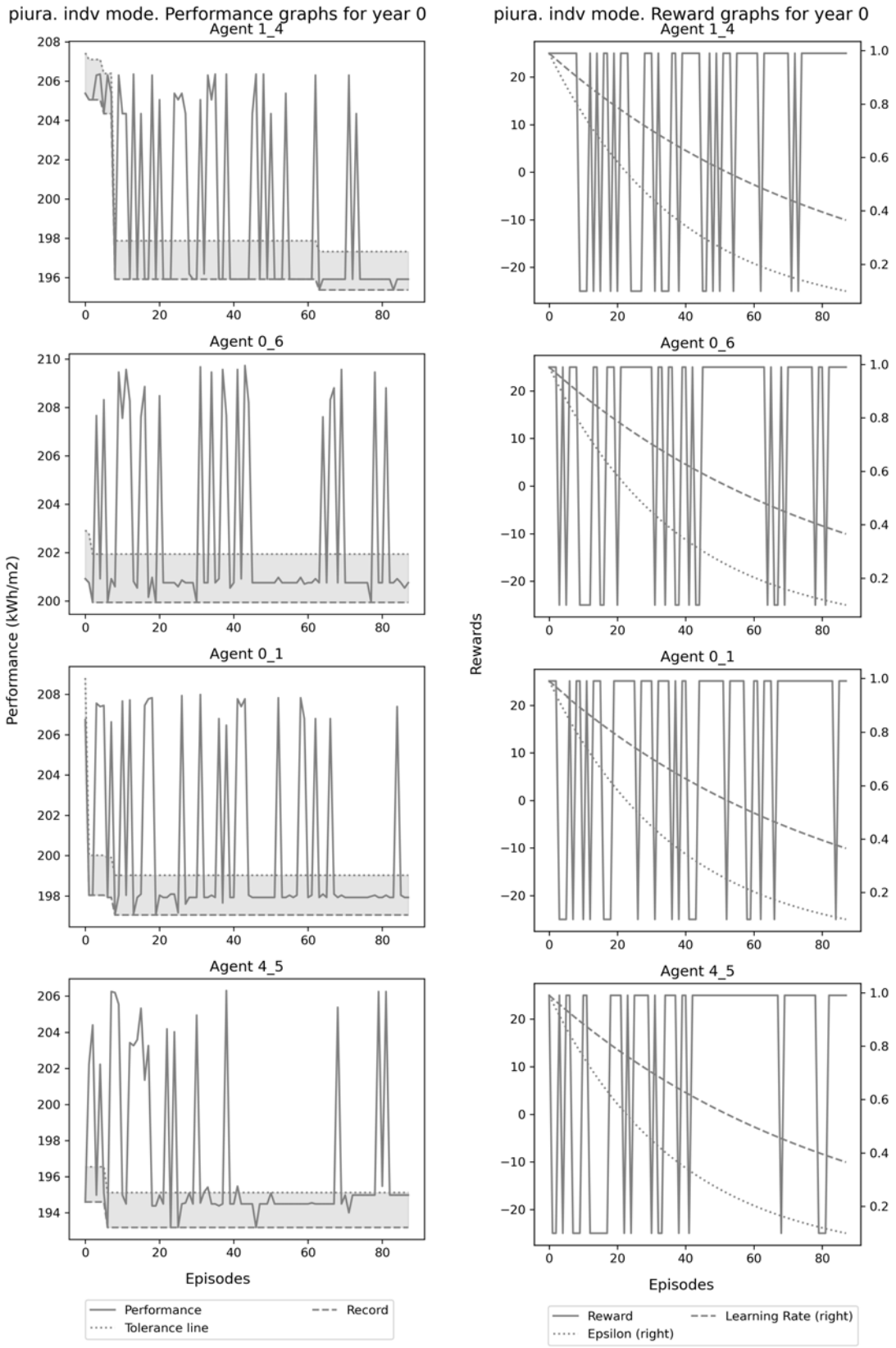
(b) Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (1)

(c) Performance

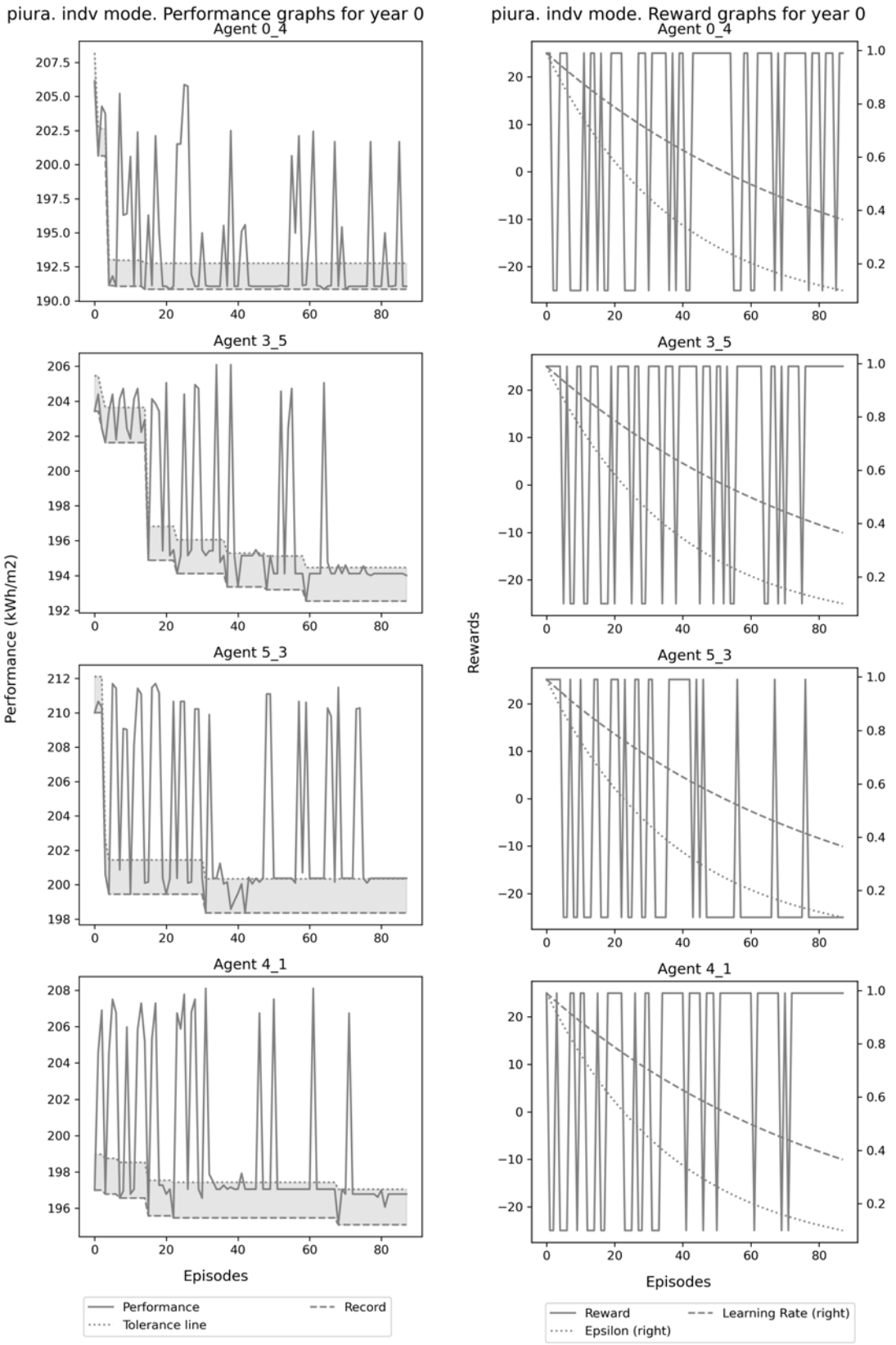
(d) Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (2)

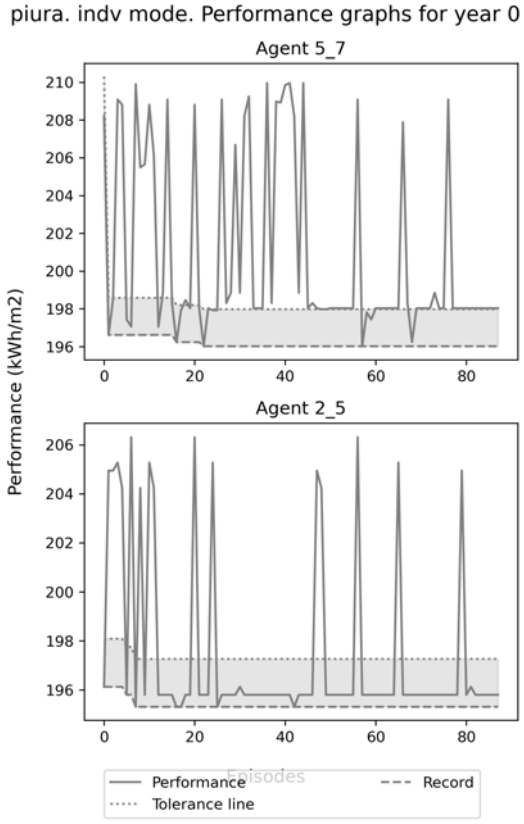
(e) Performance

(f) Rewards

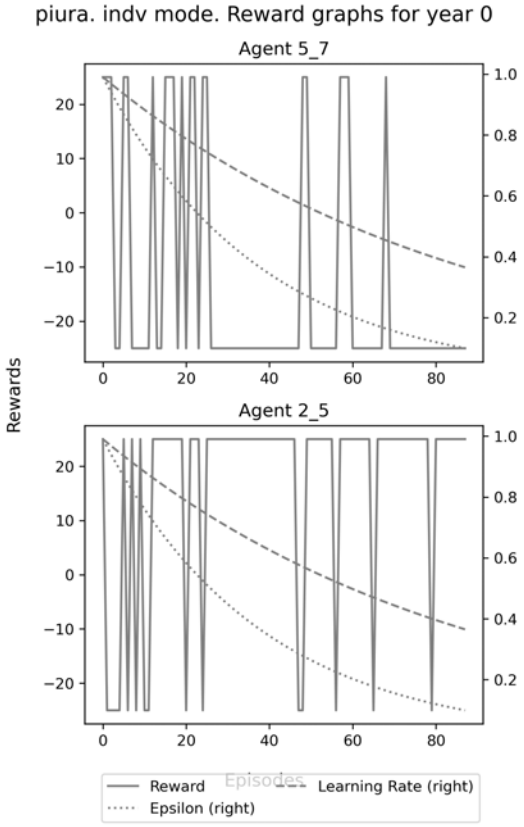


Learning process of all agents in Piura Competitive scenario for energy minimisation (3)

(g) Performance



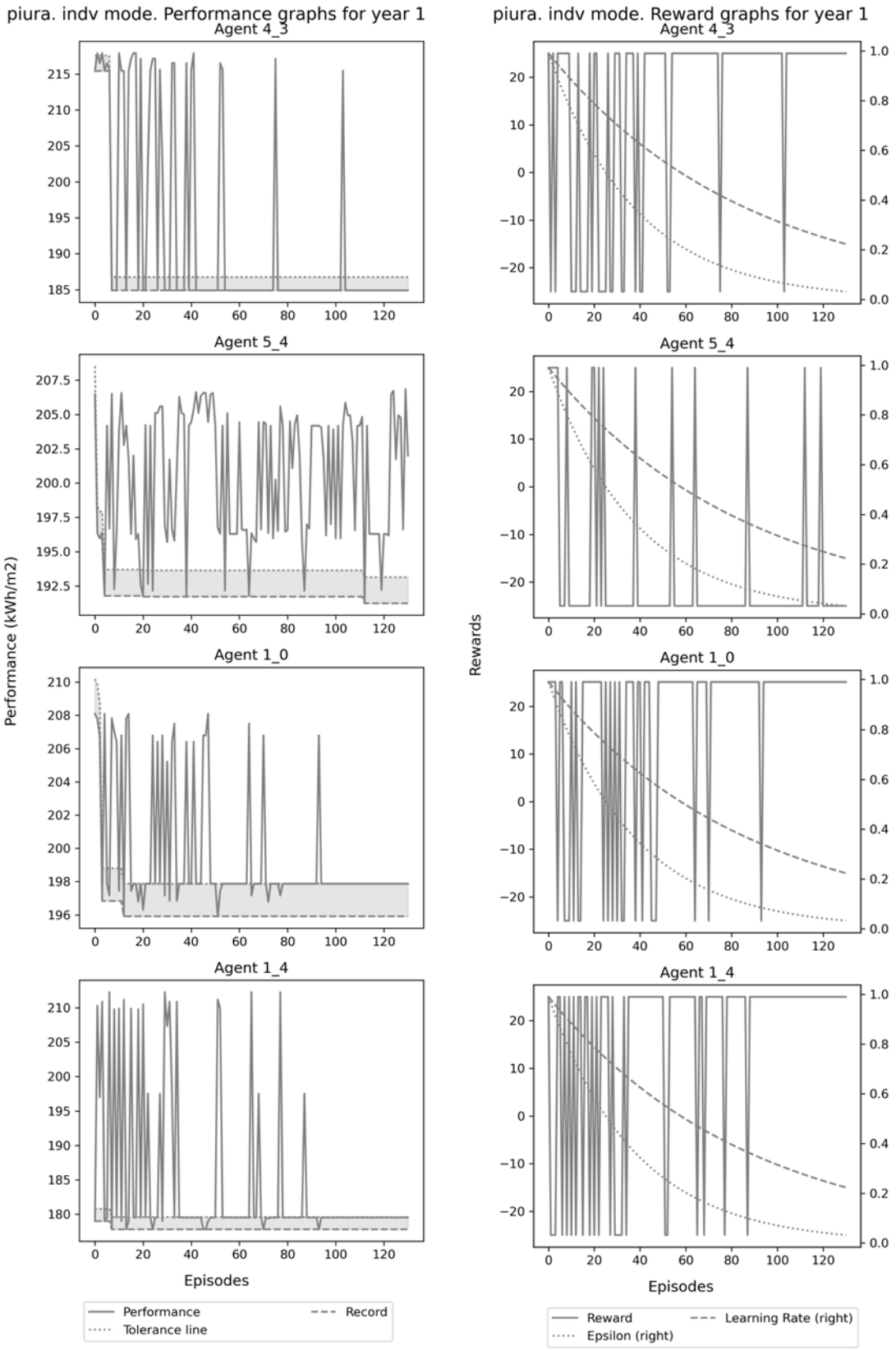
(h) Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (4)

(i) Performance

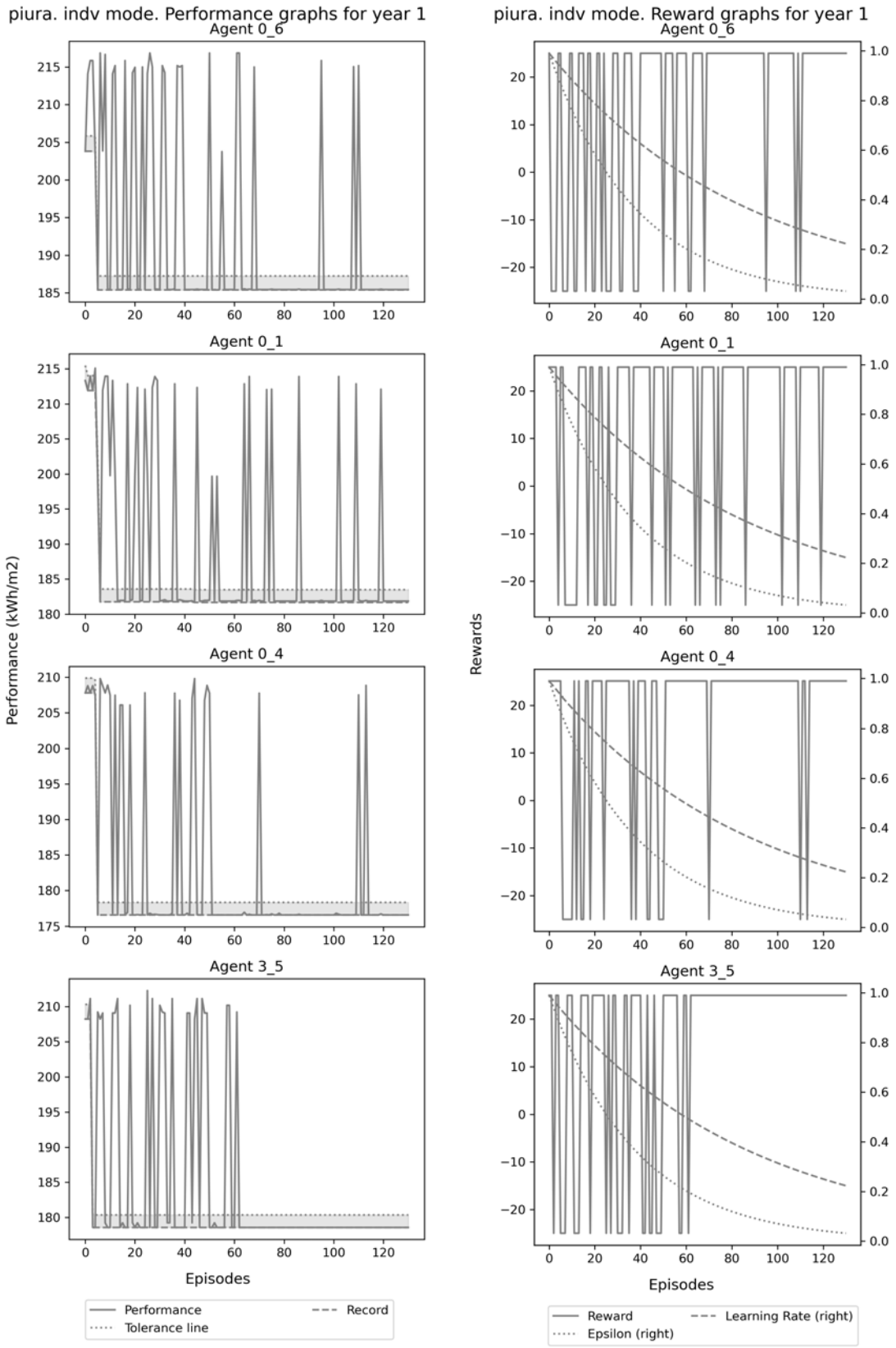
(j) Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (5)

(k) Performance

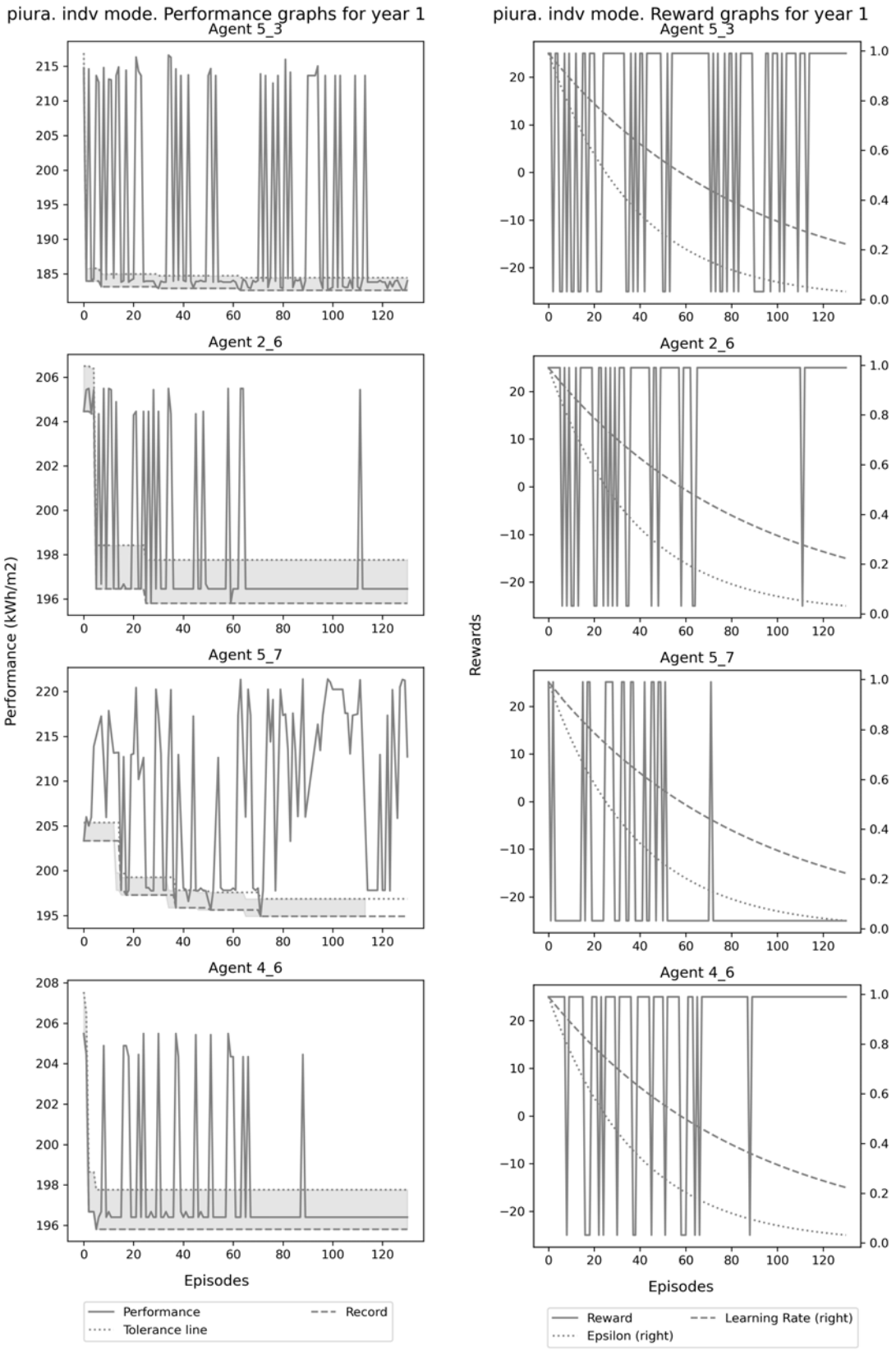
(l) Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (6)

(m) Performance

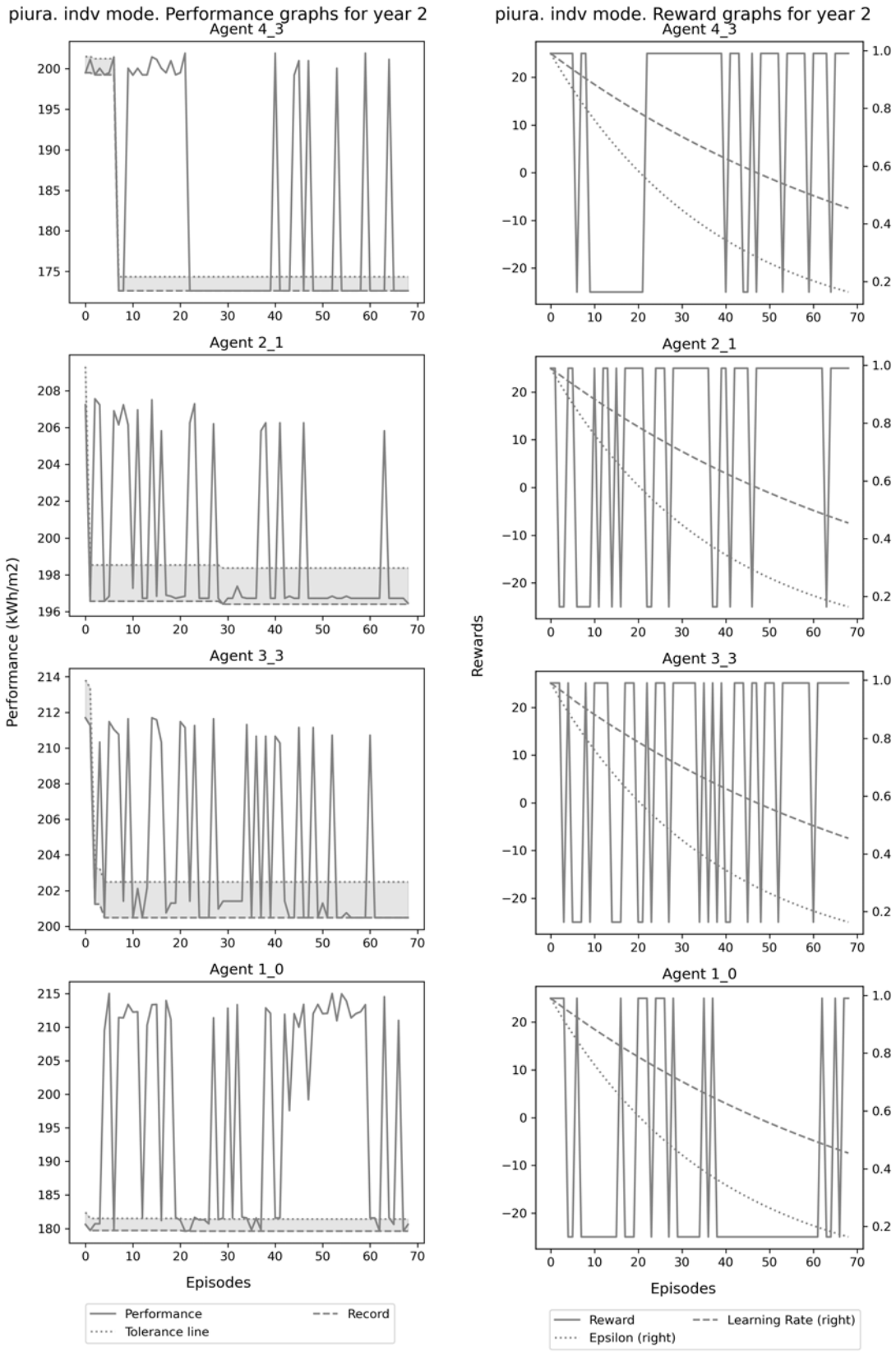
(n) Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (7)

(o) Performance

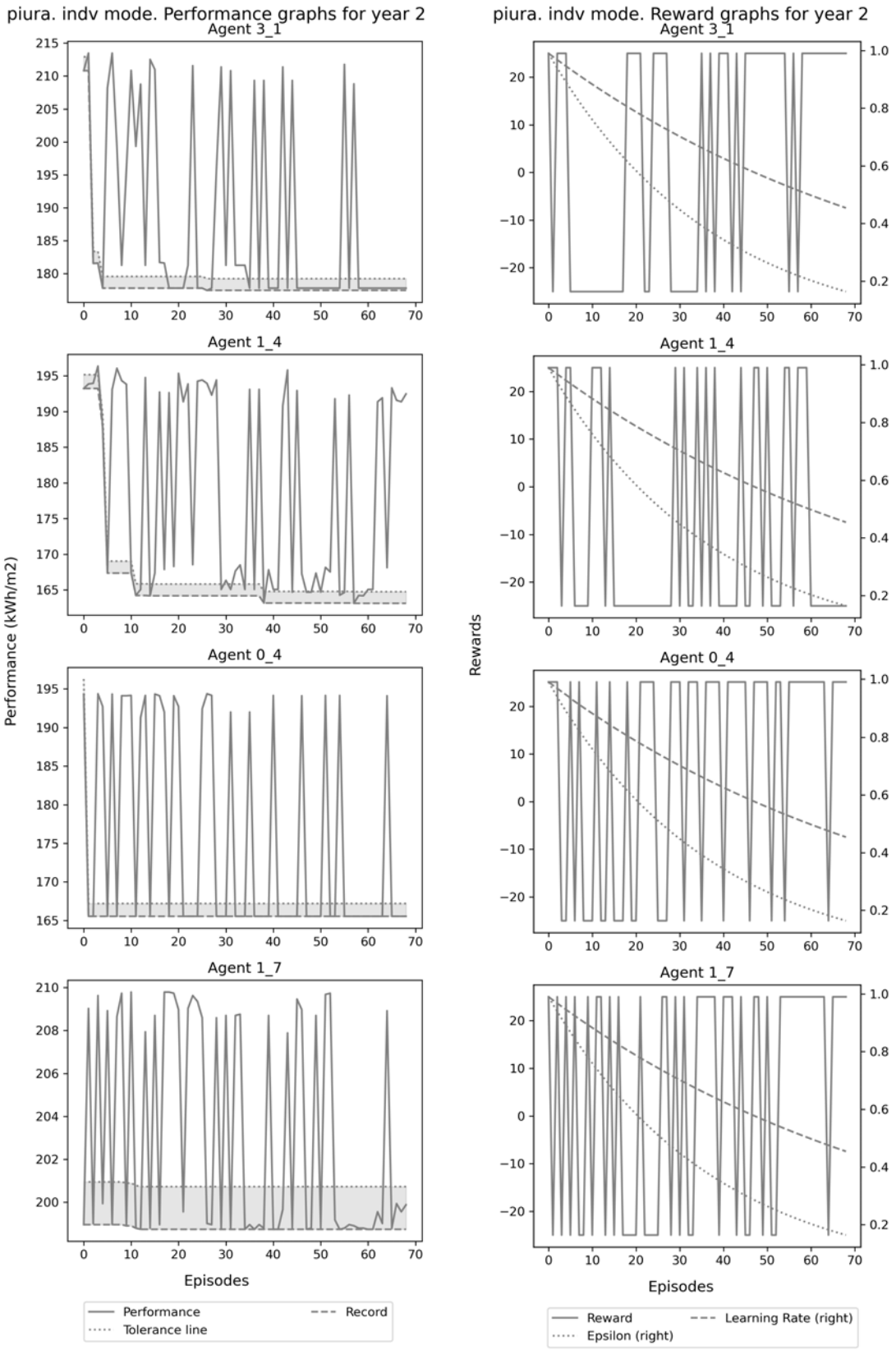
(p) Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (8)

(q) Performance

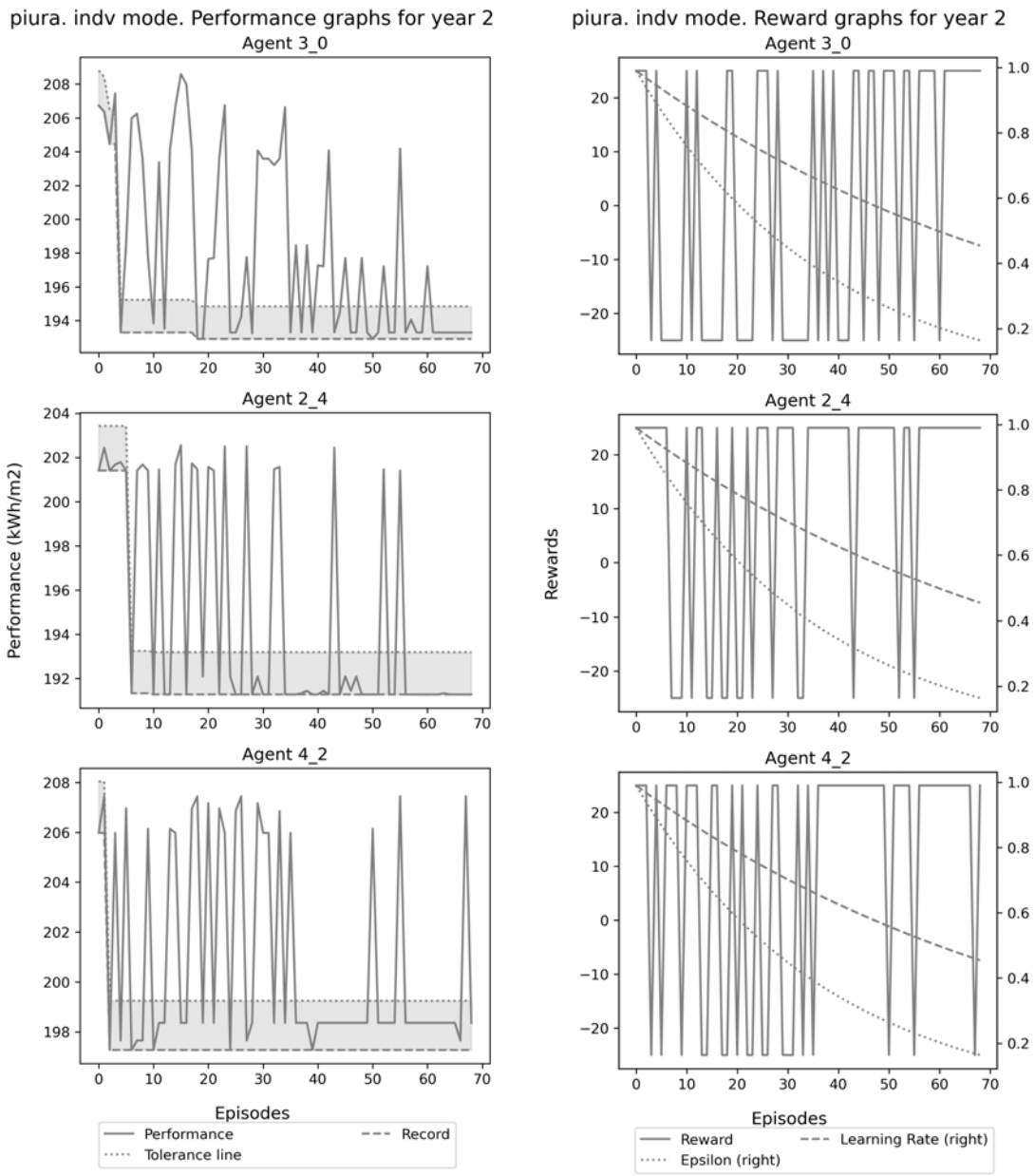
(r) Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (9)

(s) Performance

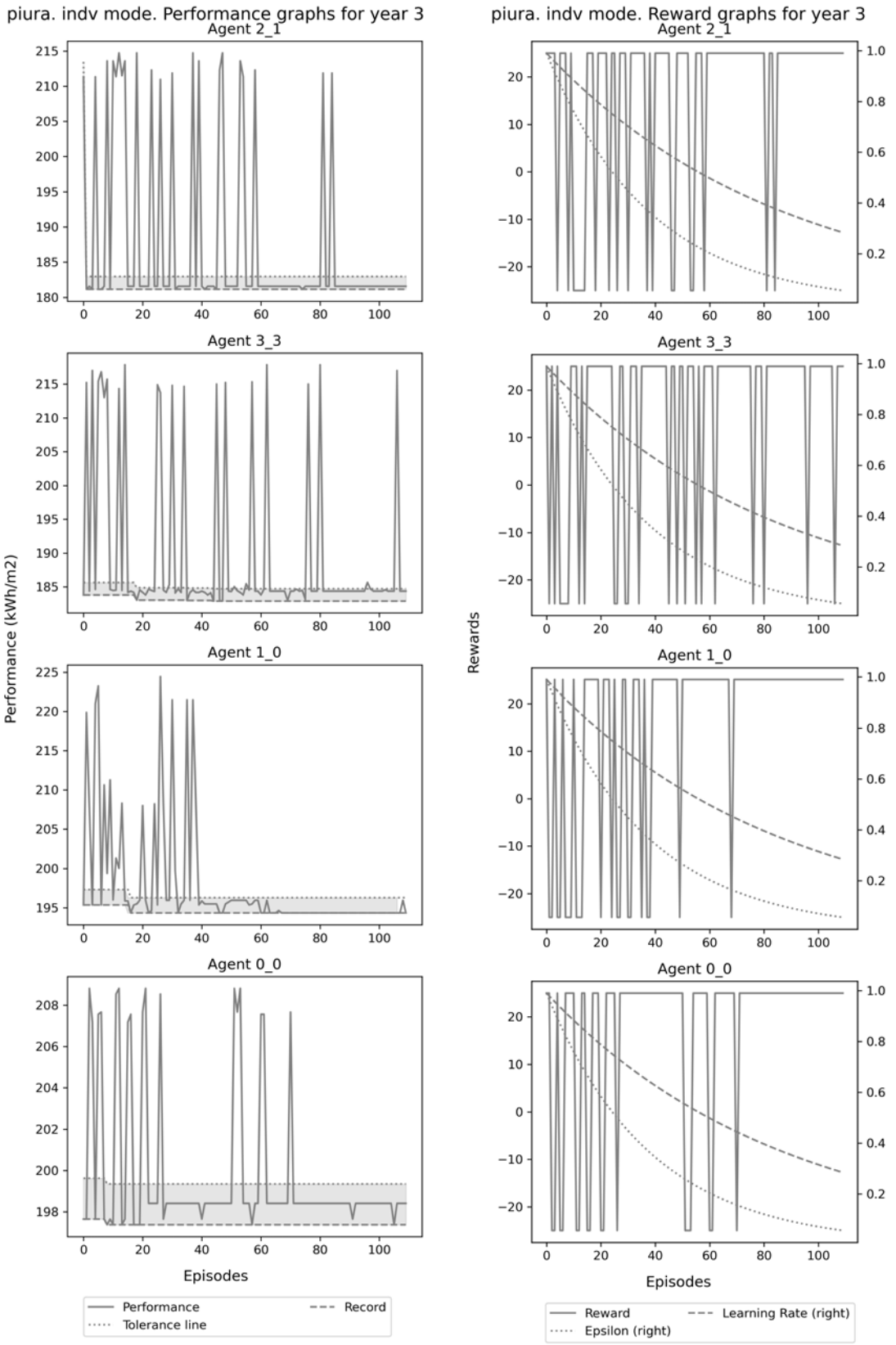
(t) Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (10)

(u) Performance

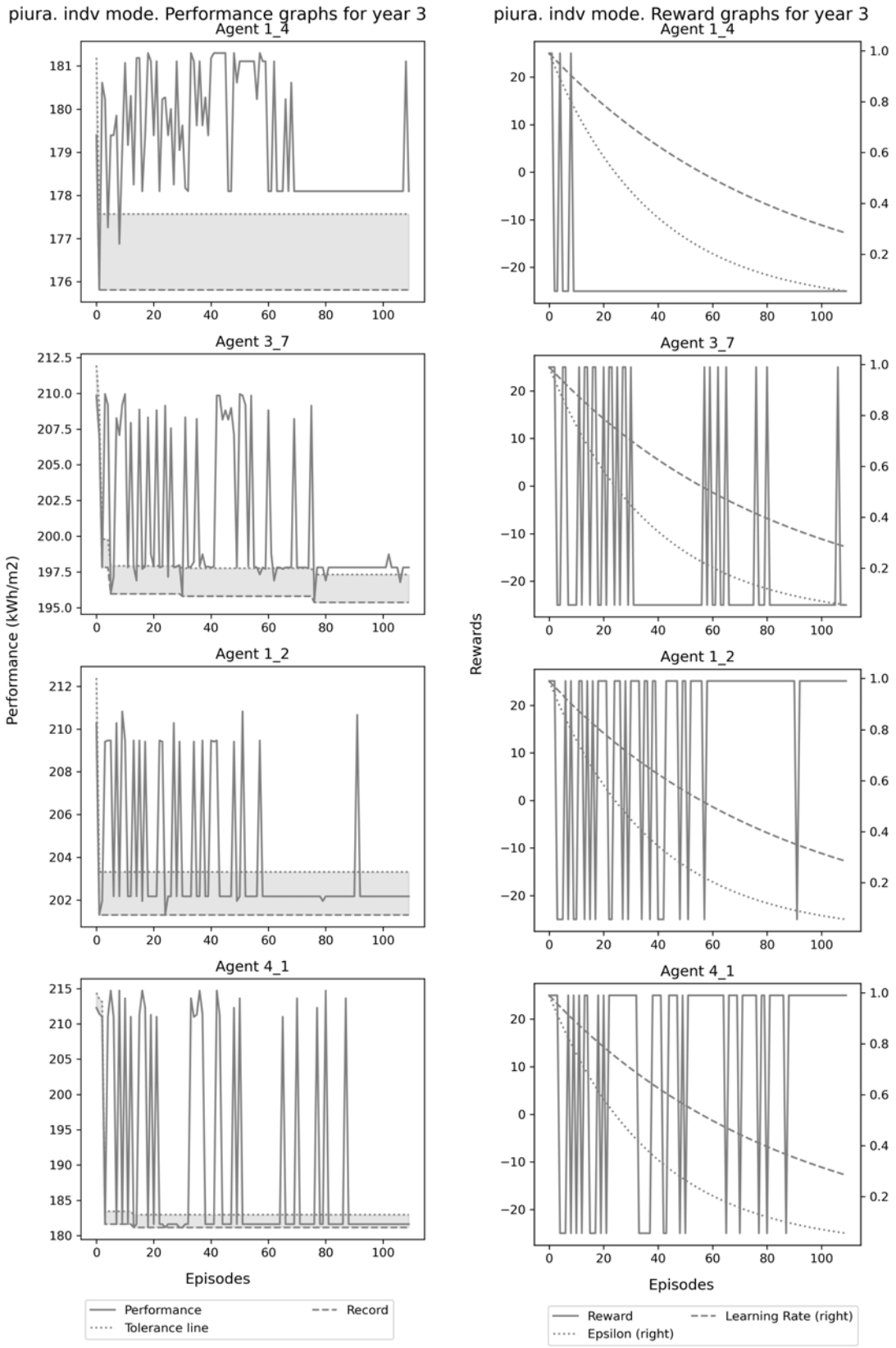
(v) Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (11)

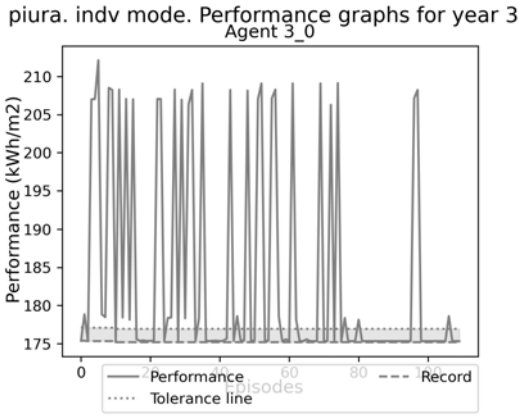
(w) Performance

(x) Rewards

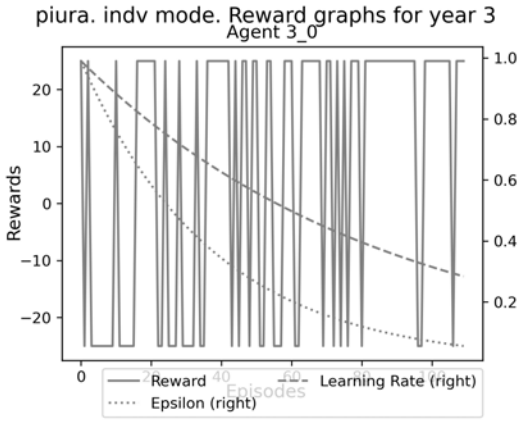


Learning process of all agents in Piura Competitive scenario for energy minimisation (12)

(y) Performance



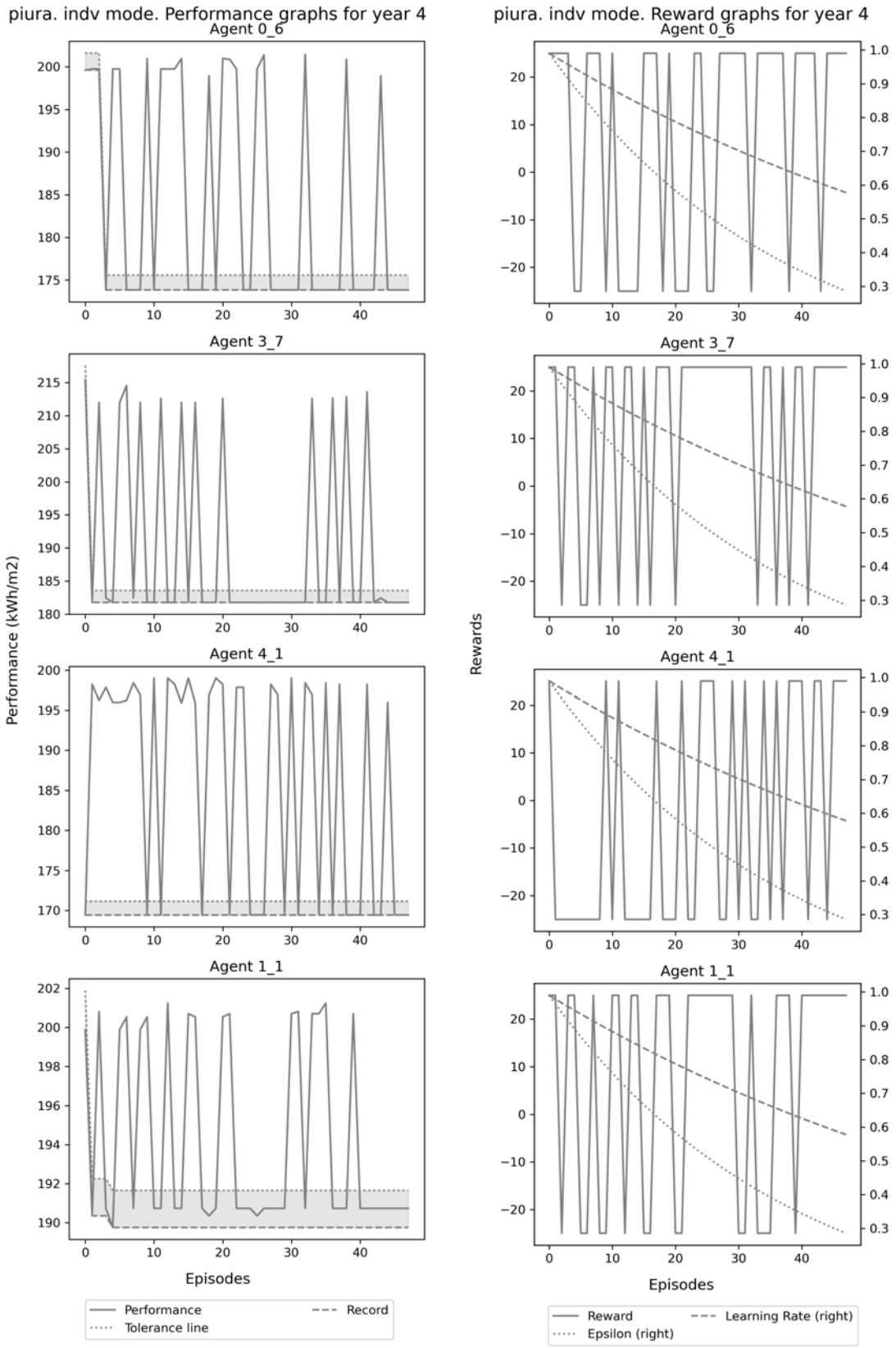
(z) Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (13)

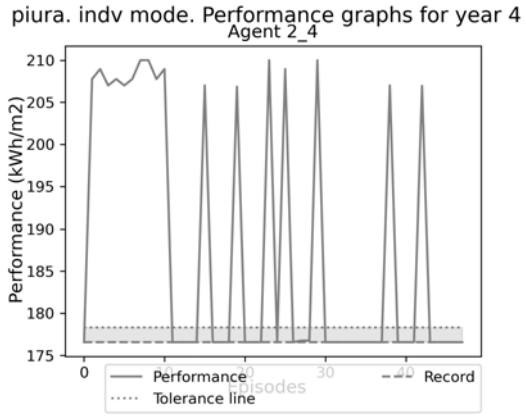
() Performance

() Rewards

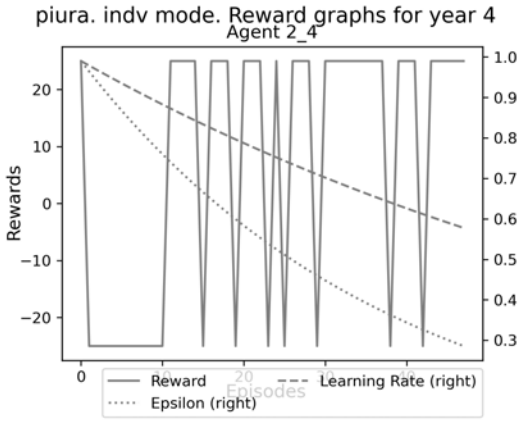


Learning process of all agents in Piura Competitive scenario for energy minimisation (14)

() Performance



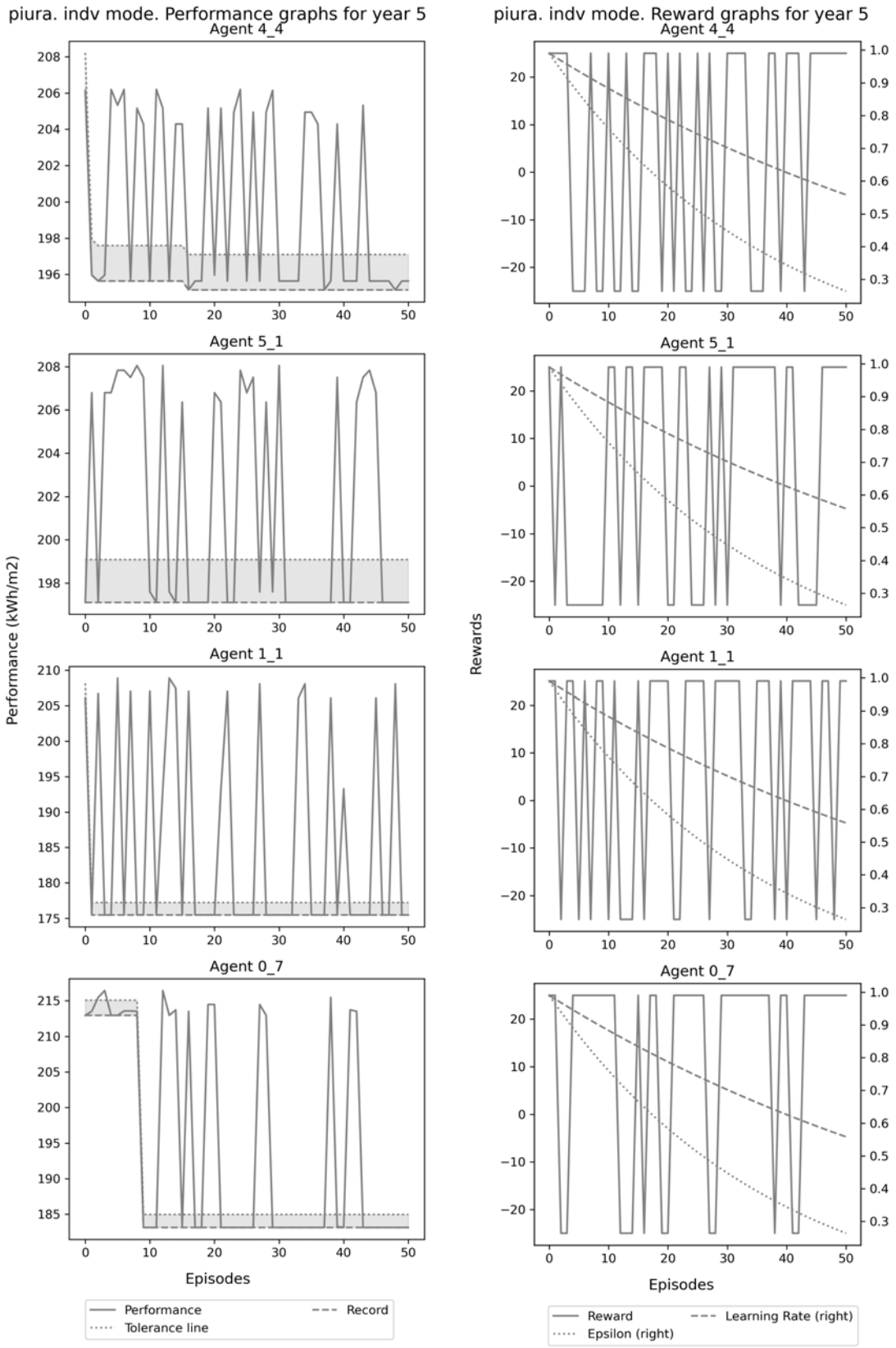
() Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (15)

() Performance

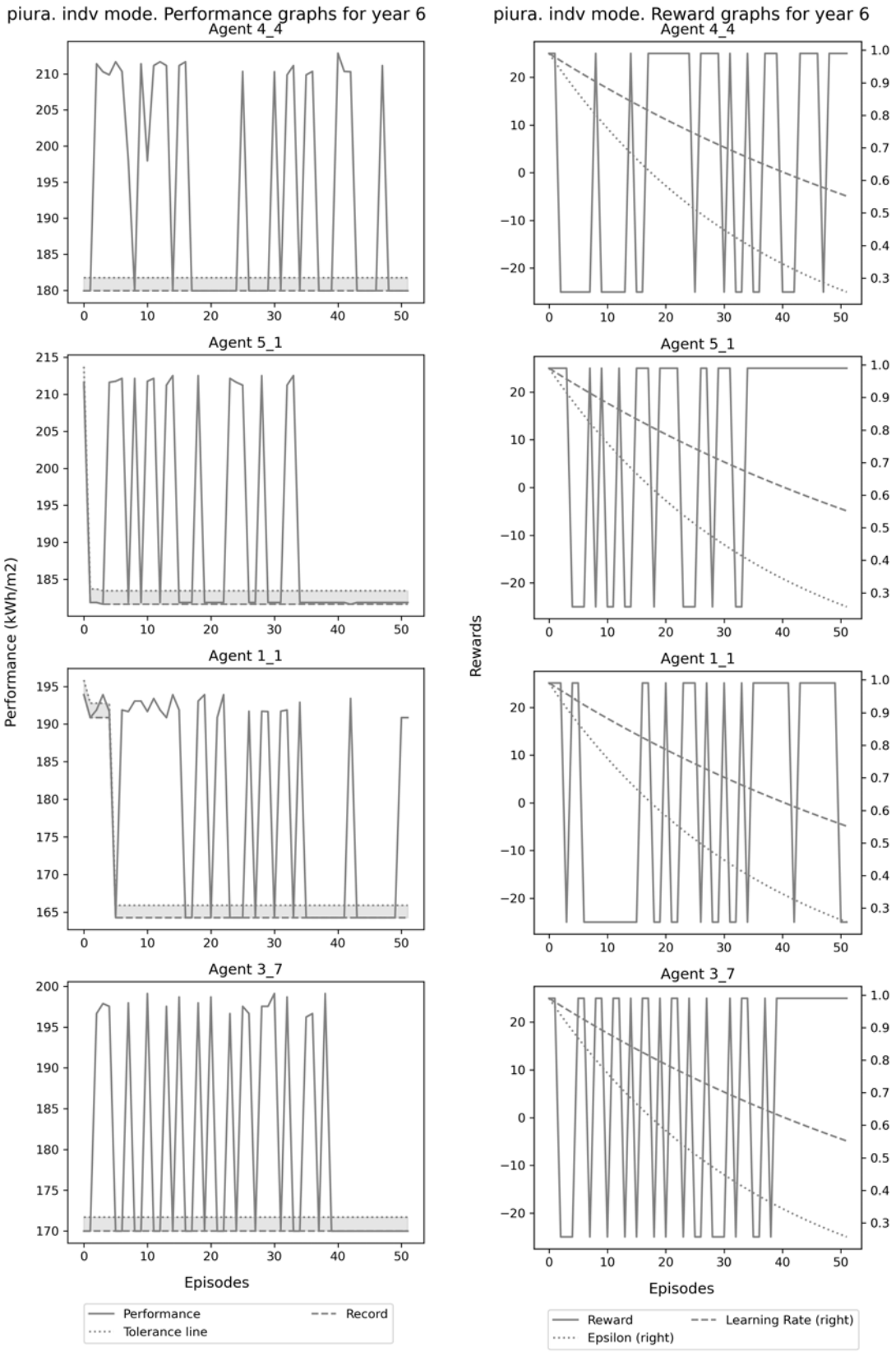
() Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (16)

() Performance

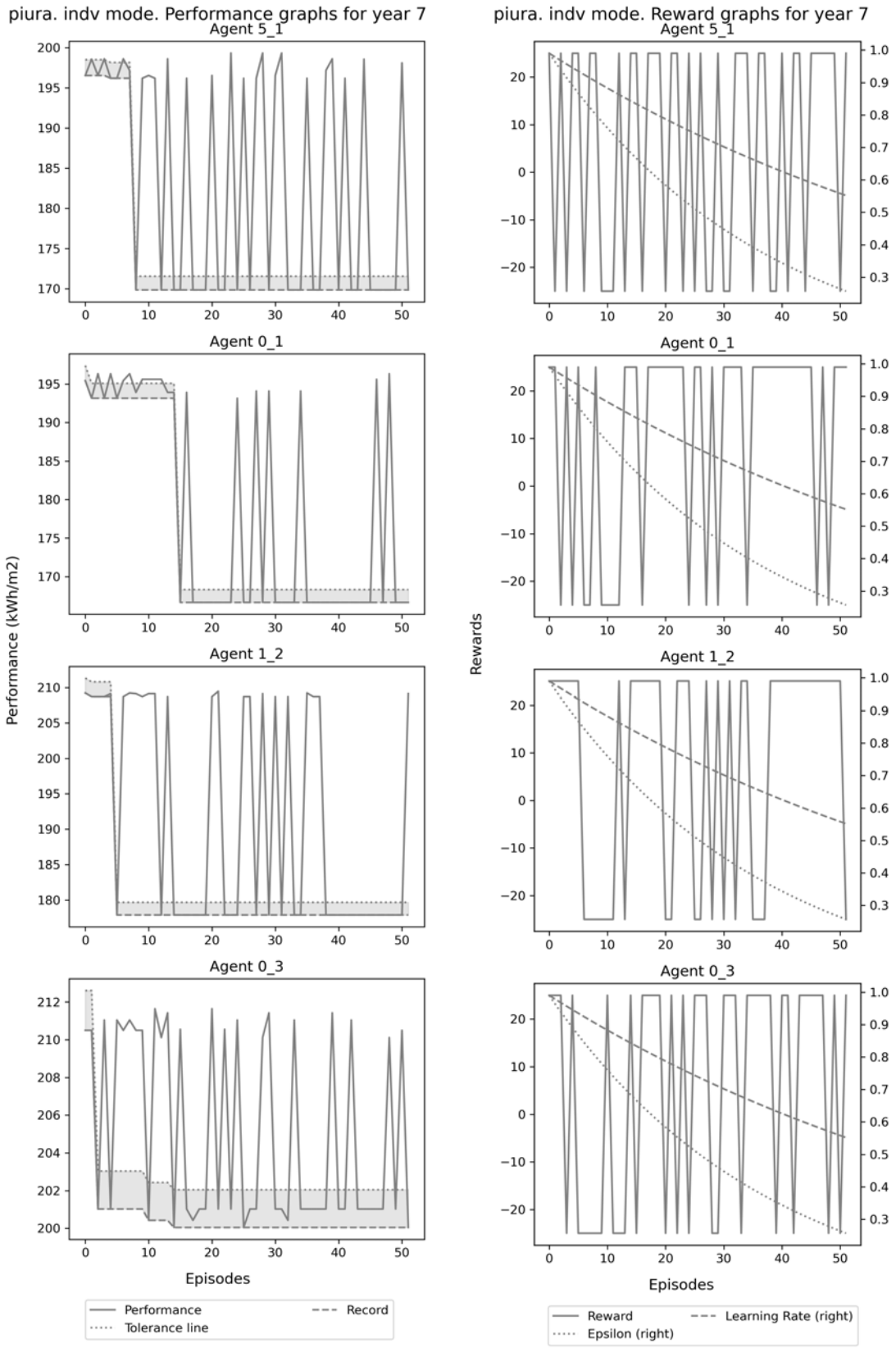
() Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (17)

() Performance

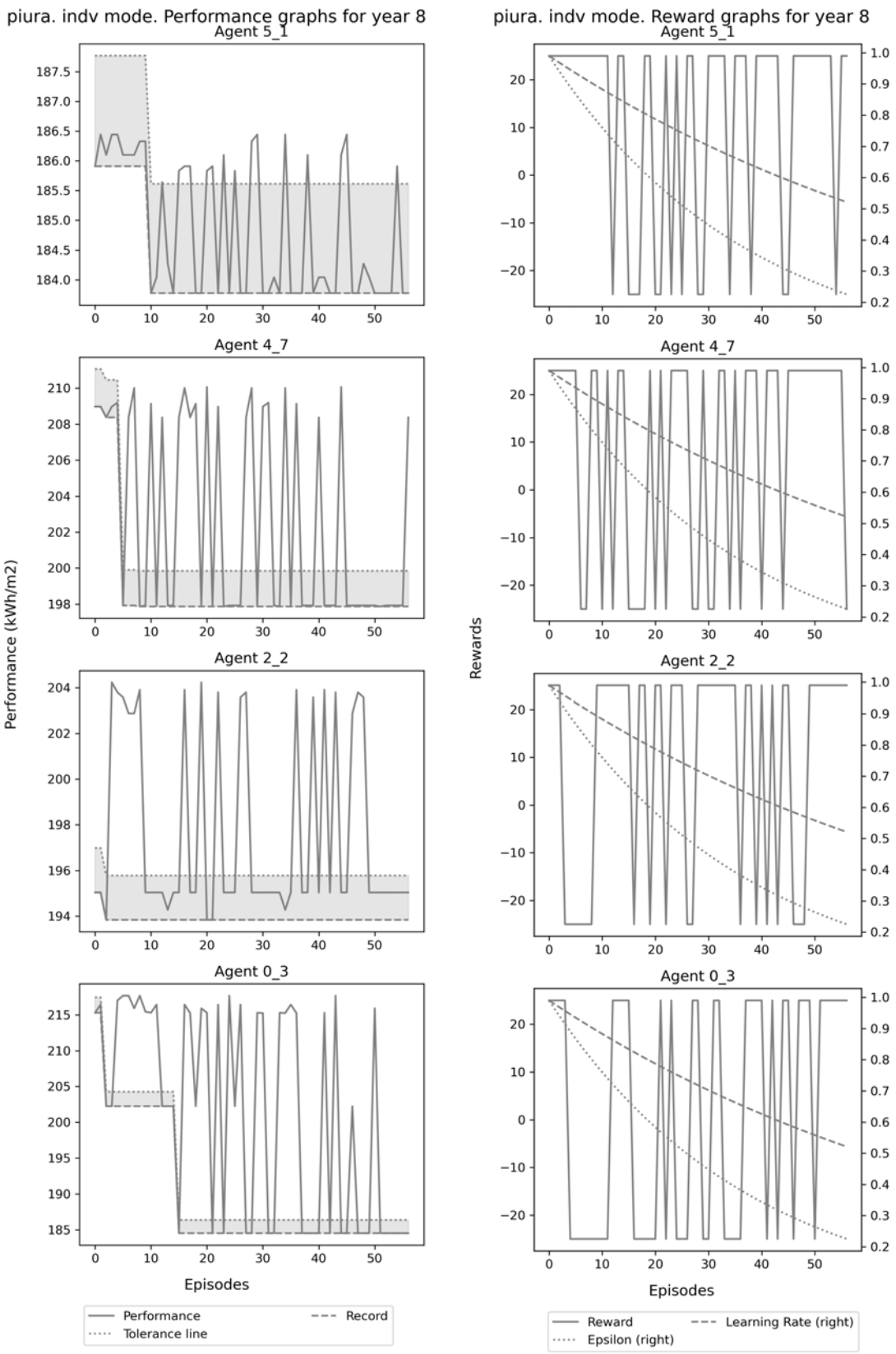
() Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (18)

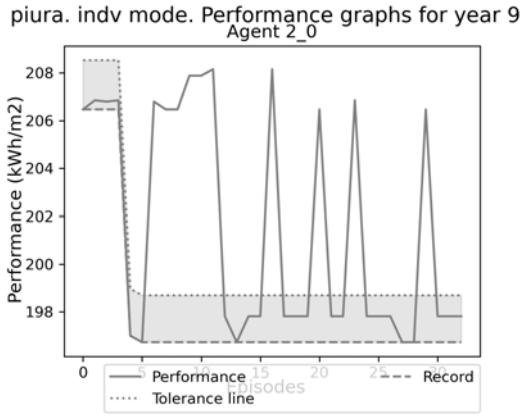
() Performance

() Rewards

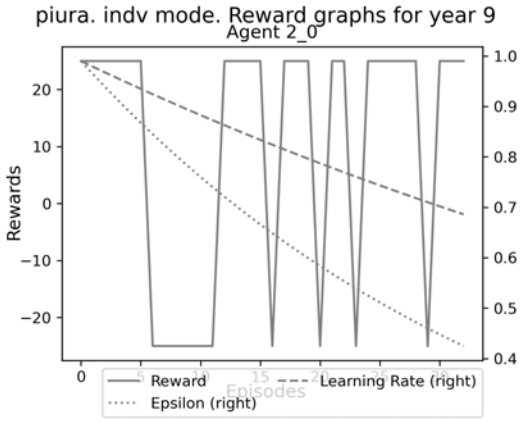


Learning process of all agents in Piura Competitive scenario for energy minimisation (19)

() Performance



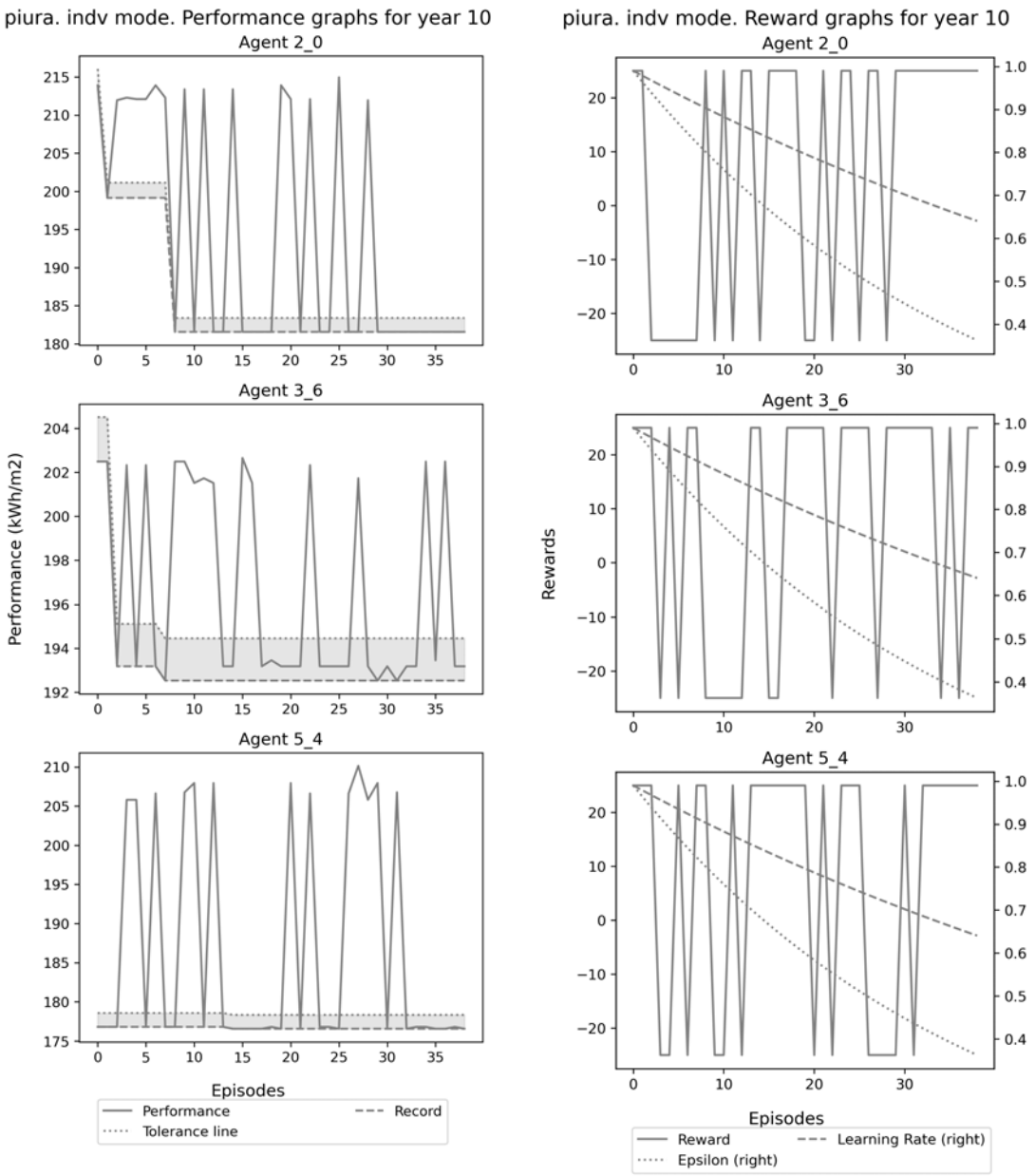
() Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (20)

() Performance

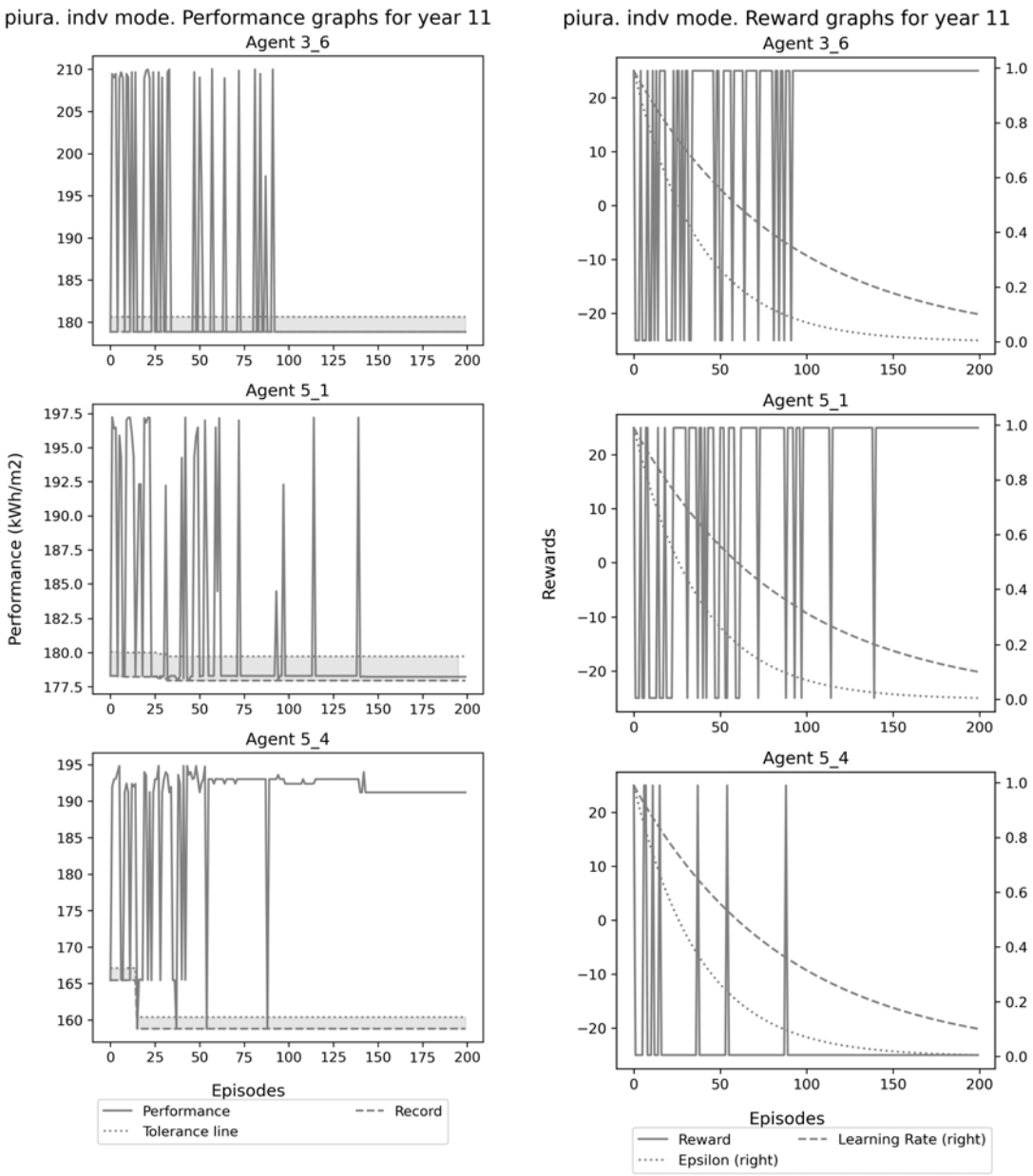
() Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (21)

() Performance

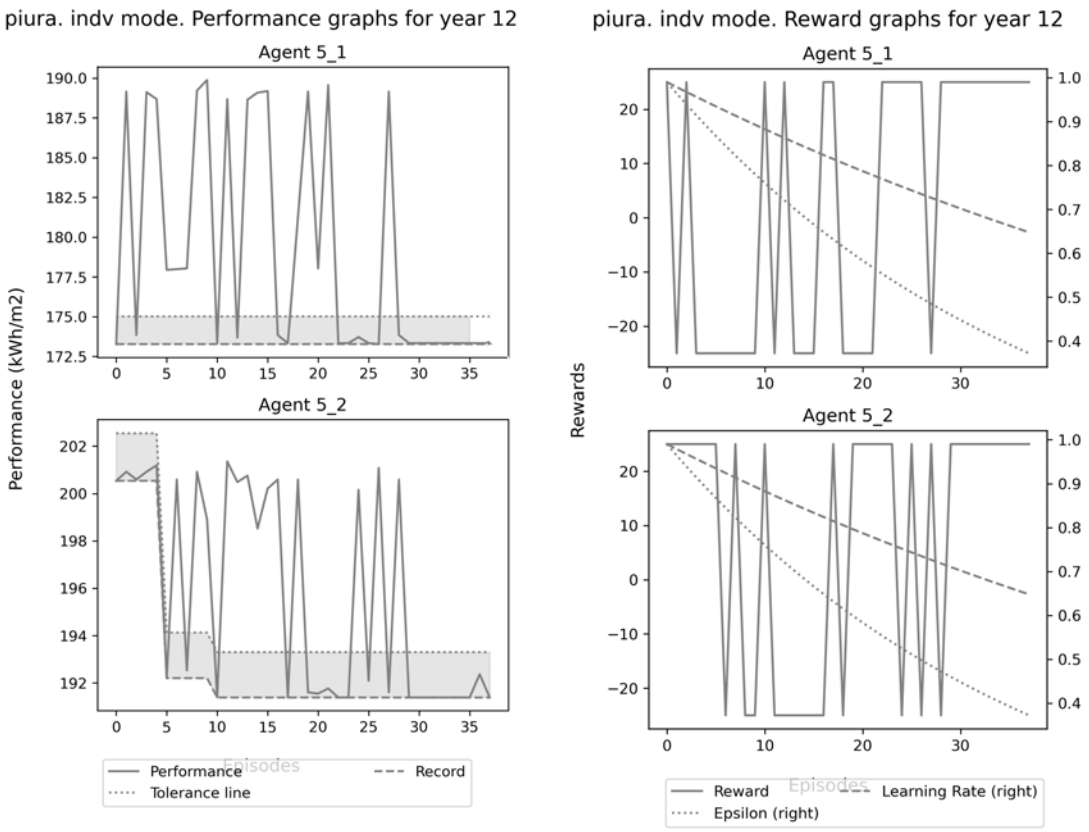
() Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (22)

() Performance

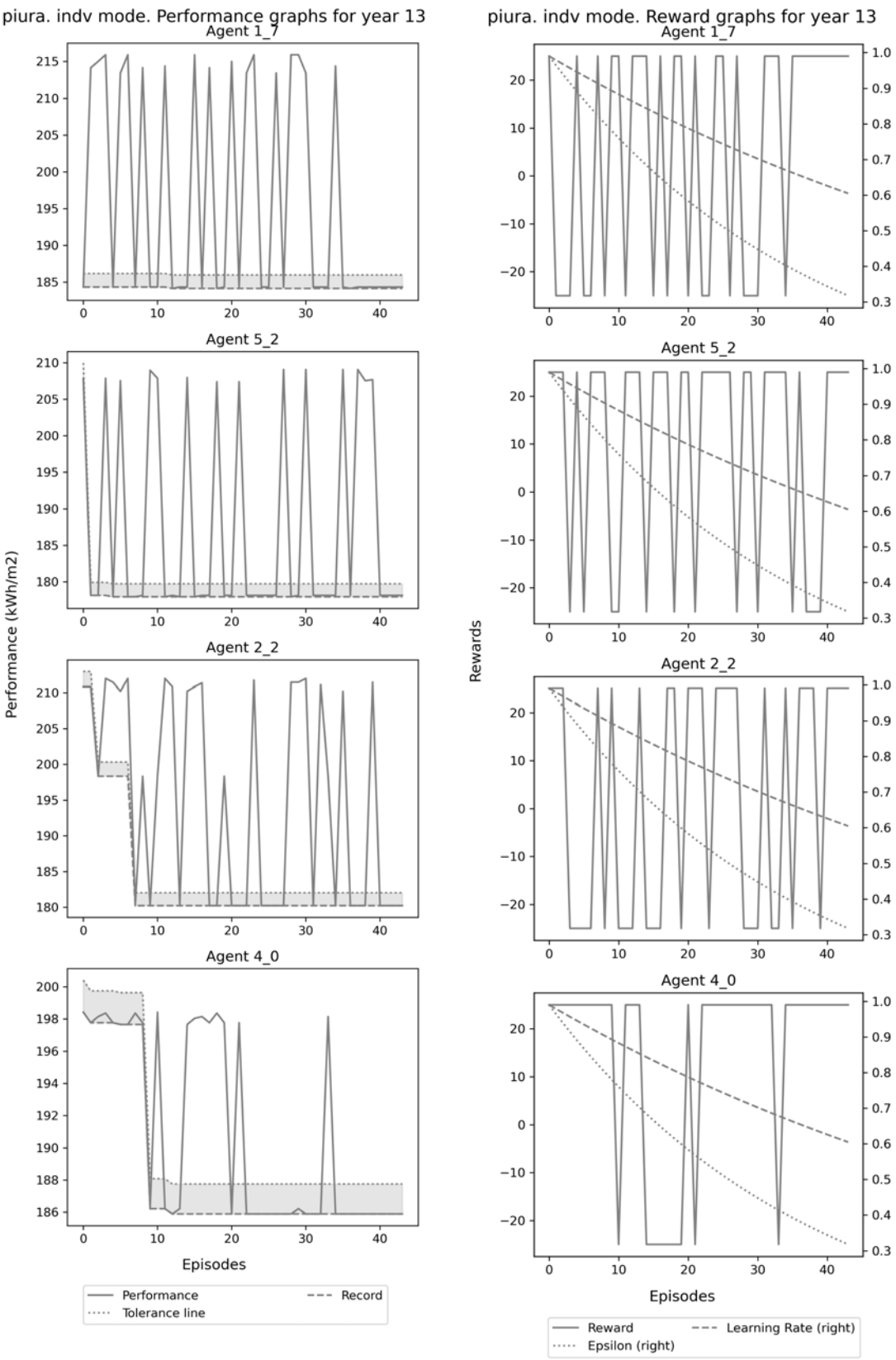
() Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (23)

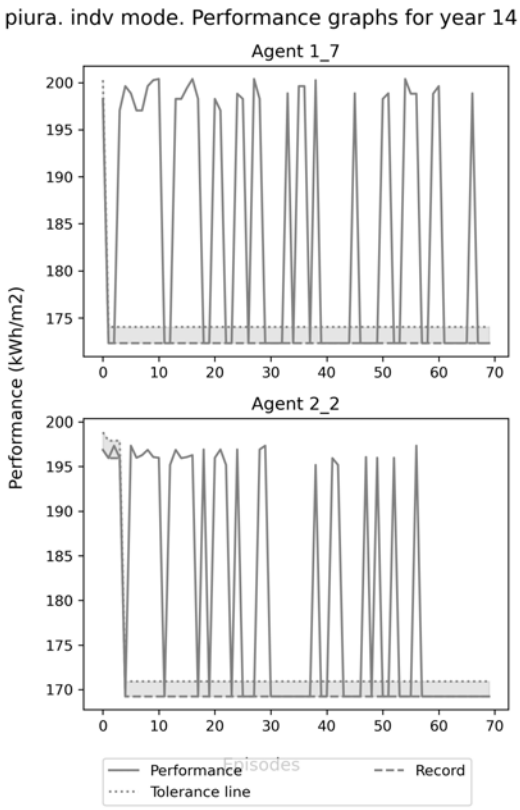
() Performance

() Rewards

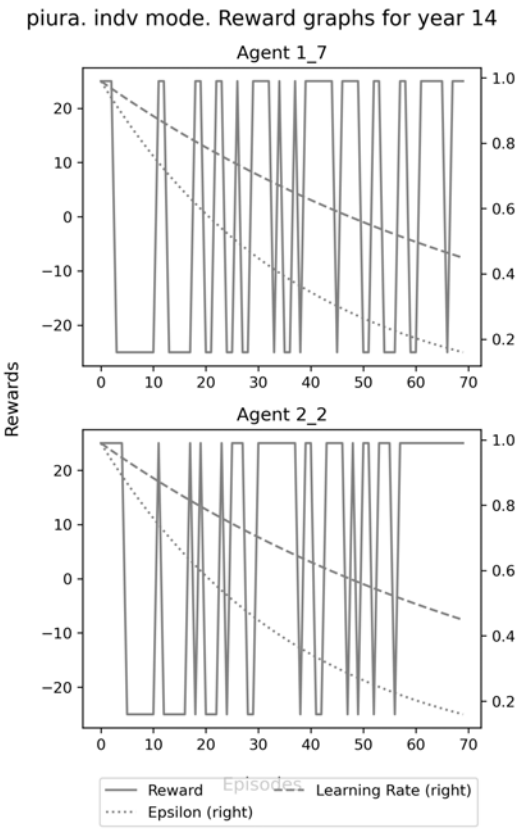


Learning process of all agents in Piura Competitive scenario for energy minimisation (23)

() Performance

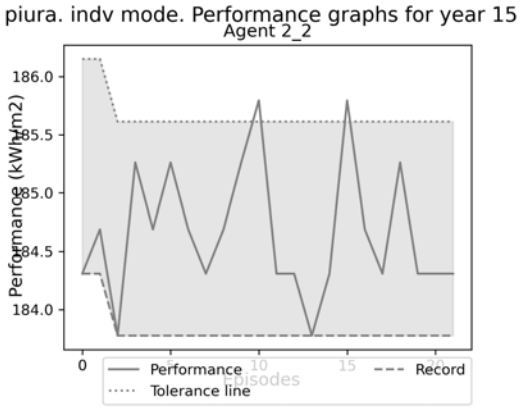


() Rewards

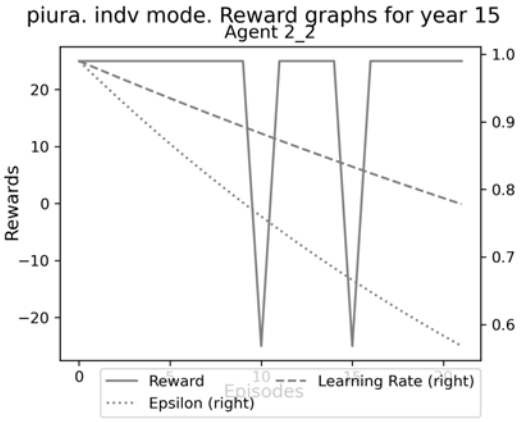


Learning process of all agents in Piura Competitive scenario for energy minimisation (24)

() Performance

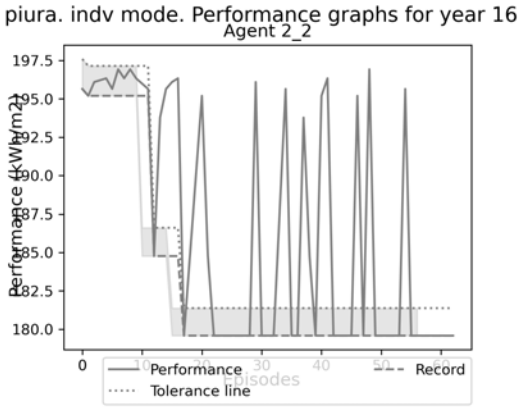


() Rewards

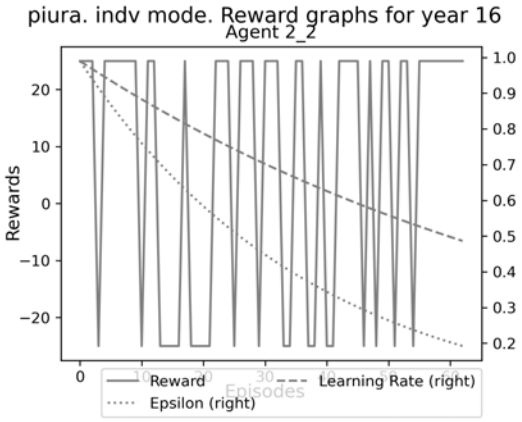


Learning process of all agents in Piura Competitive scenario for energy minimisation (25)

() Performance

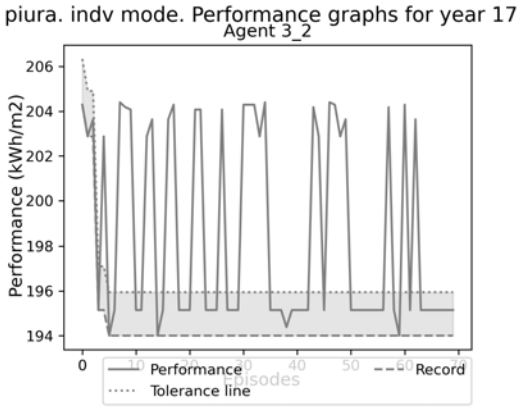


() Rewards

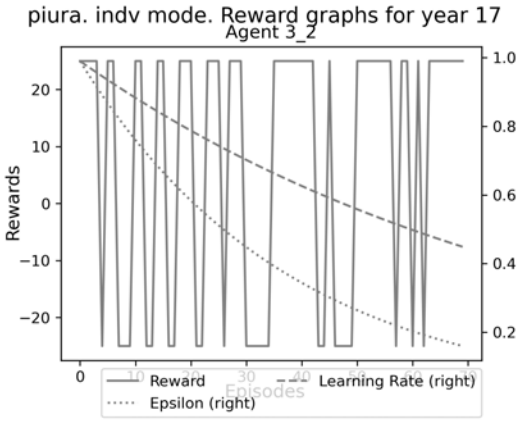


Learning process of all agents in Piura Competitive scenario for energy minimisation (26)

() Performance

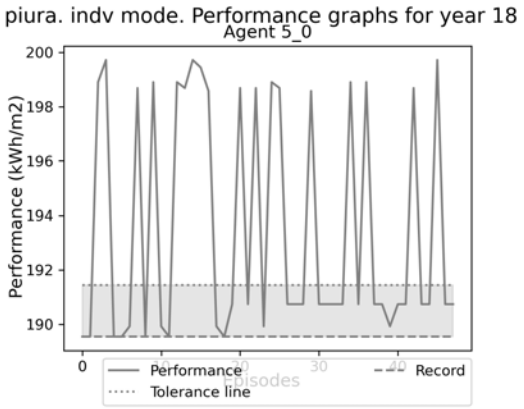


() Rewards

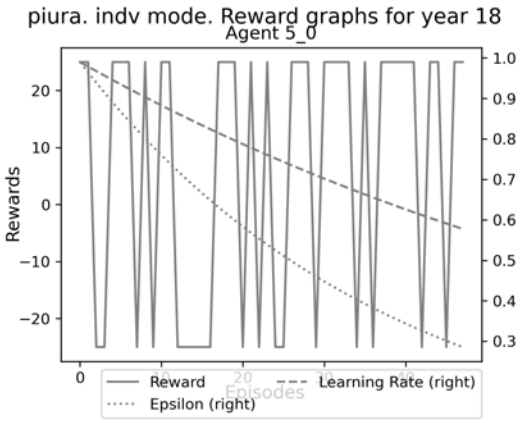


Learning process of all agents in Piura Competitive scenario for energy minimisation (27)

() Performance

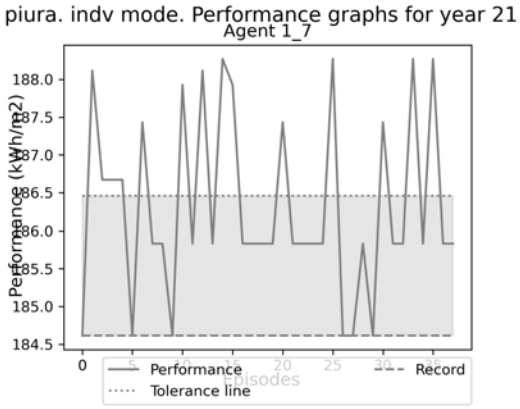


() Rewards

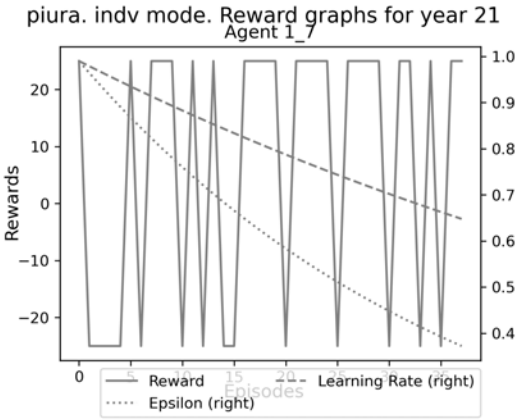


Learning process of all agents in Piura Competitive scenario for energy minimisation (28)

() Performance

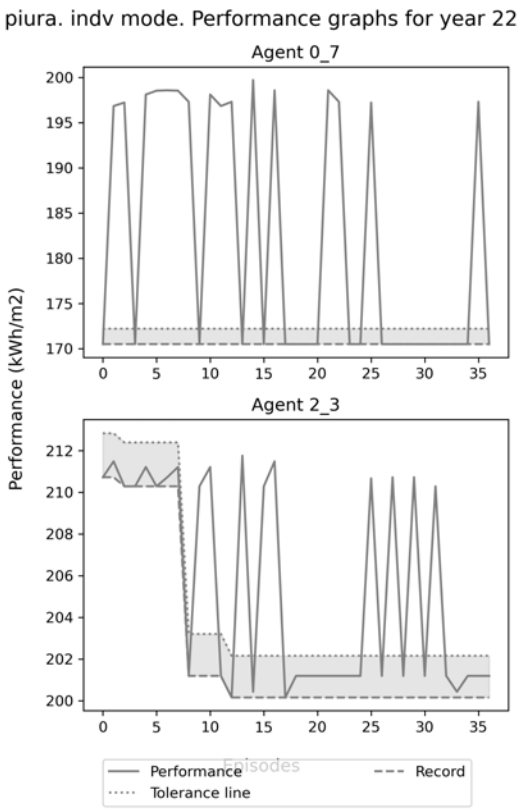


() Rewards

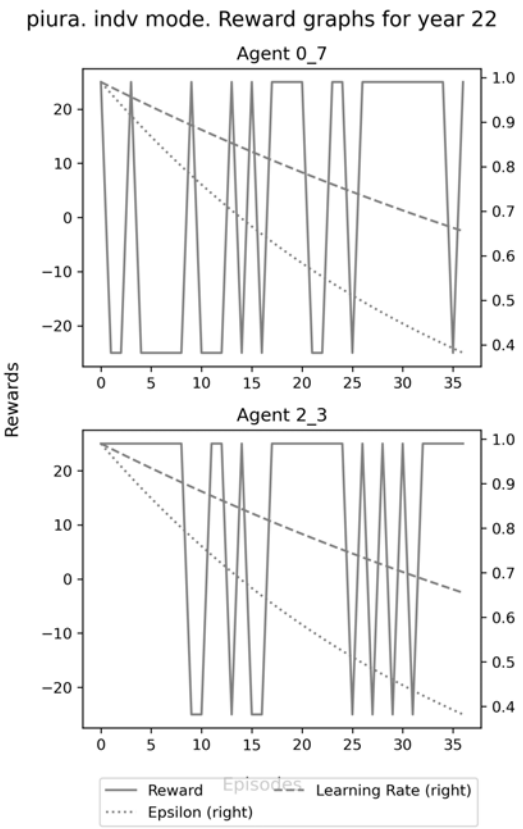


Learning process of all agents in Piura Competitive scenario for energy minimisation (29)

() Performance

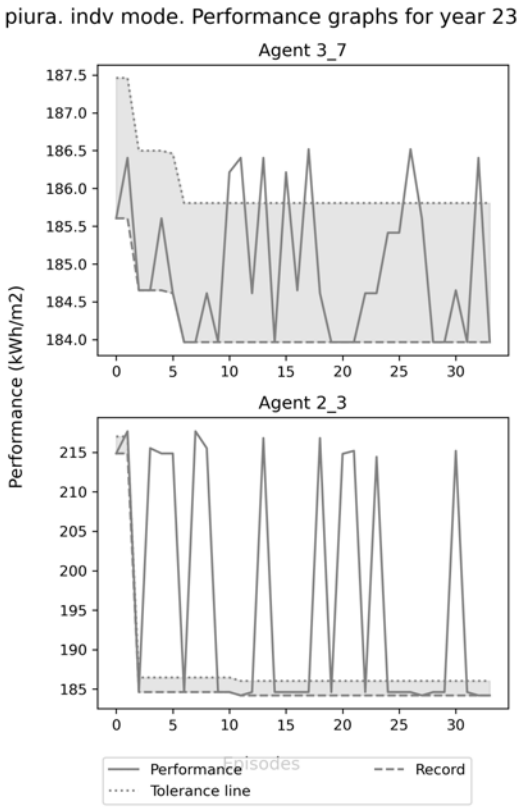


() Rewards

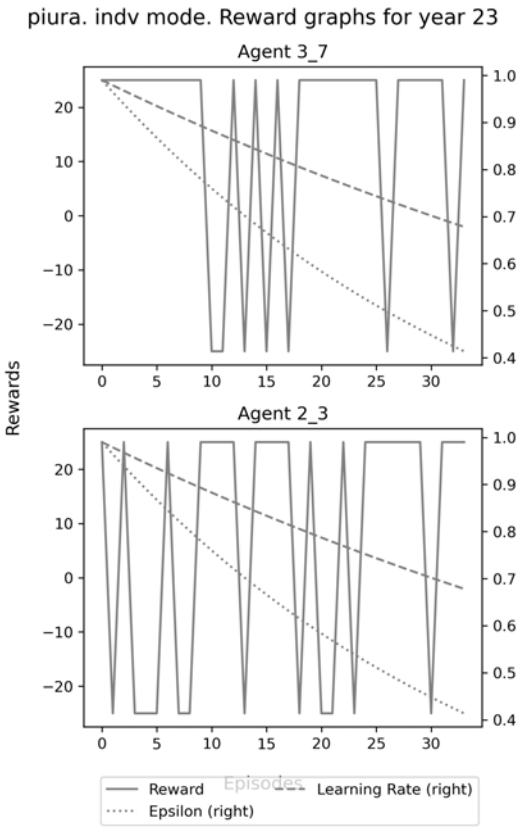


Learning process of all agents in Piura Competitive scenario for energy minimisation (30)

() Performance



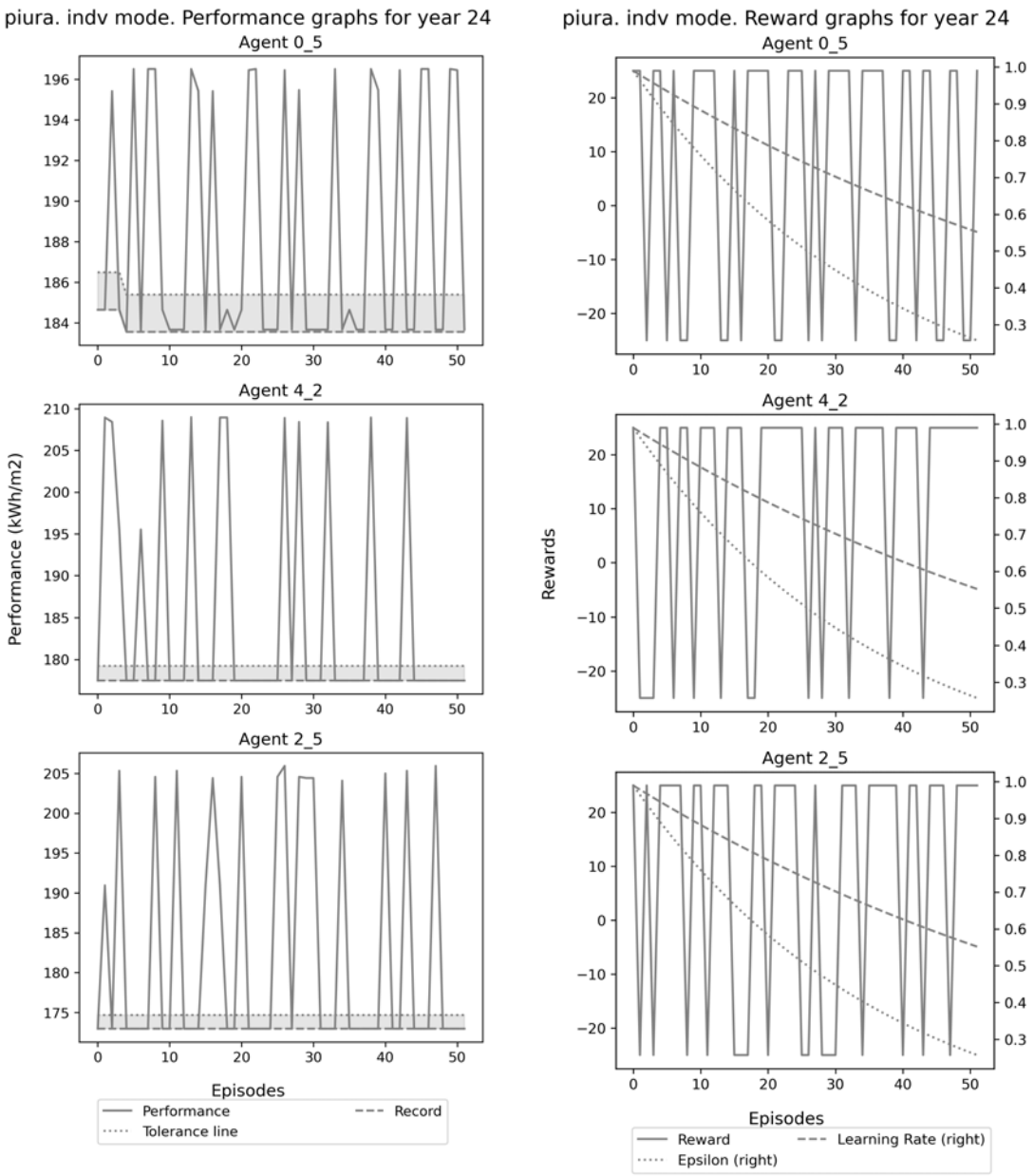
() Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (31)

() Performance

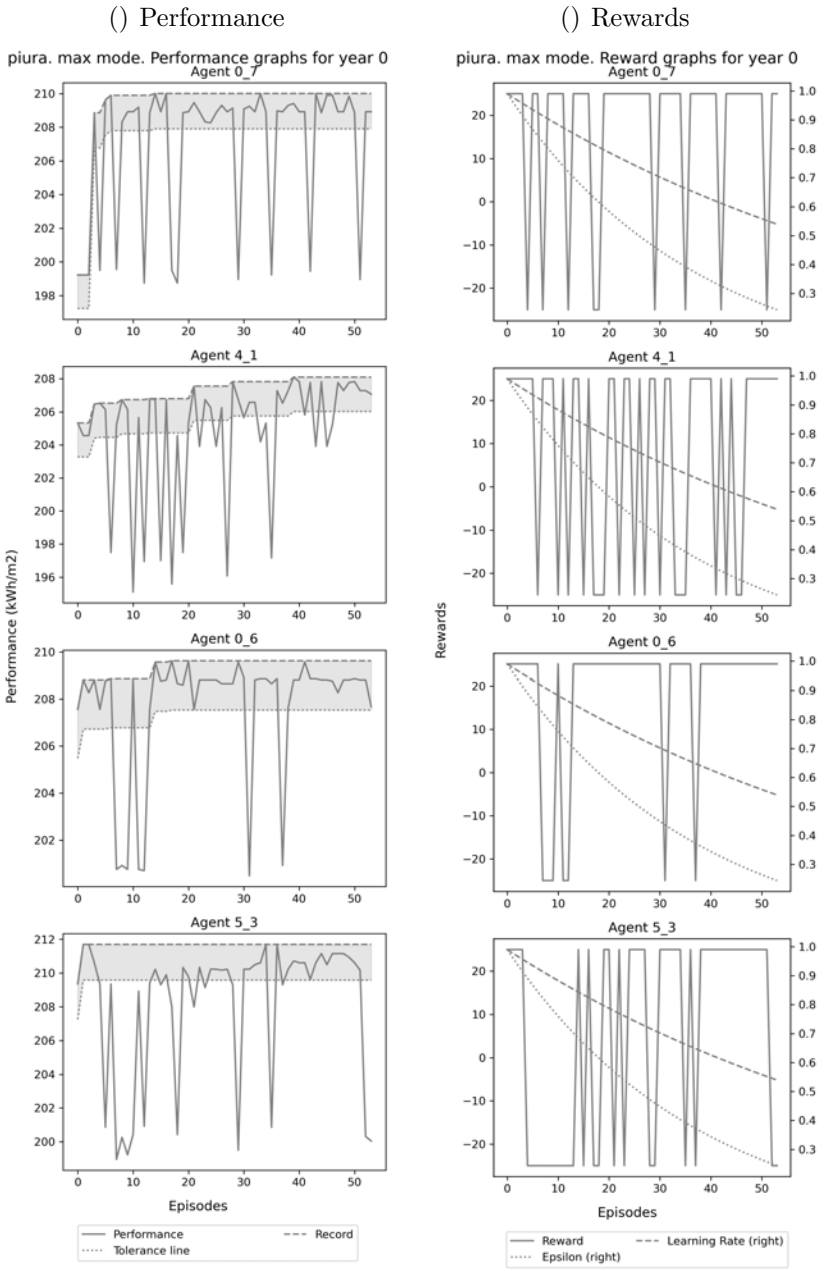
() Rewards



Learning process of all agents in Piura Competitive scenario for energy minimisation (32)

Appendix 17

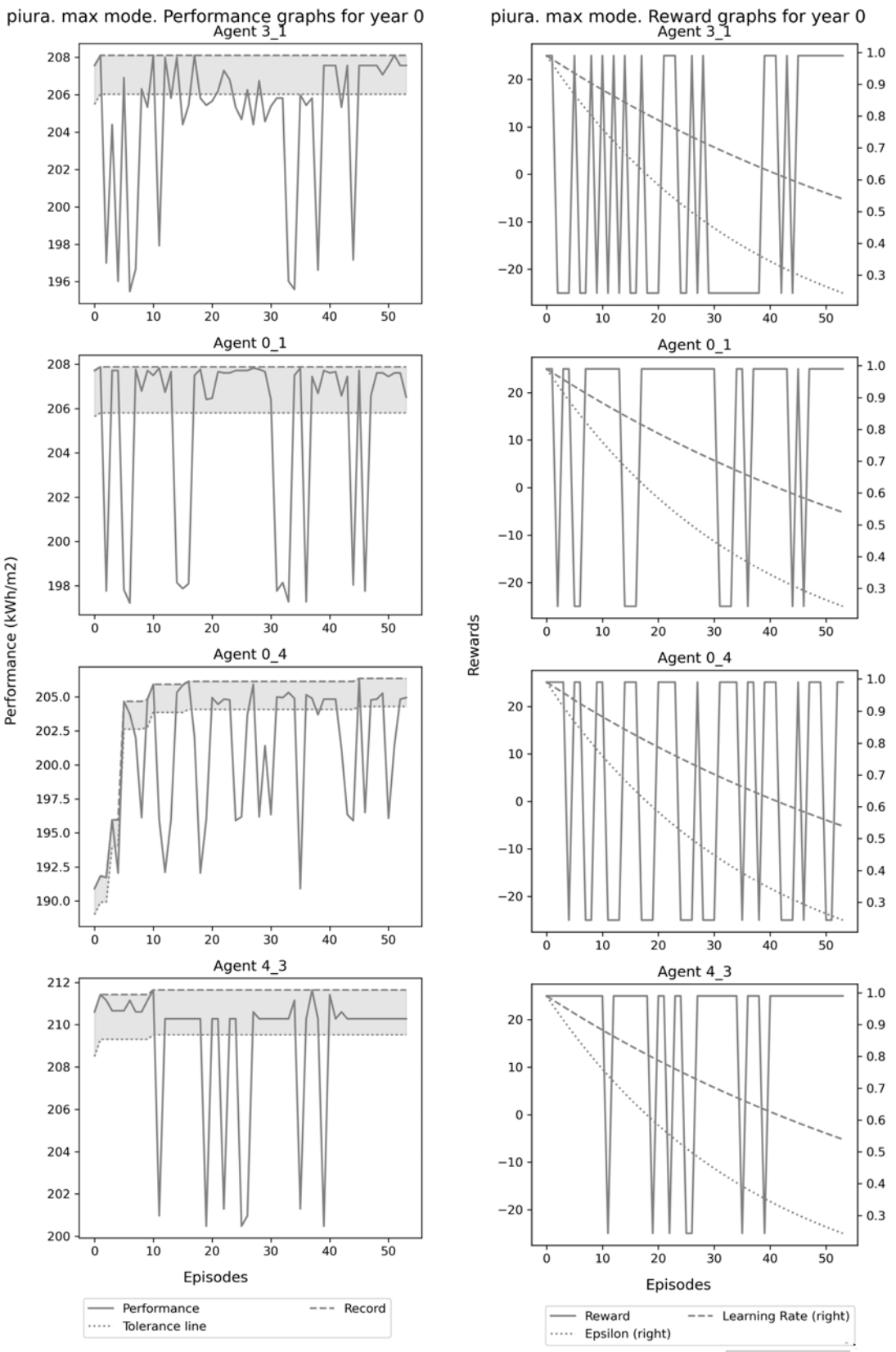
Learning process by agent for each simulation year (Piura, Maximisation, competitive)



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (1)

() Performance

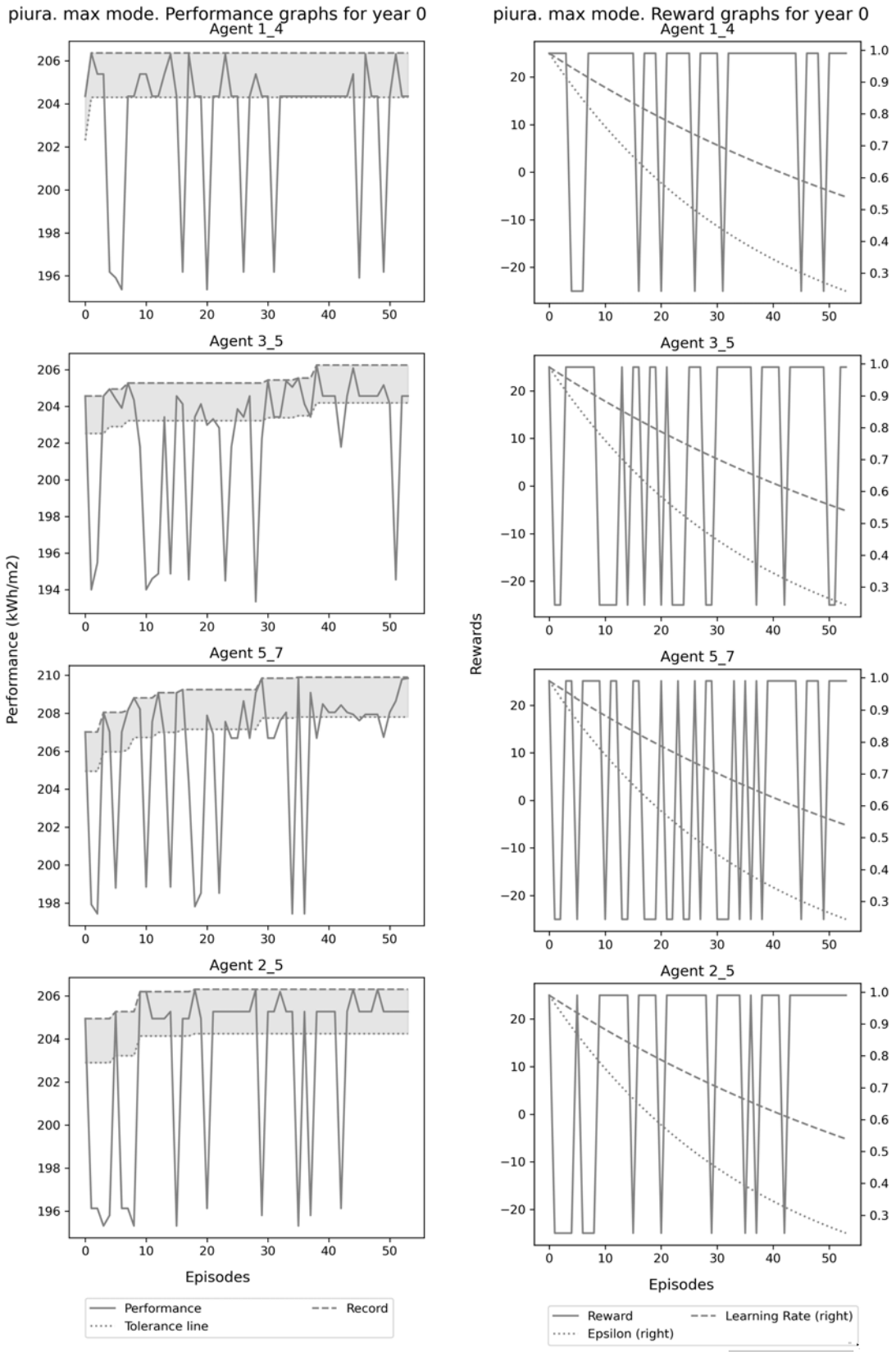
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (2)

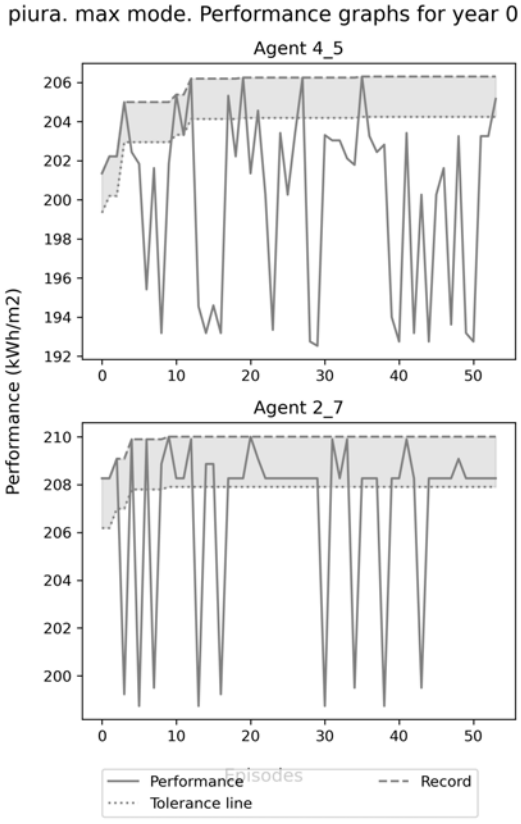
() Performance

() Rewards

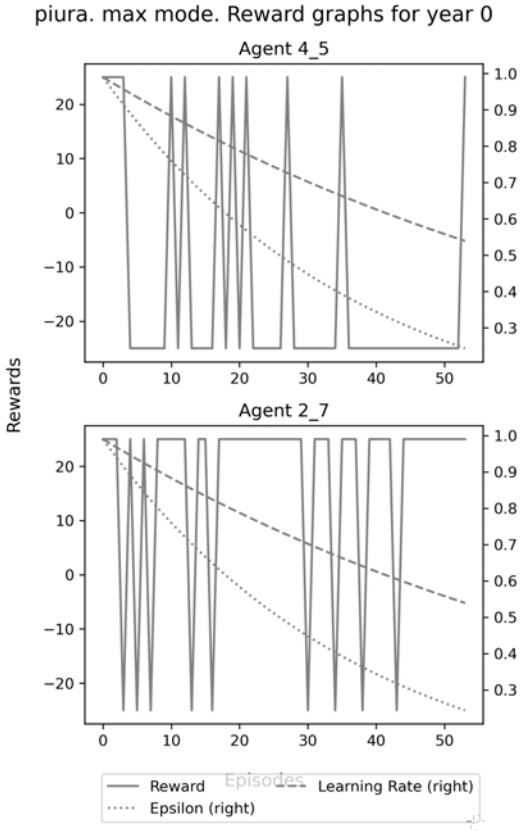


Learning process of all agents in Piura Competitive scenario for energy use Maximisation (3)

() Performance



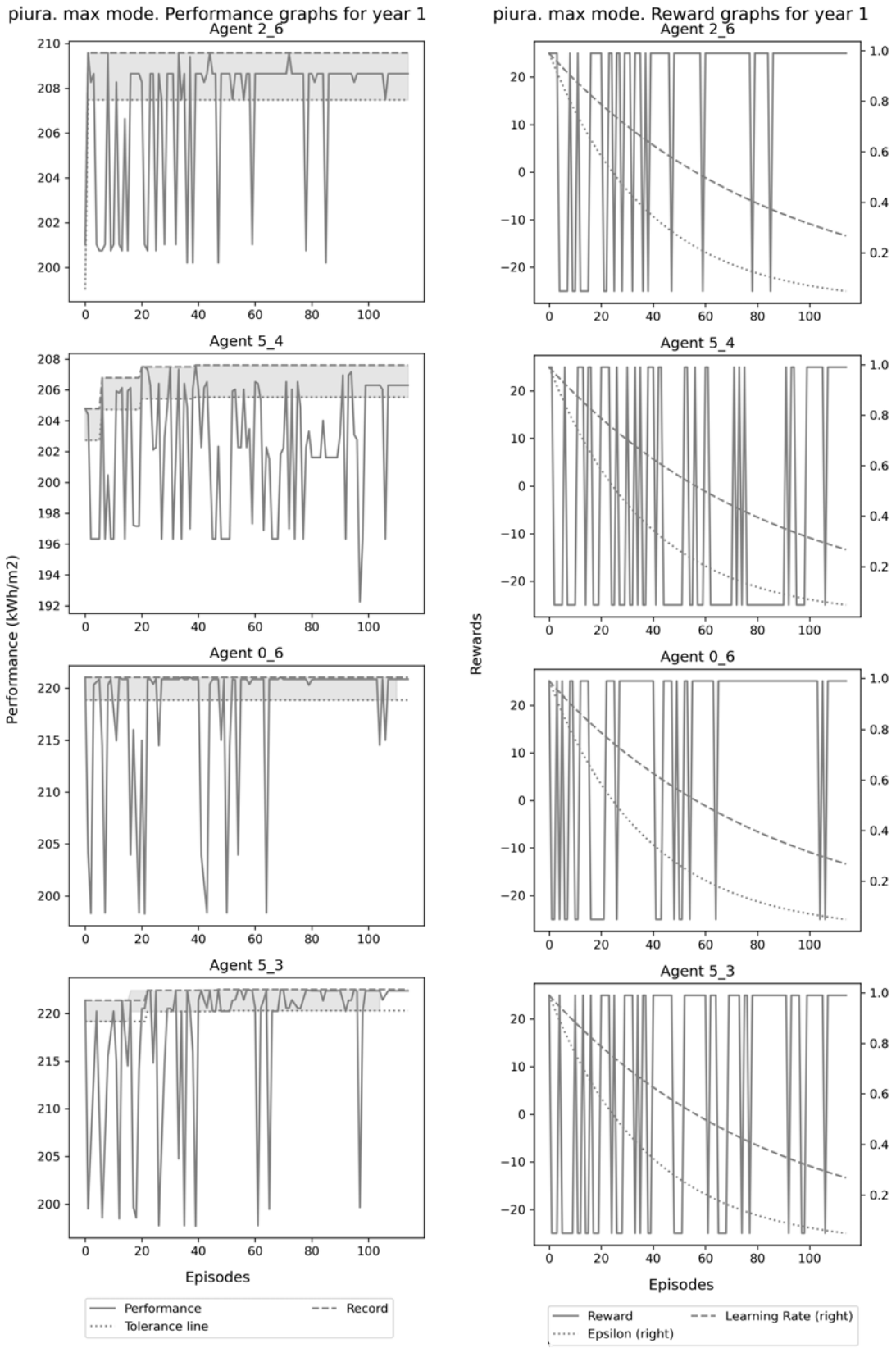
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (4)

() Performance

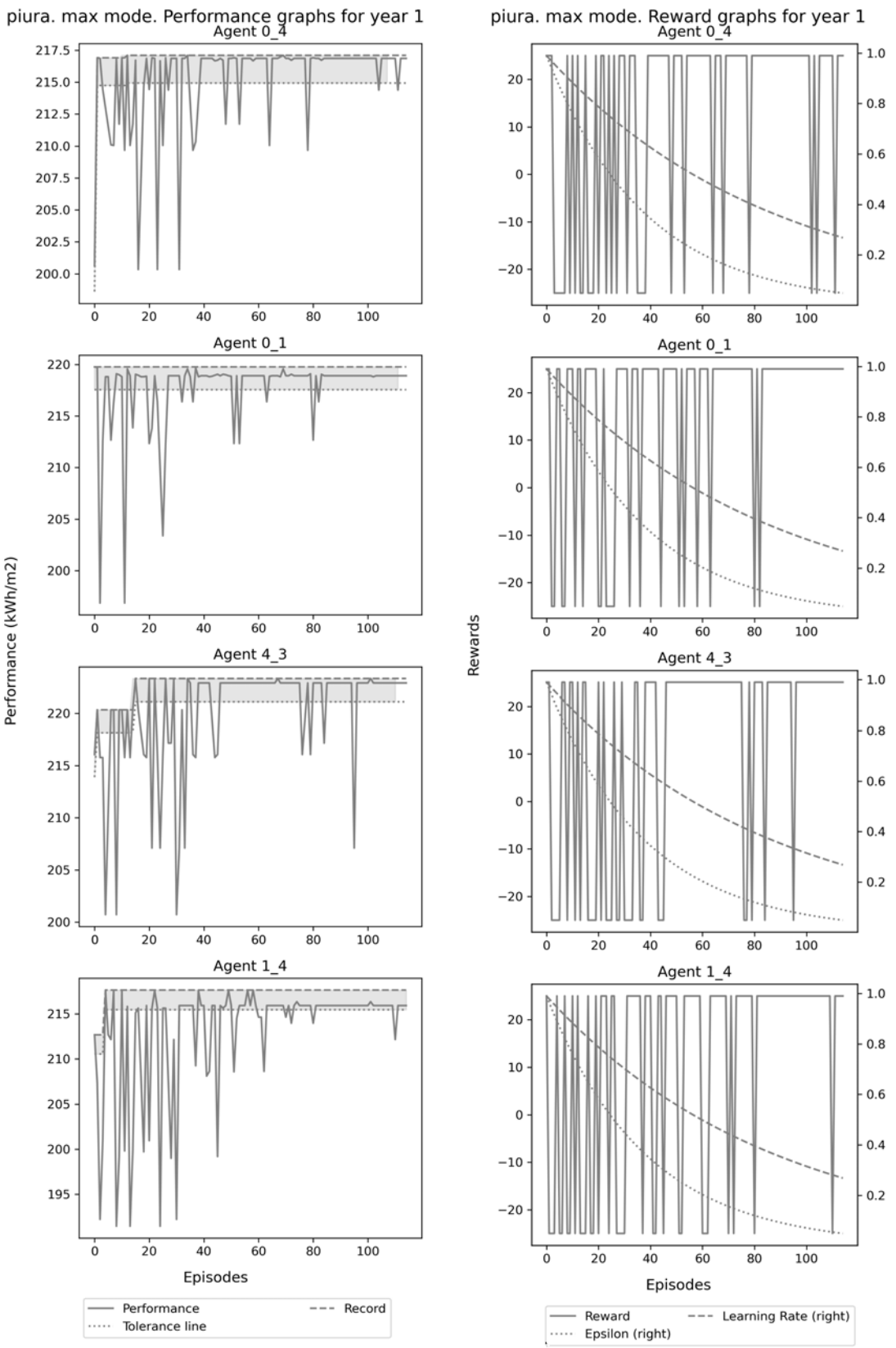
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (5)

() Performance

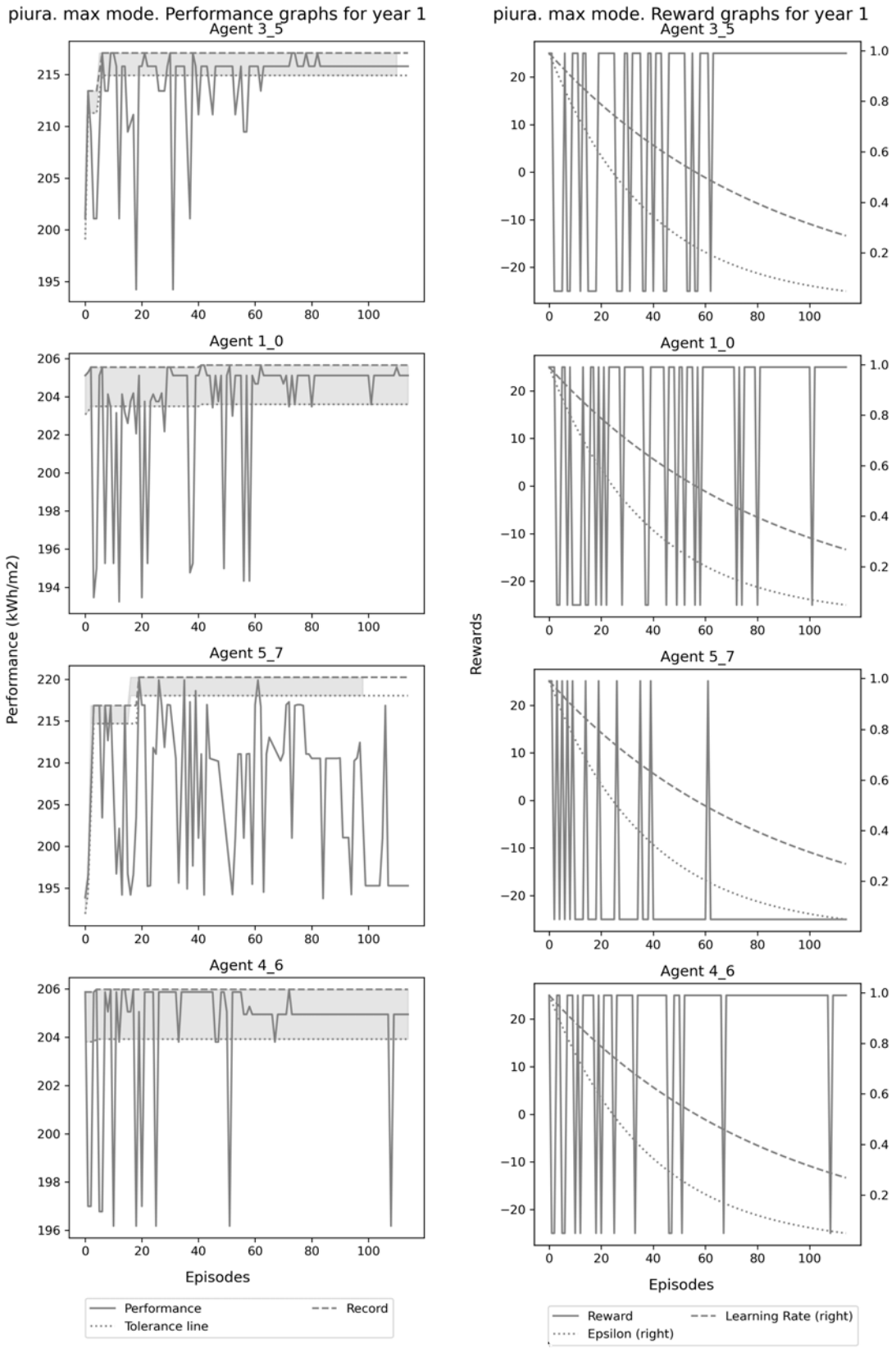
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (6)

() Performance

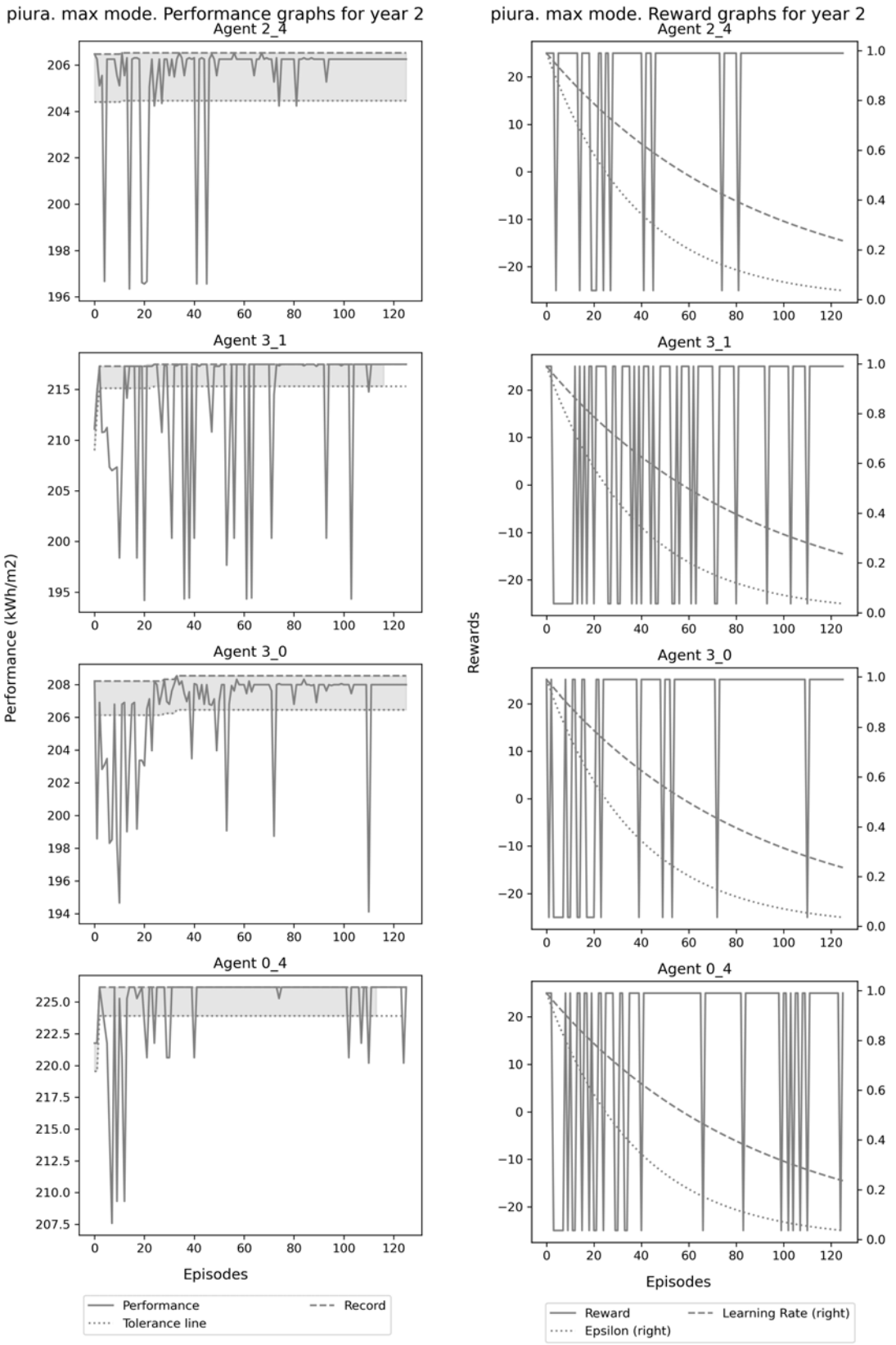
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (7)

() Performance

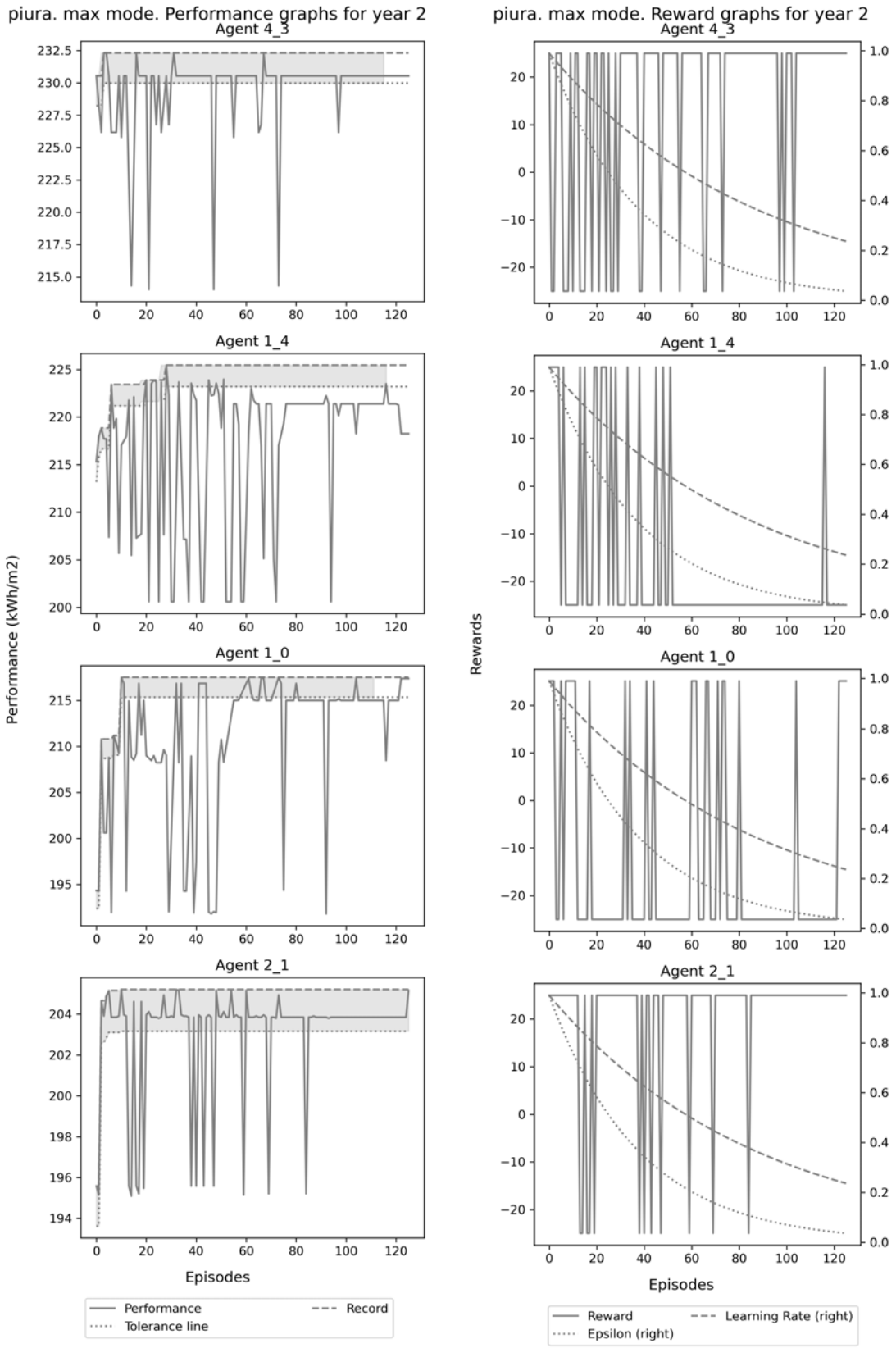
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (8)

() Performance

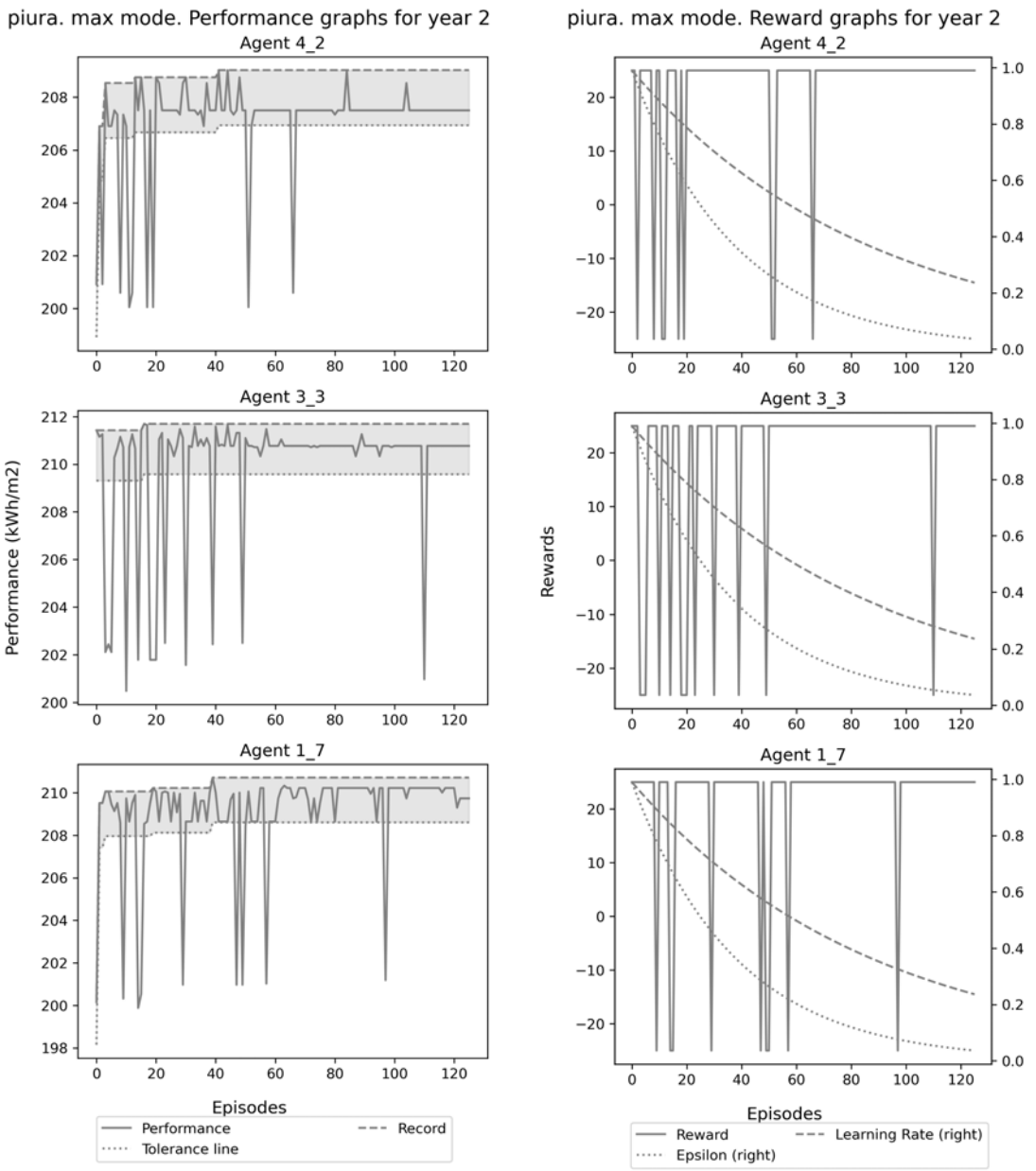
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (9)

() Performance

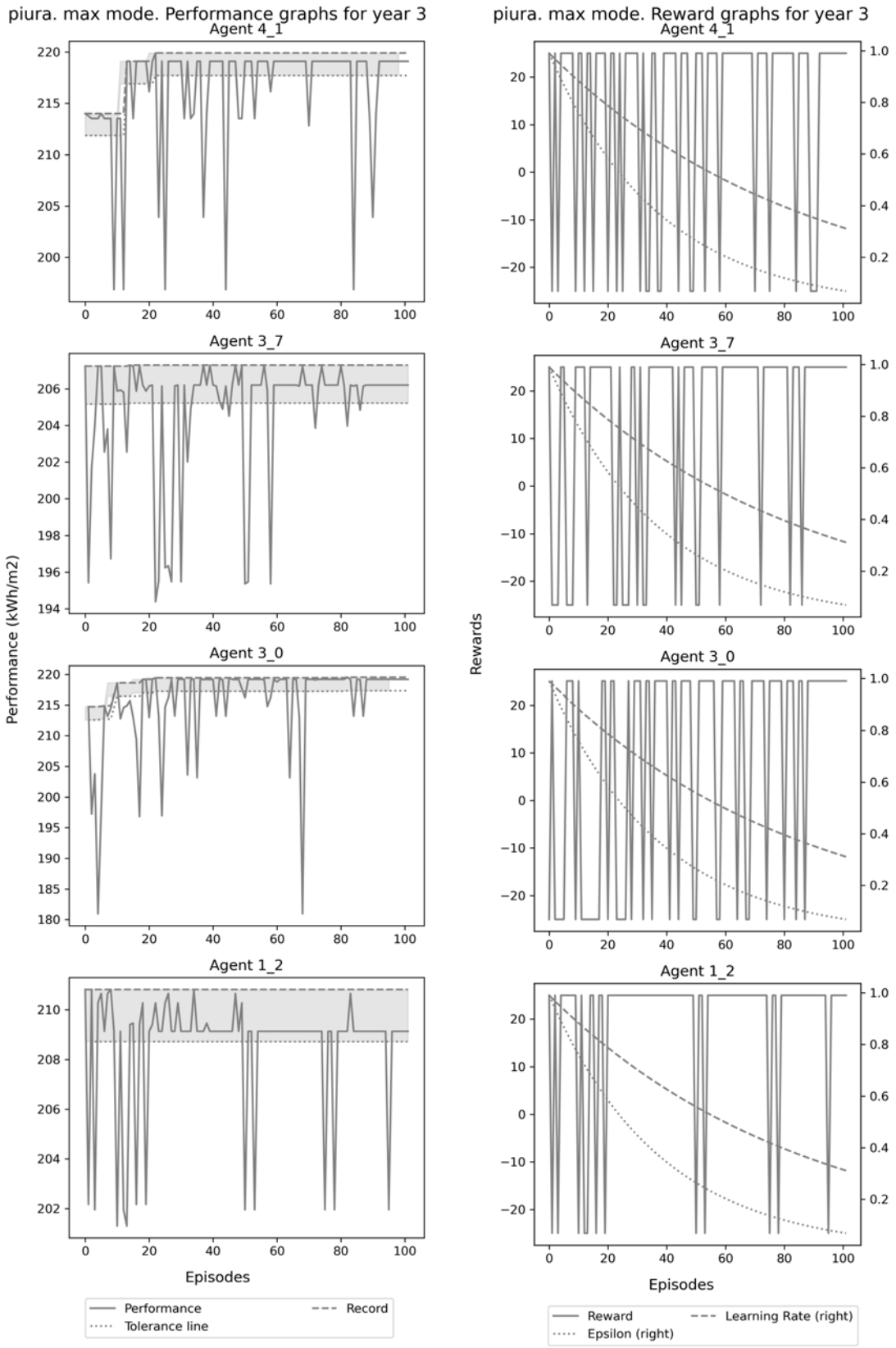
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (10)

() Performance

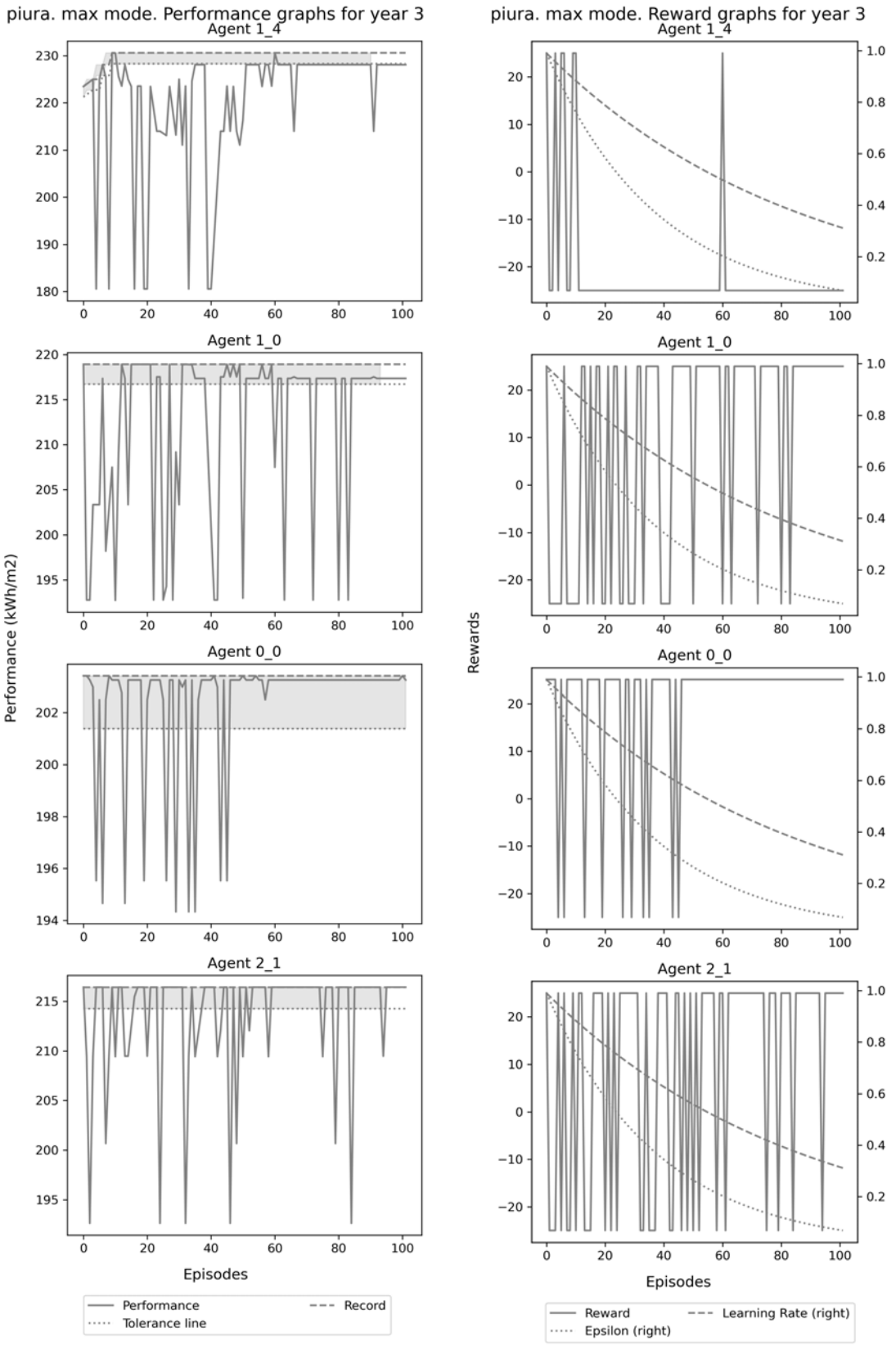
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (11)

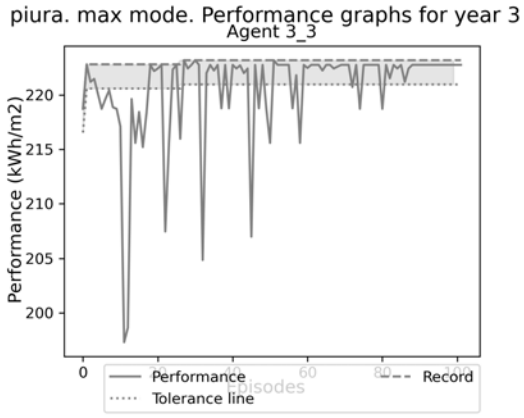
() Performance

() Rewards

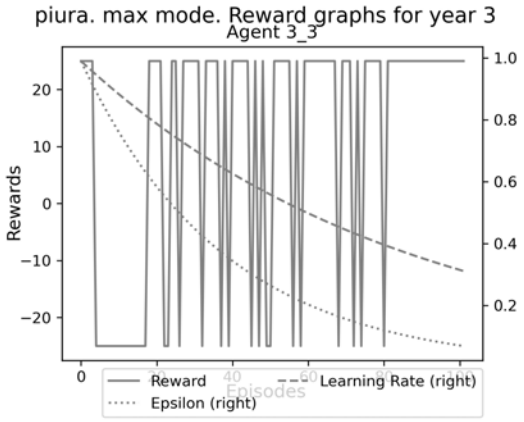


Learning process of all agents in Piura Competitive scenario for energy use Maximisation (12)

() Performance



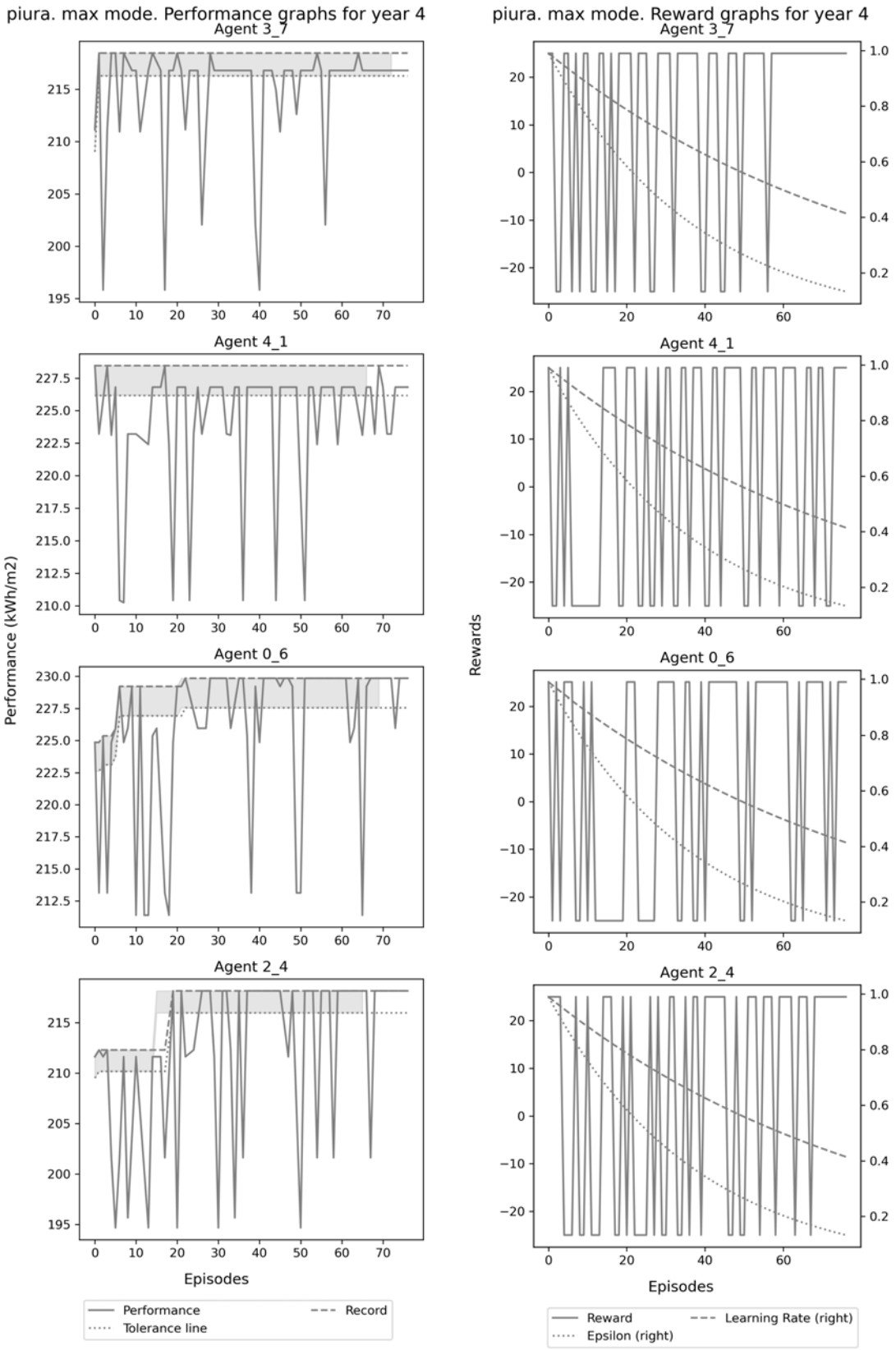
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (13)

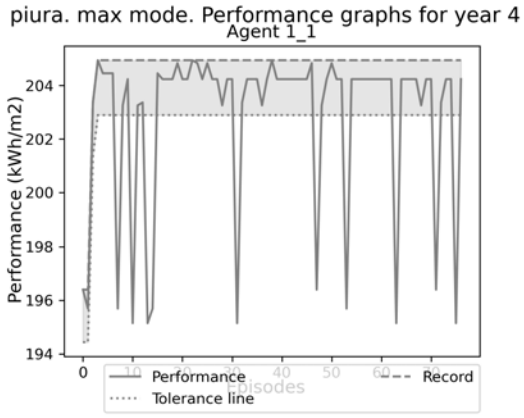
() Performance

() Rewards

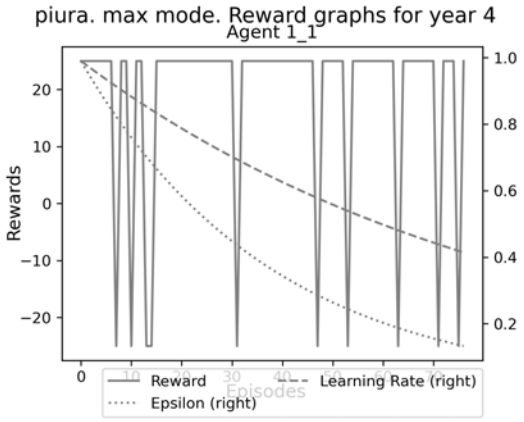


Learning process of all agents in Piura Competitive scenario for energy use Maximisation (14)

() Performance



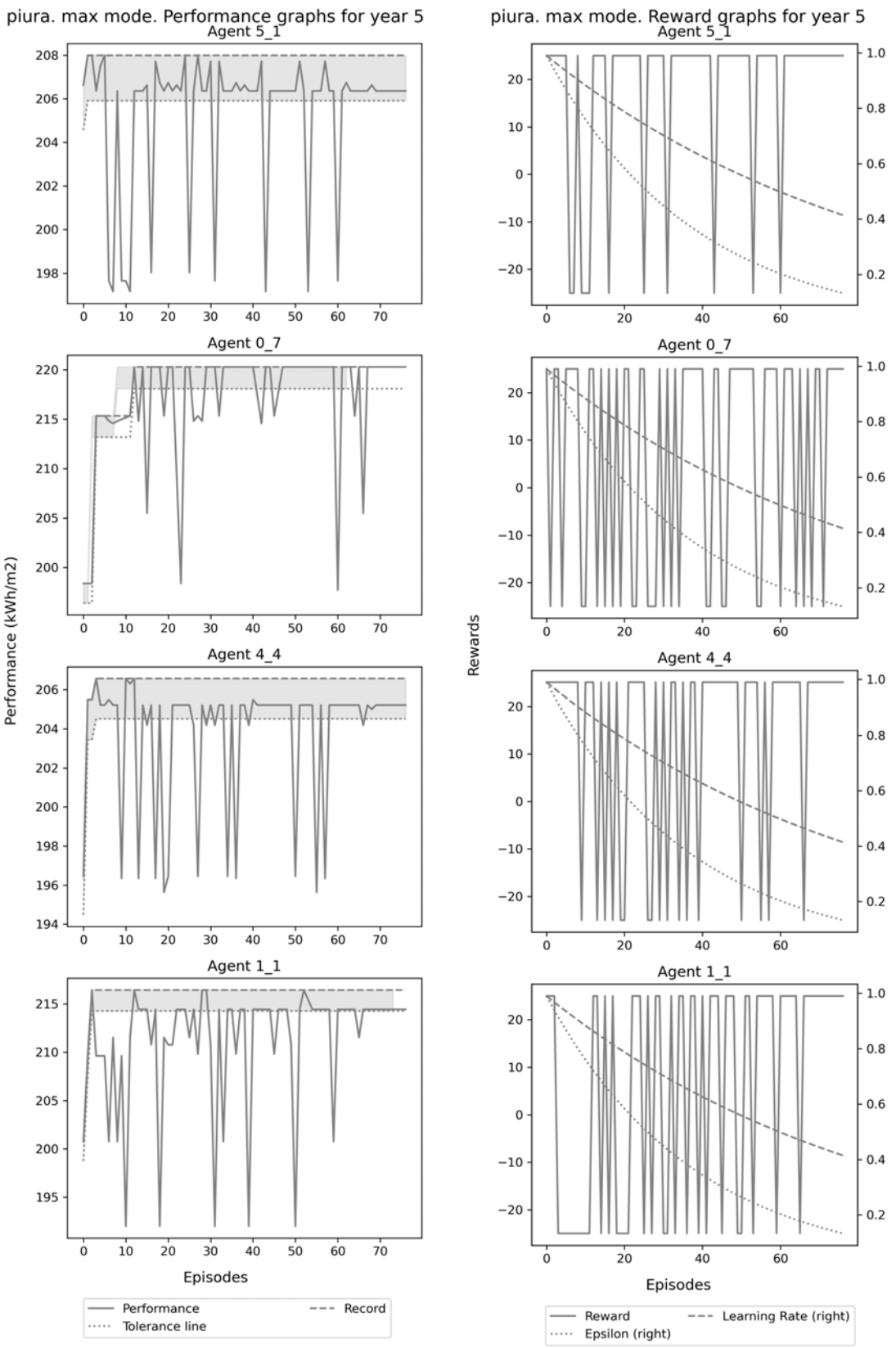
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (15)

() Performance

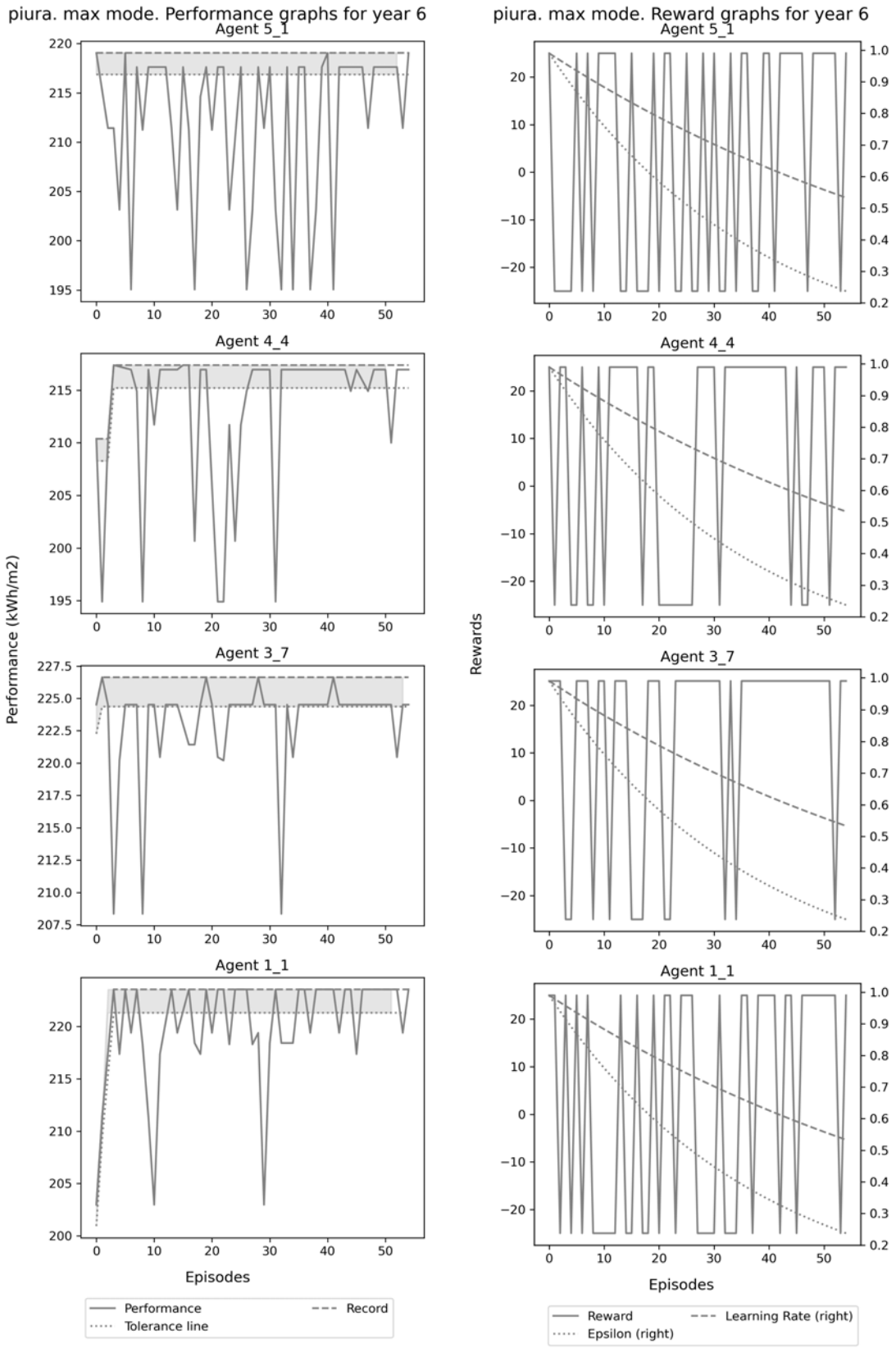
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (16)

() Performance

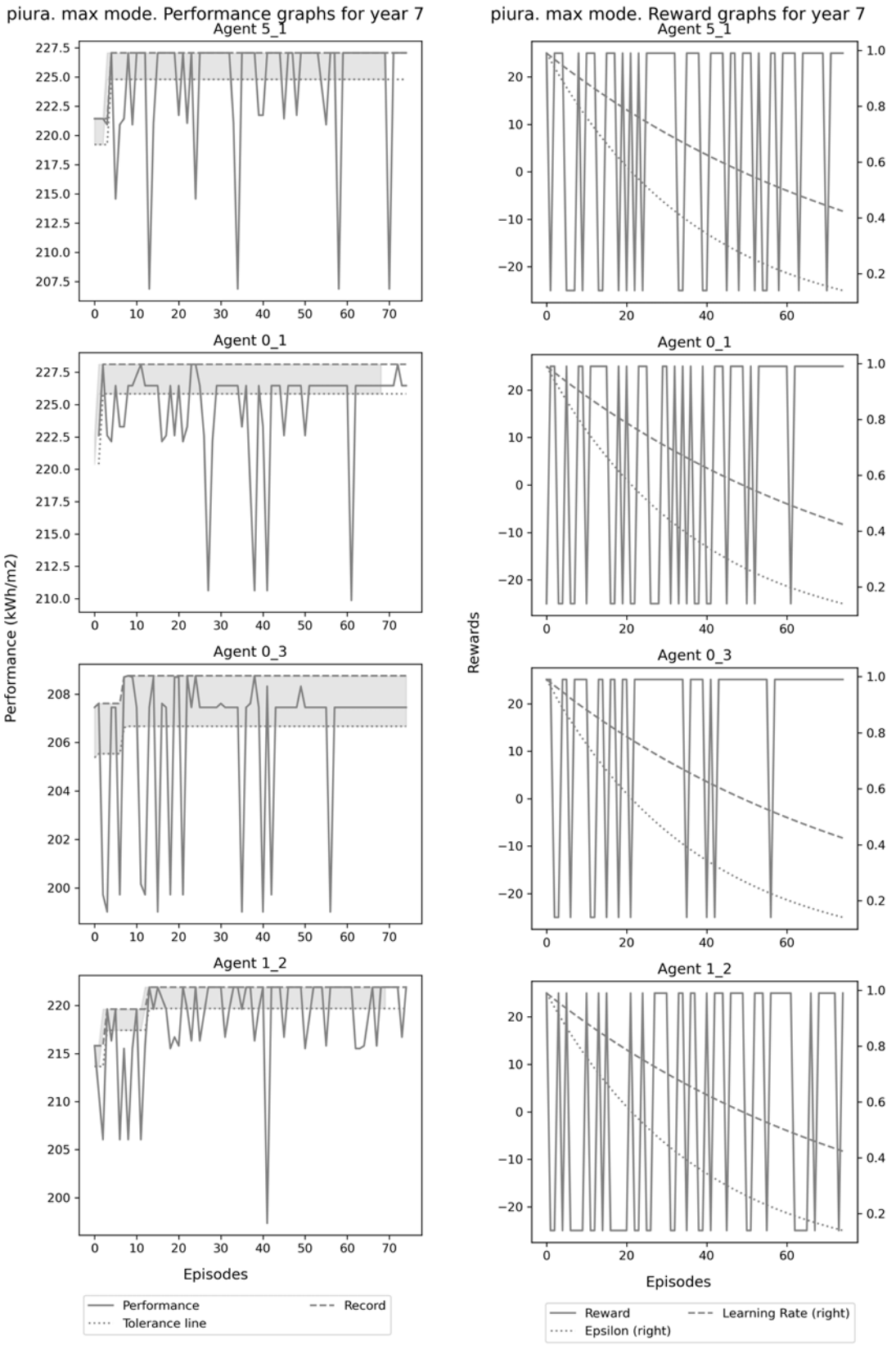
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (17)

() Performance

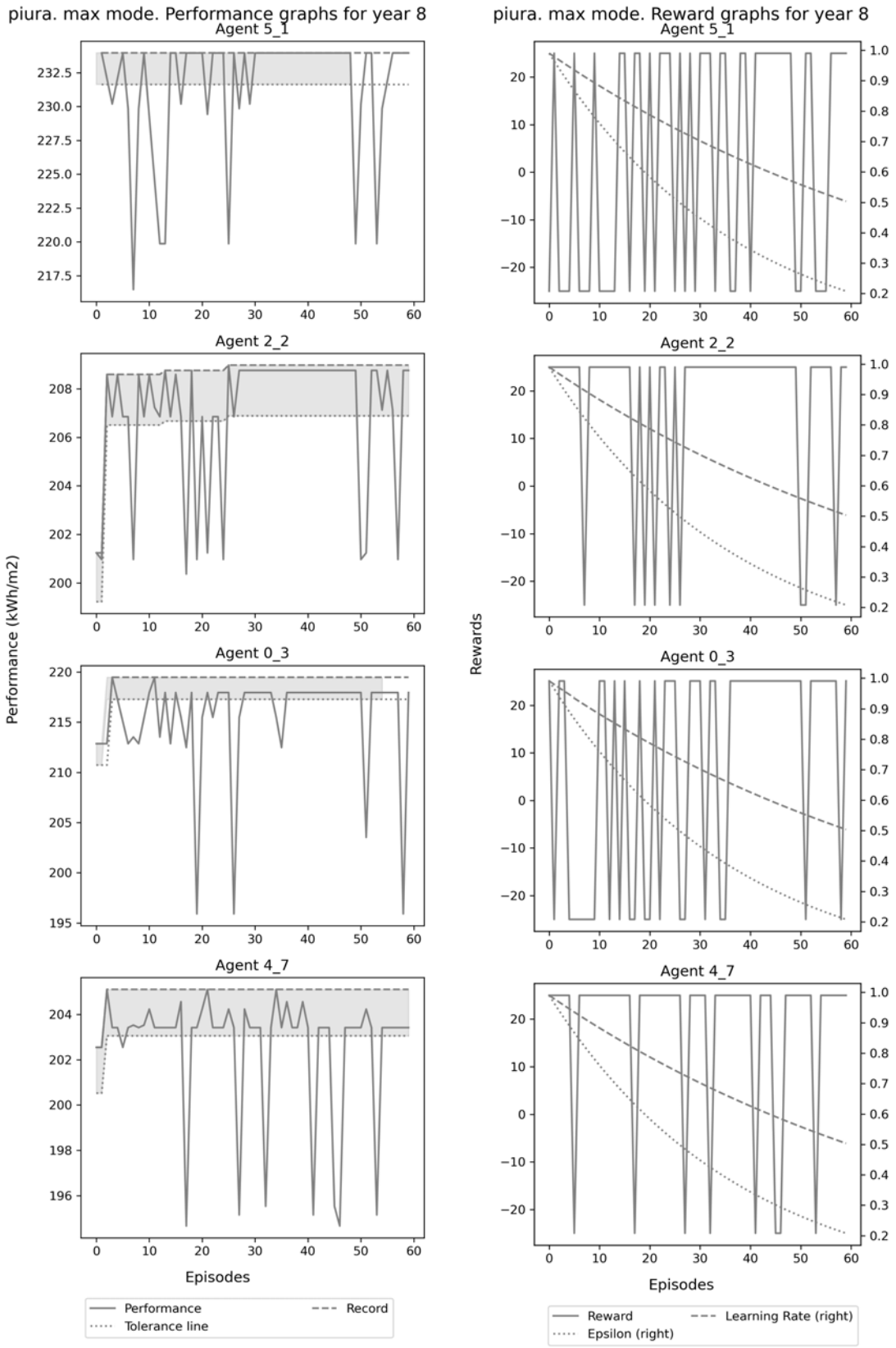
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (18)

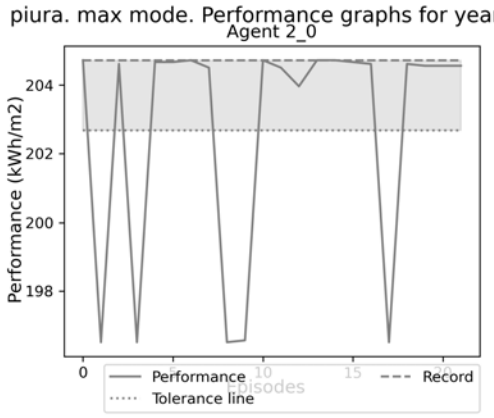
() Performance

() Rewards

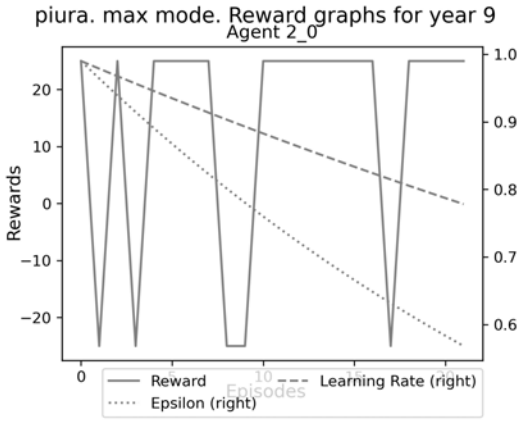


Learning process of all agents in Piura Competitive scenario for energy use Maximisation (19)

() Performance



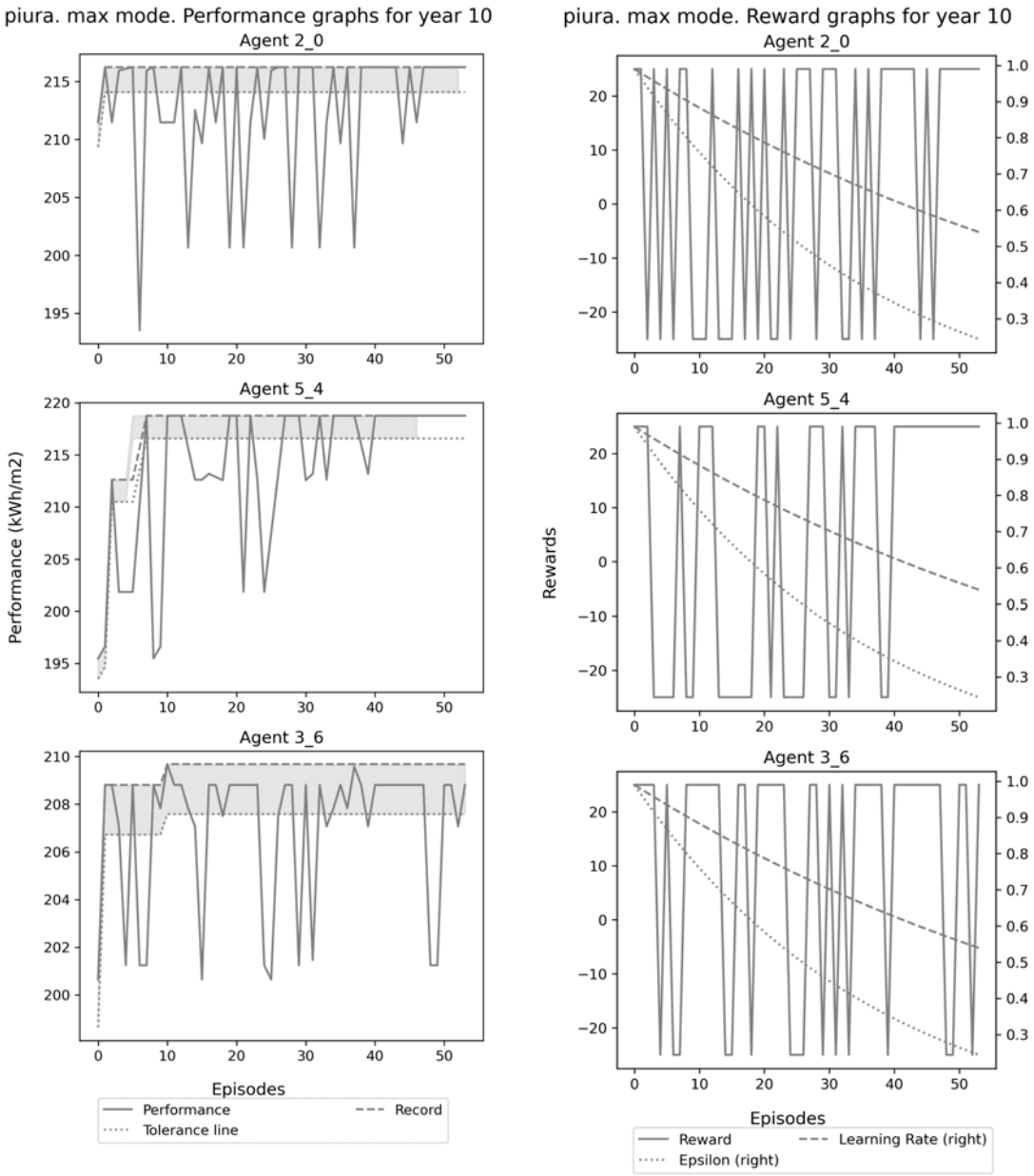
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (20)

() Performance

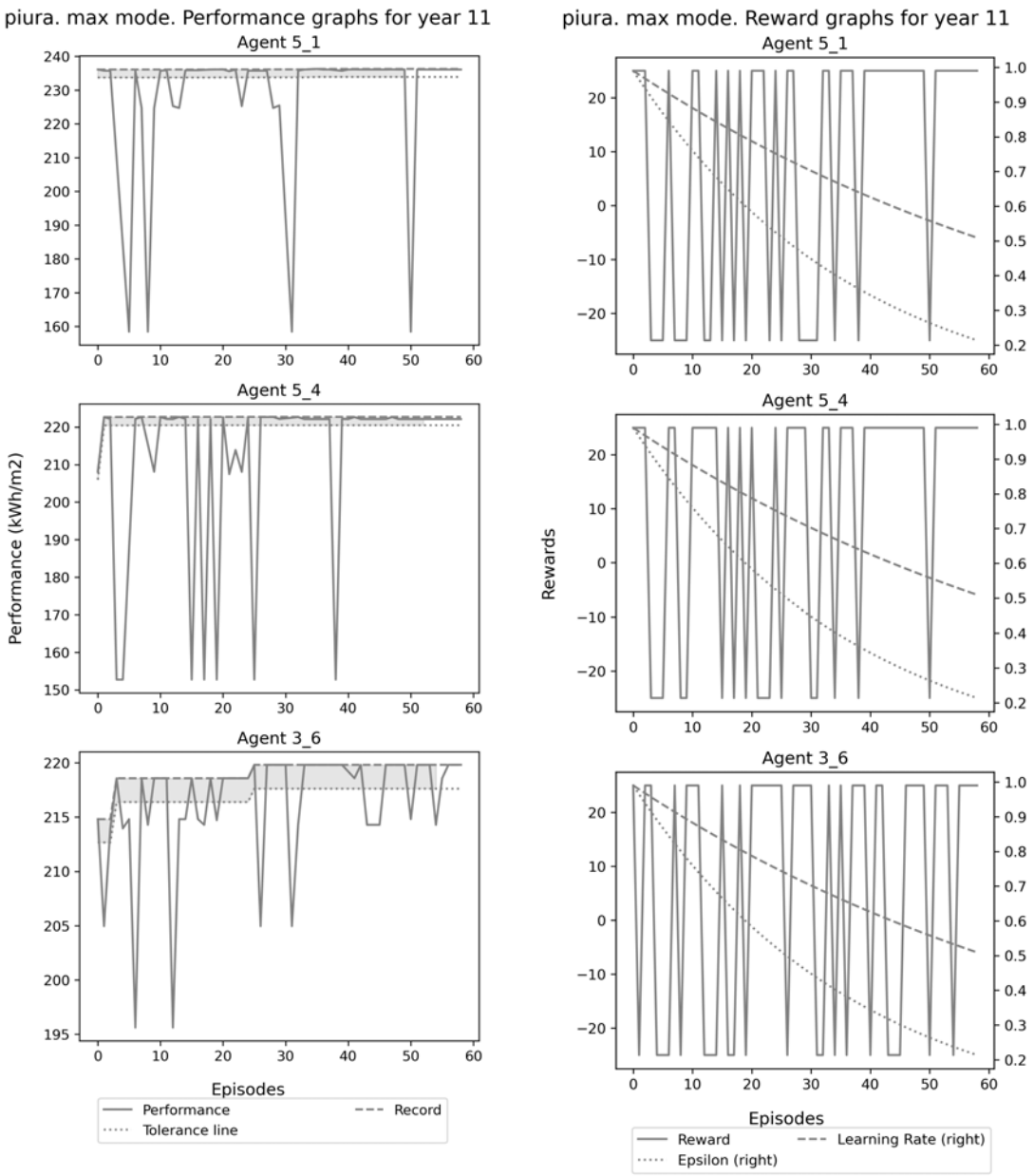
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (21)

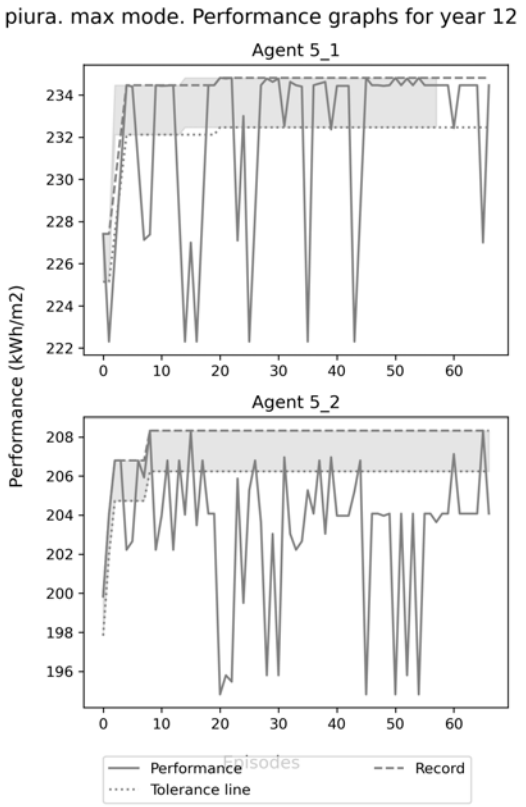
() Performance

() Rewards

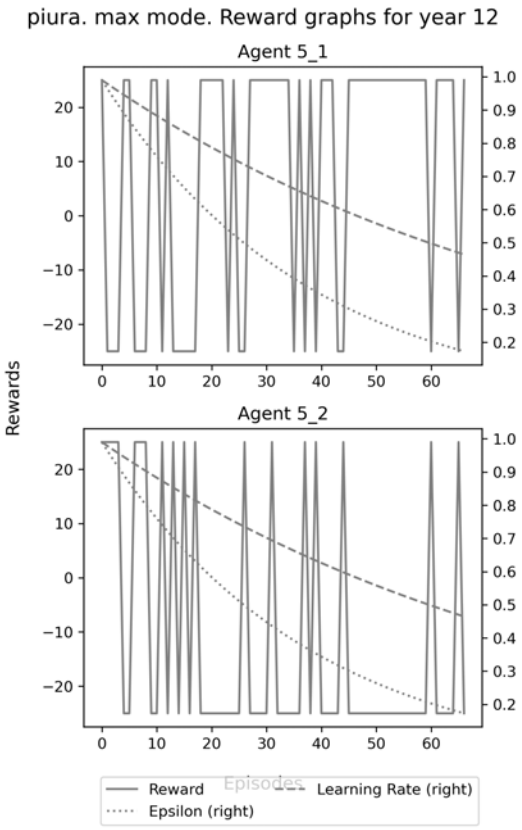


Learning process of all agents in Piura Competitive scenario for energy use Maximisation (22)

() Performance



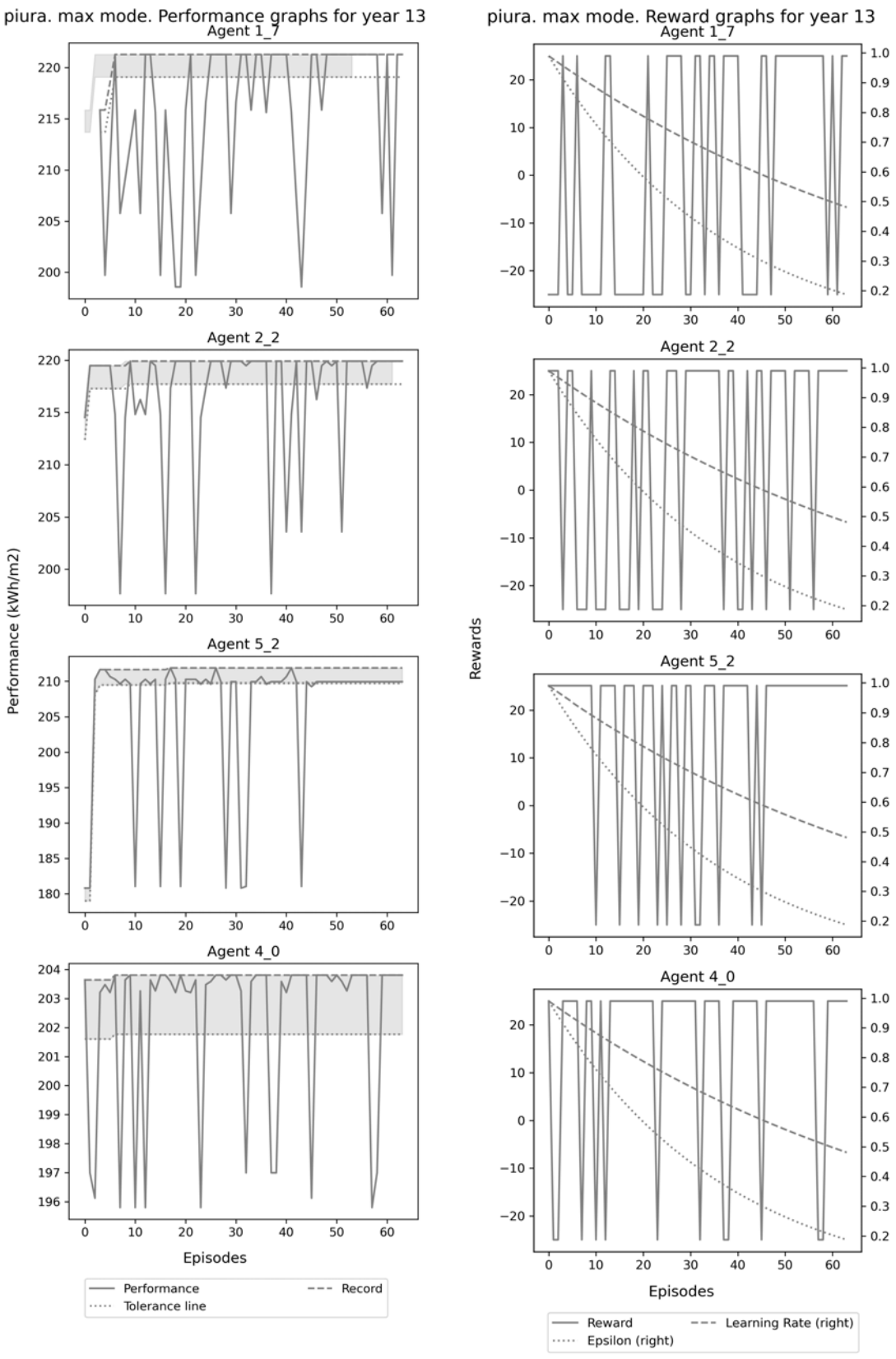
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (23)

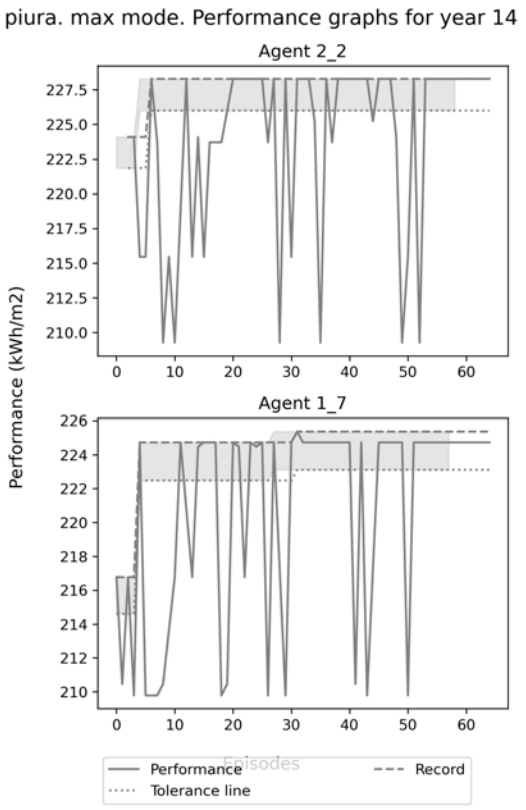
() Performance

() Rewards

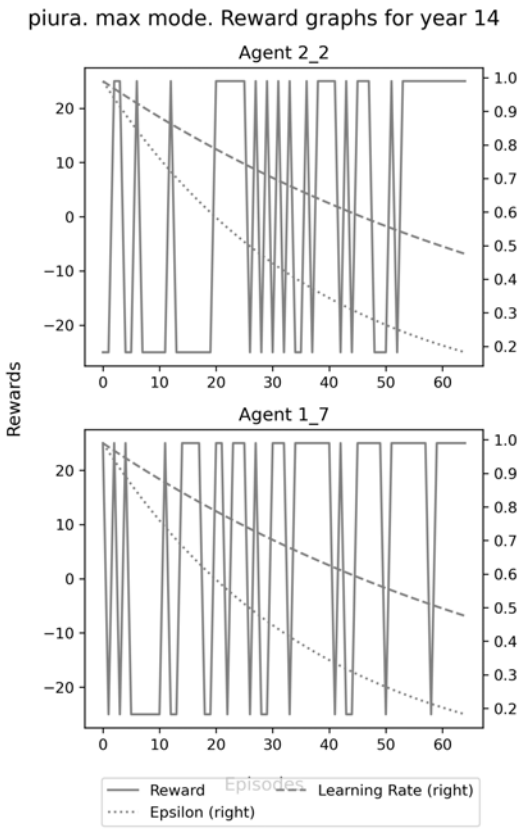


Learning process of all agents in Piura Competitive scenario for energy use Maximisation (24)

() Performance

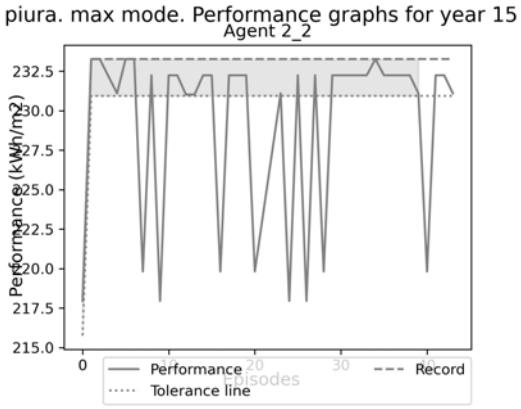


() Rewards

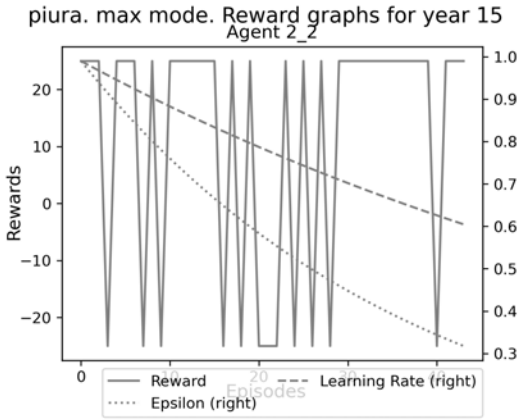


Learning process of all agents in Piura Competitive scenario for energy use Maximisation (25)

() Performance

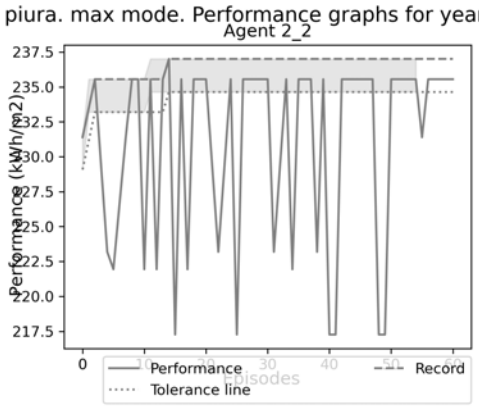


() Rewards

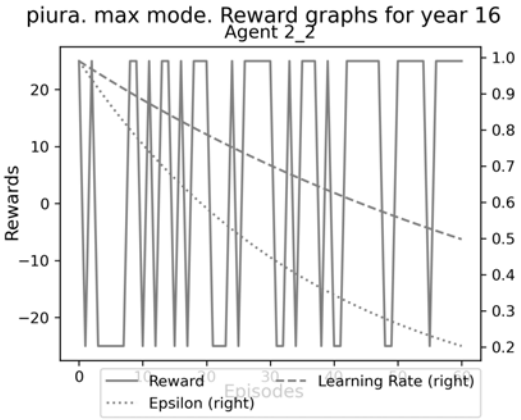


Learning process of all agents in Piura Competitive scenario for energy use Maximisation (26)

() Performance

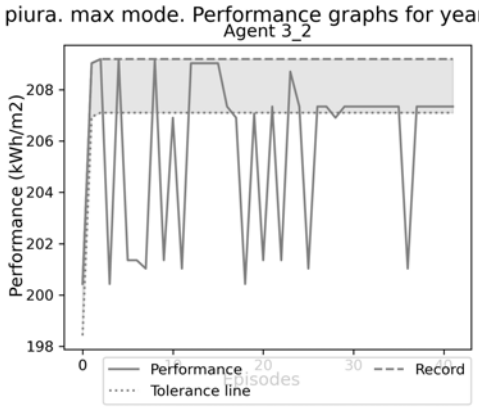


() Rewards

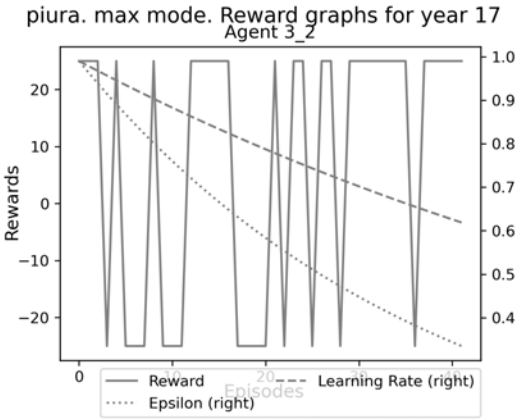


Learning process of all agents in Piura Competitive scenario for energy use Maximisation (27)

() Performance

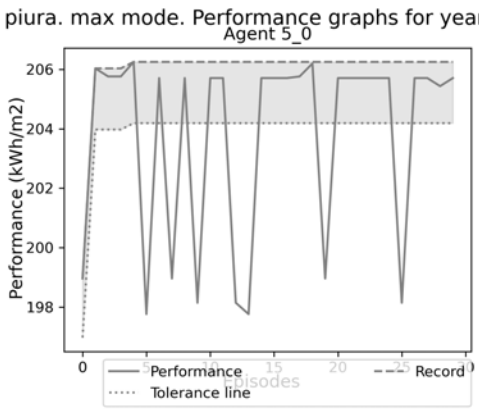


() Rewards

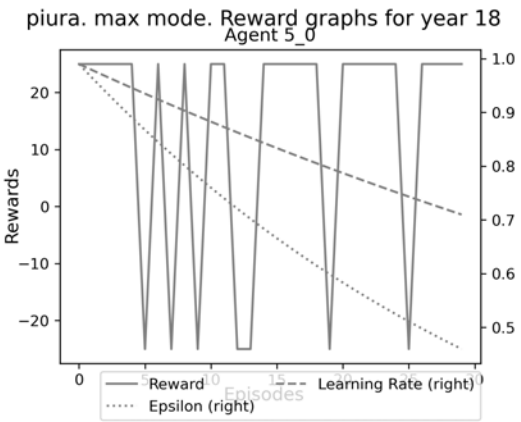


Learning process of all agents in Piura Competitive scenario for energy use Maximisation (28)

() Performance

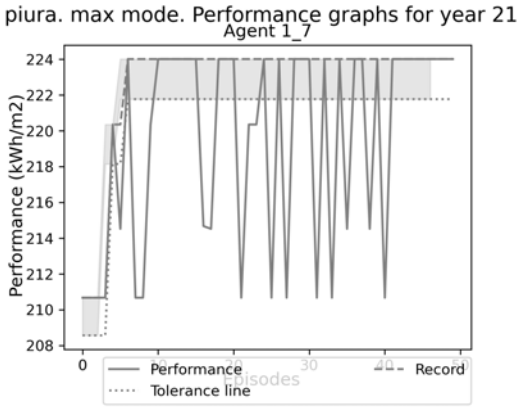


() Rewards

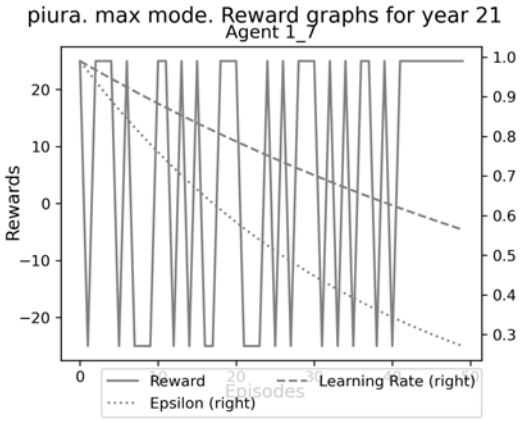


Learning process of all agents in Piura Competitive scenario for energy use Maximisation (29)

() Performance

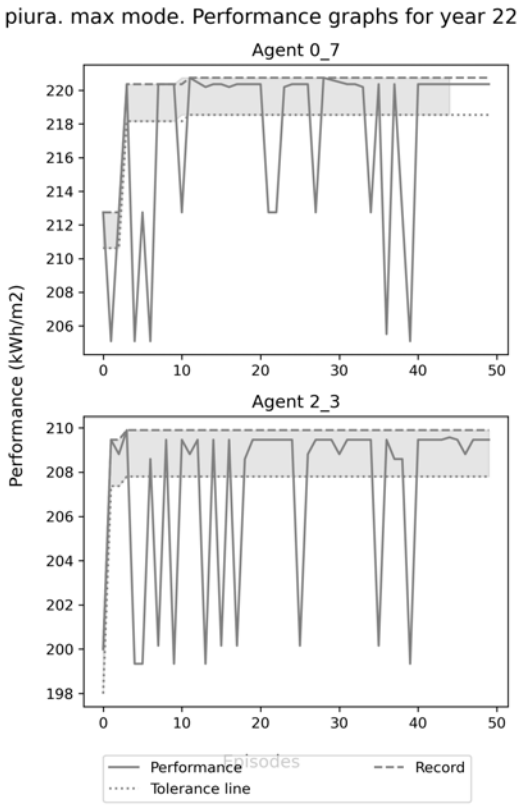


() Rewards

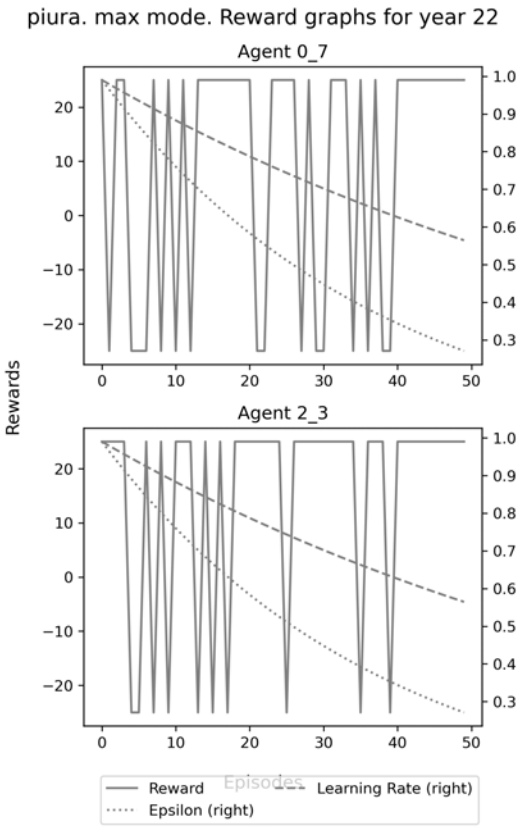


Learning process of all agents in Piura Competitive scenario for energy use Maximisation (30)

() Performance



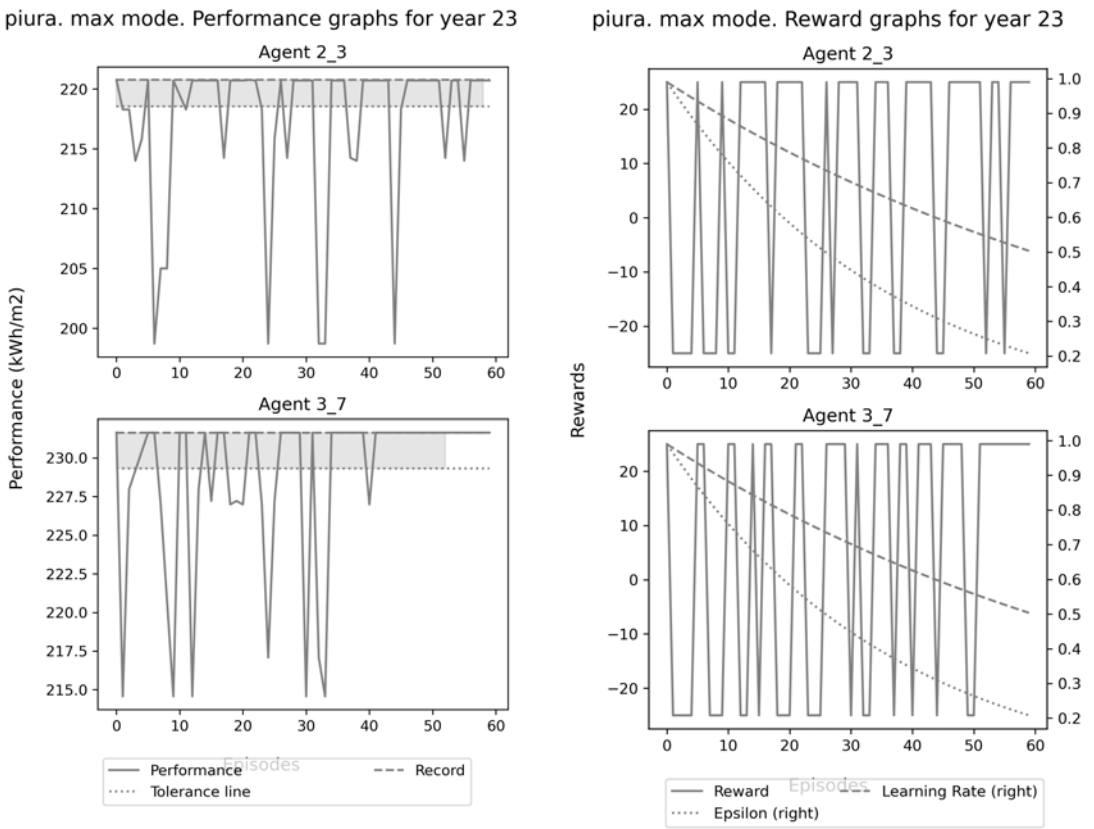
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (31)

() Performance

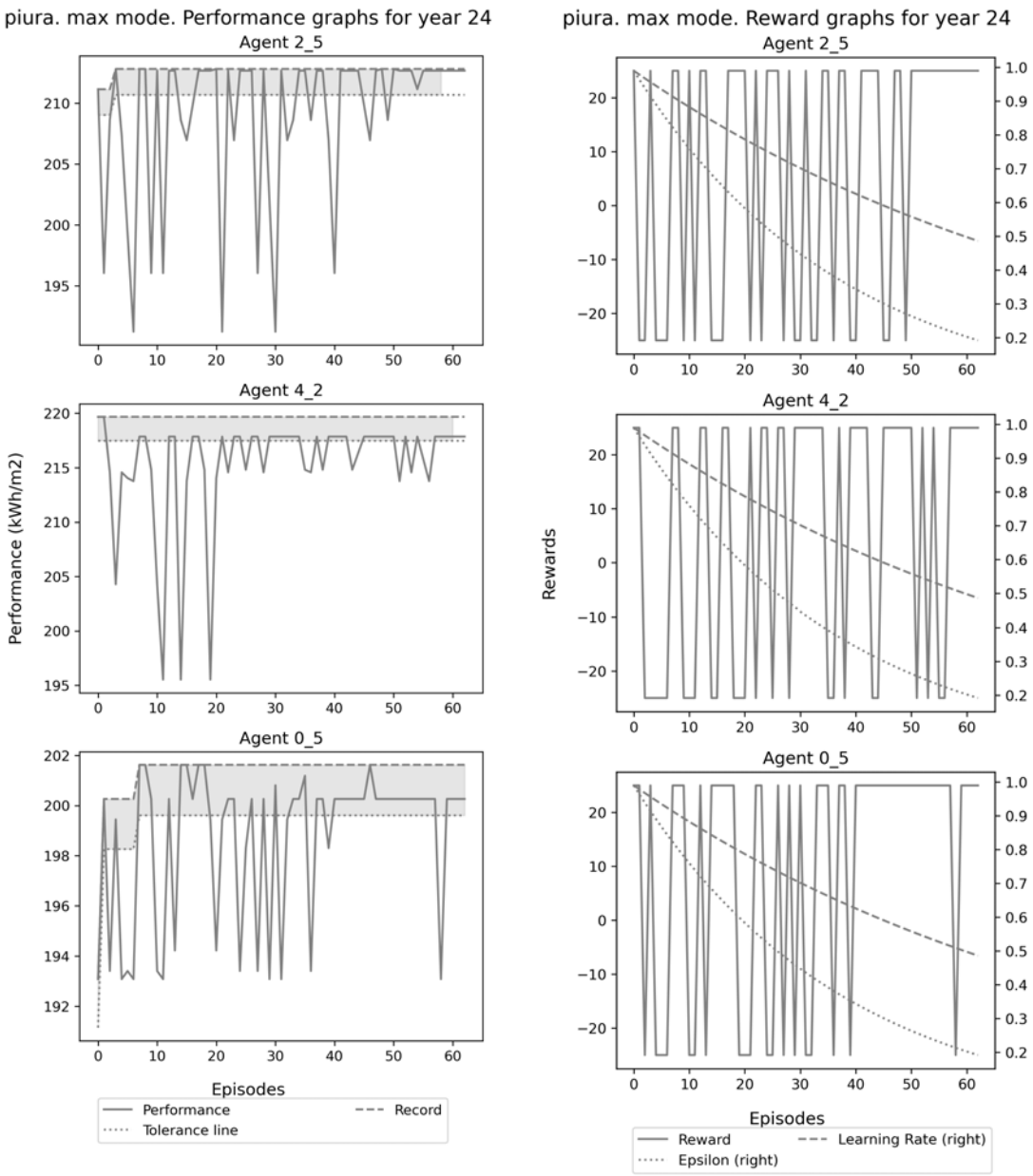
() Rewards



Learning process of all agents in Piura Competitive scenario for energy use Maximisation (32)

() Performance

() Rewards

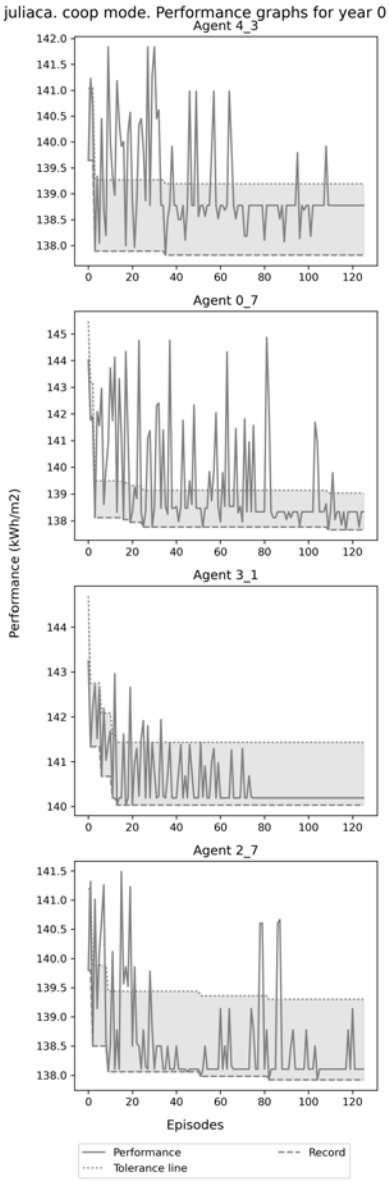


Learning process of all agents in Piura Competitive scenario for energy use Maximisation (33)

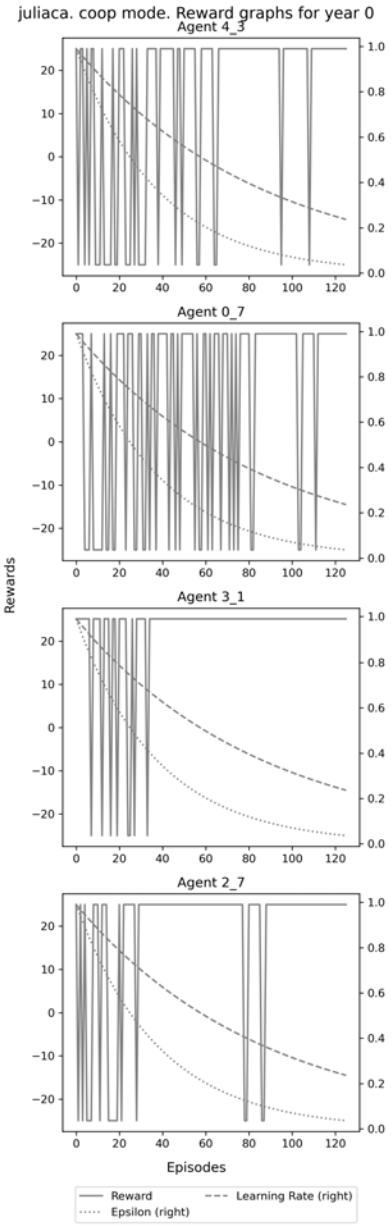
Appendix 18

Learning process by agent for each simulation year (Juliaca, Minimisation, cooperative)

() Performance



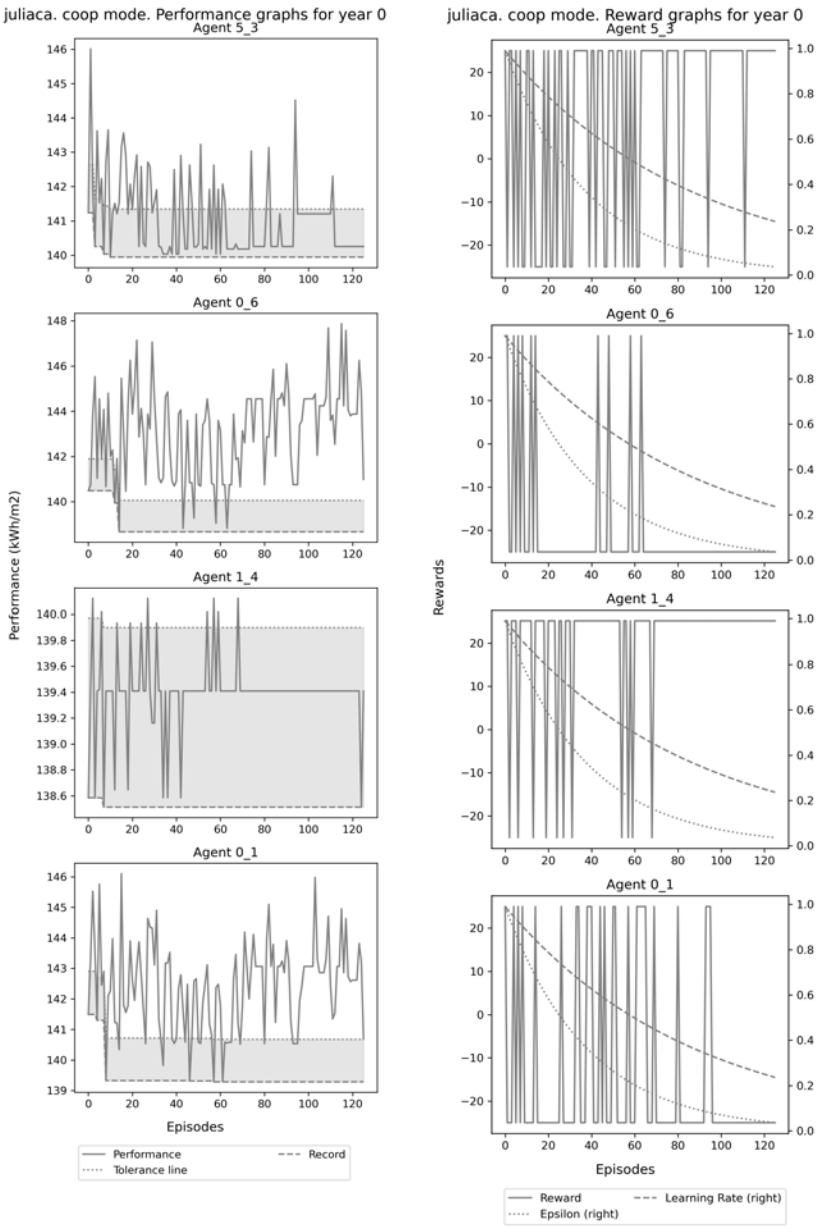
() Rewards



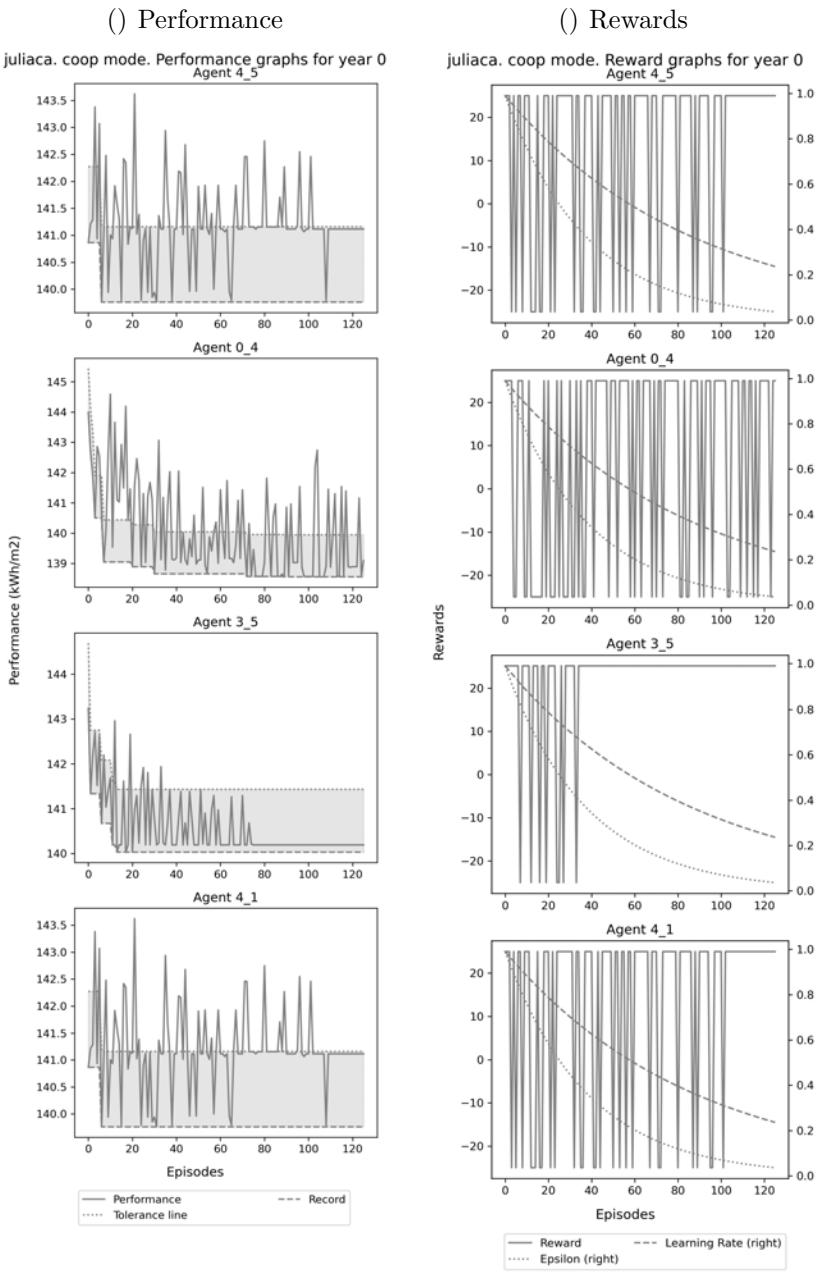
Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (1)

() Performance

() Rewards



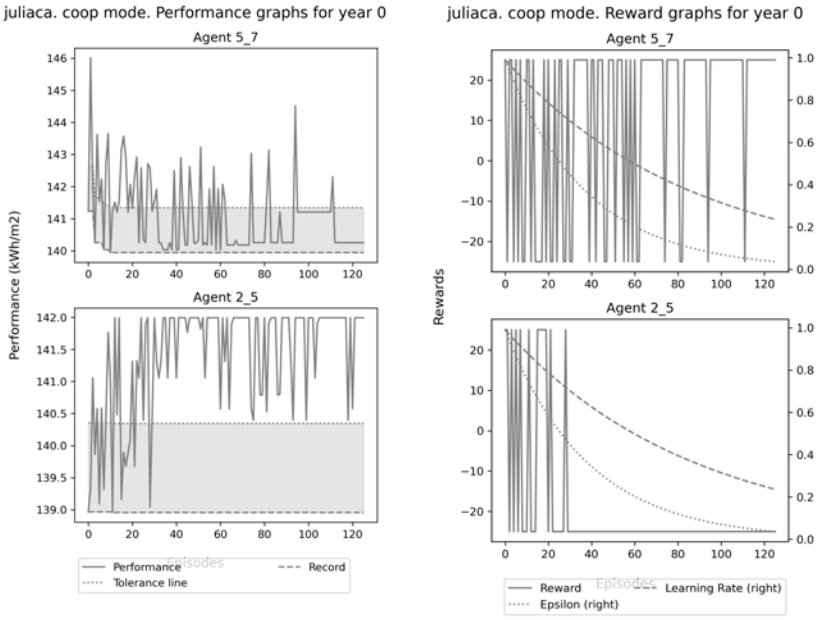
Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (2)



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (3)

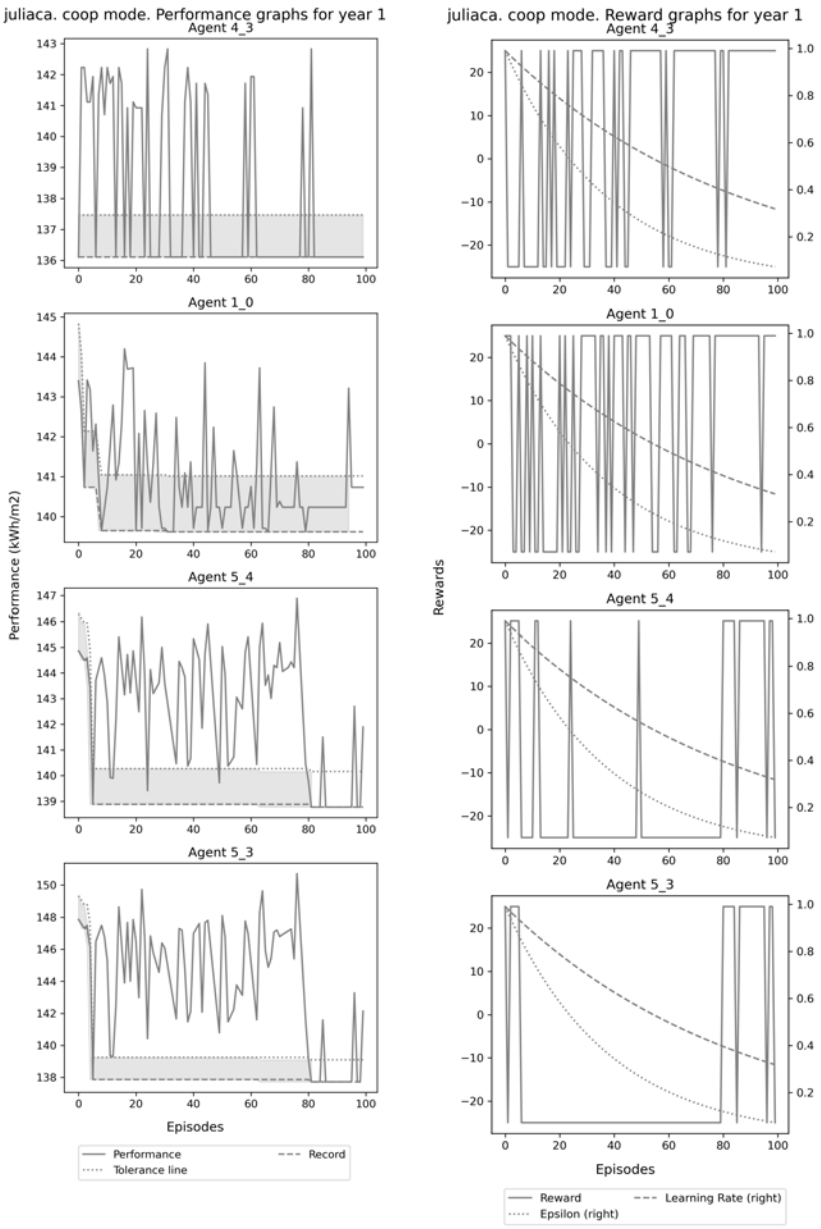
() Performance

() Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (4)

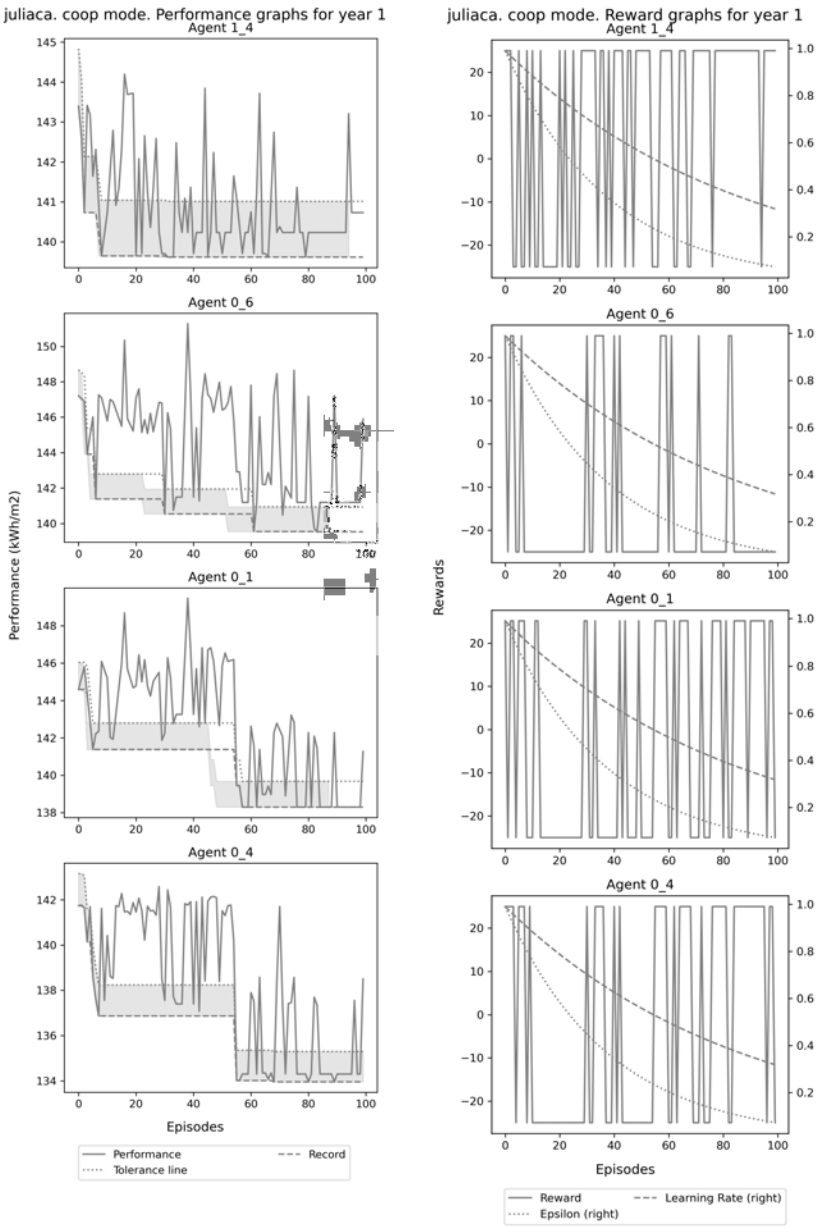
() Performance () Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (5)

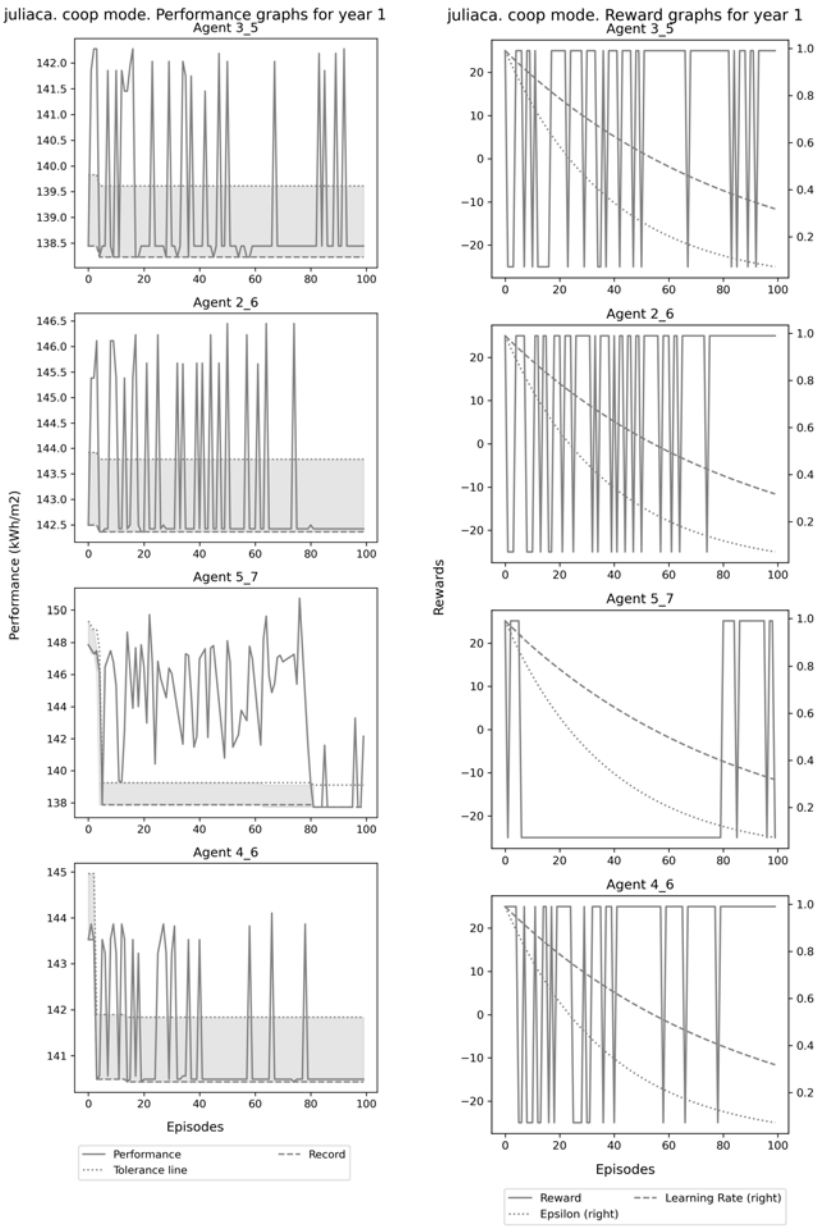
() Performance

() Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (6)

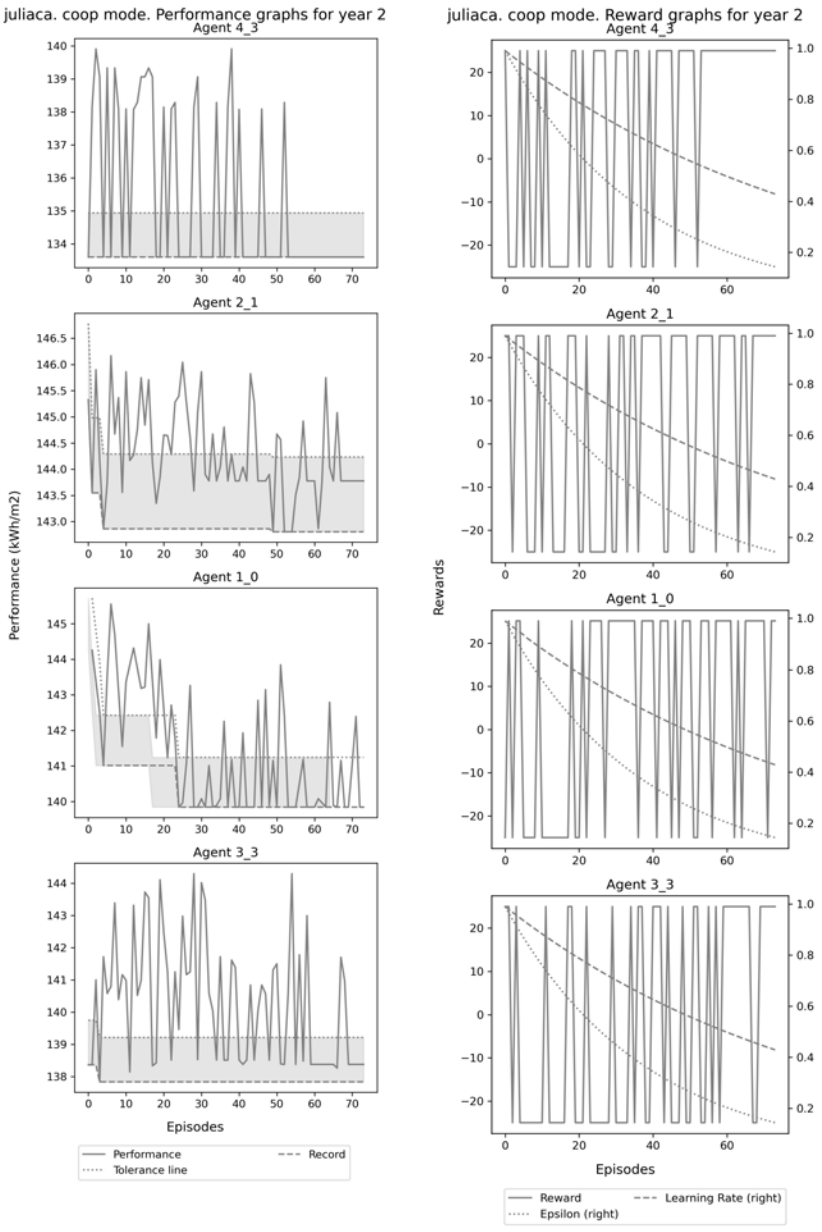
() Performance () Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (7)

() Performance

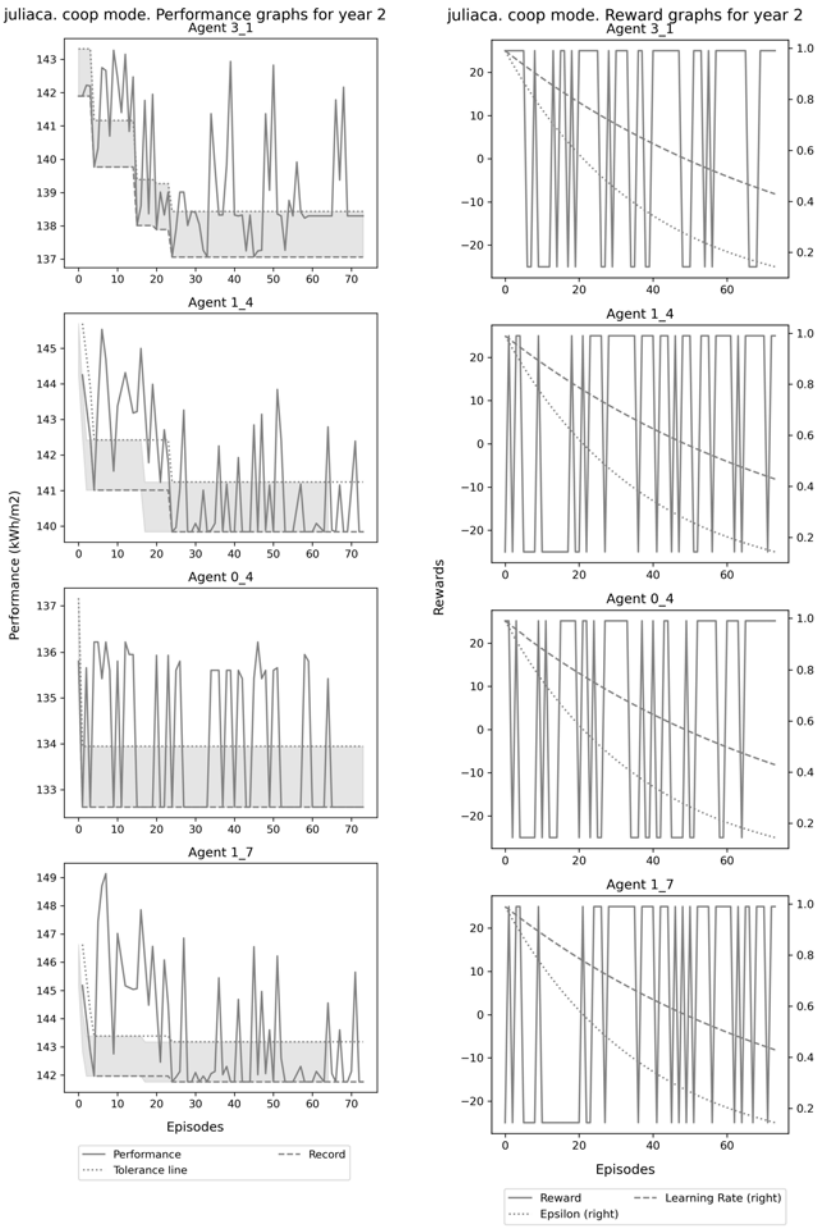
() Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (8)

() Performance

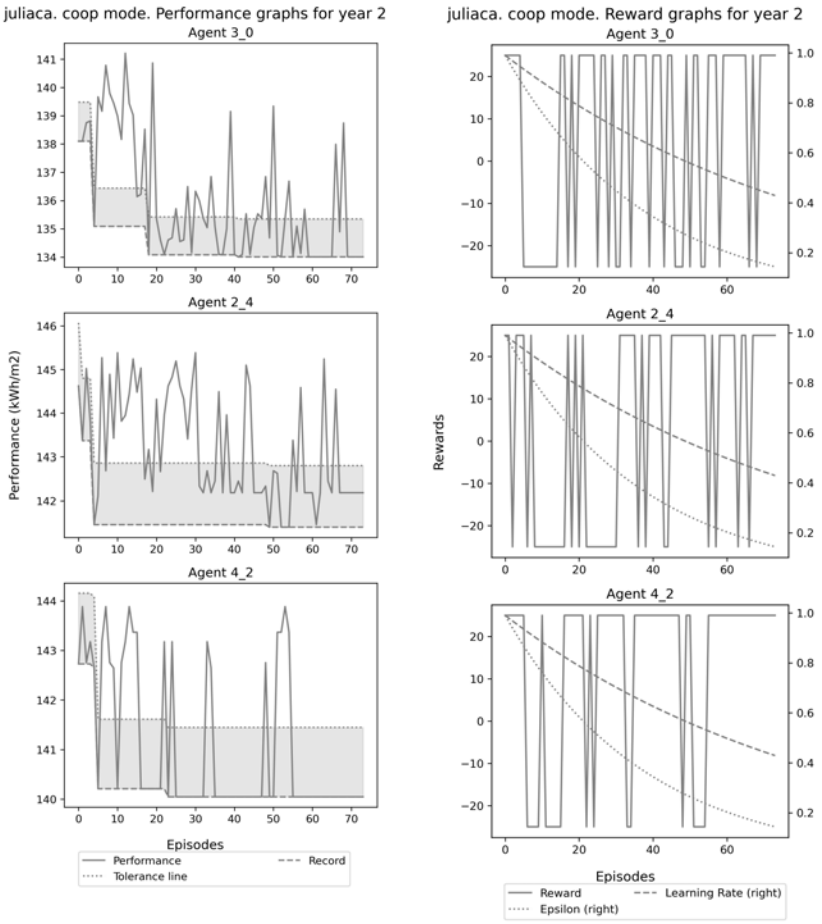
() Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (9)

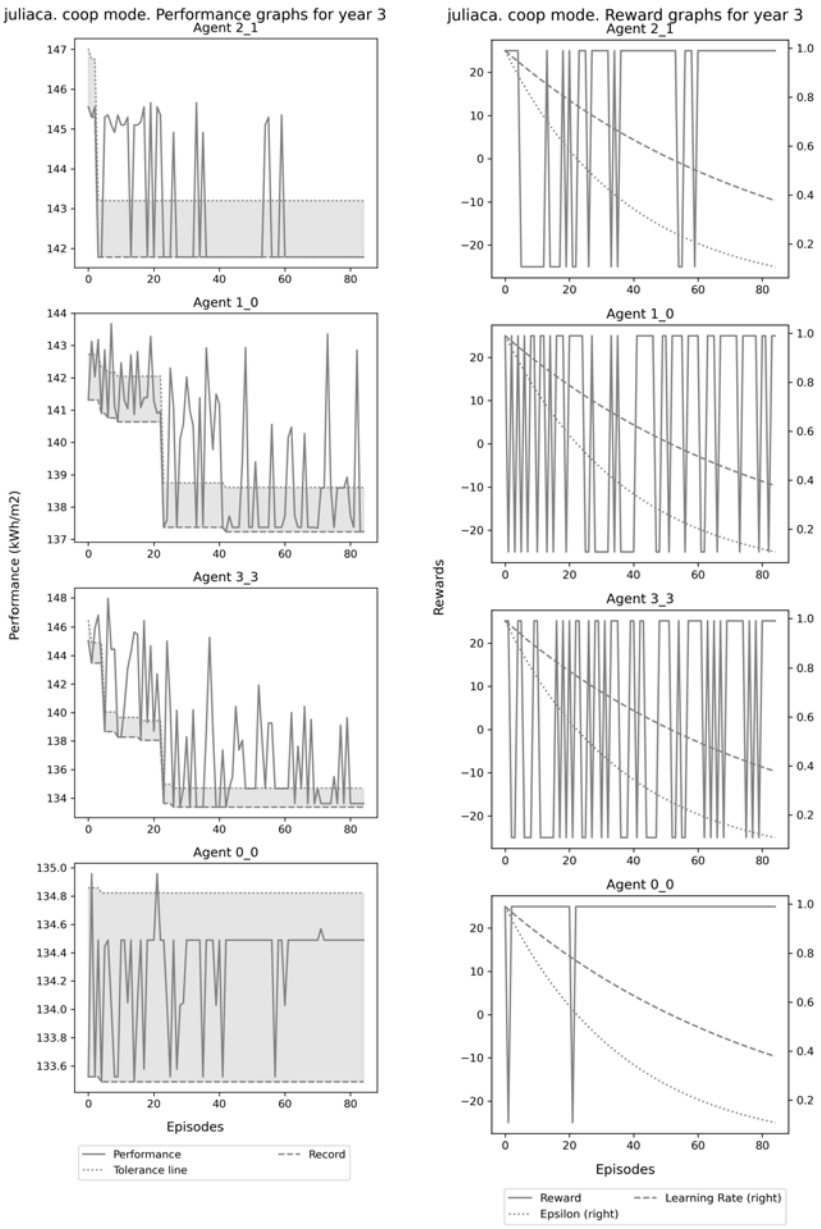
() Performance

() Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (10)

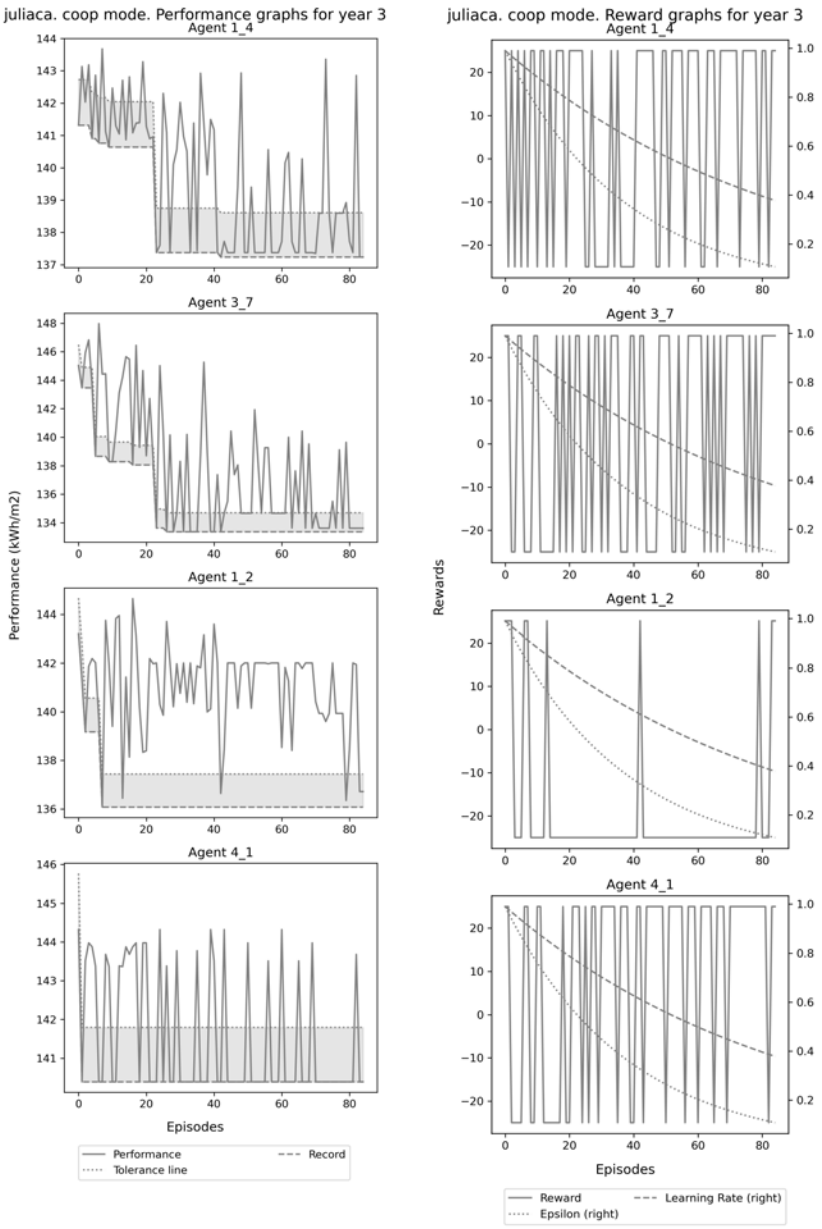
() Performance () Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (11)

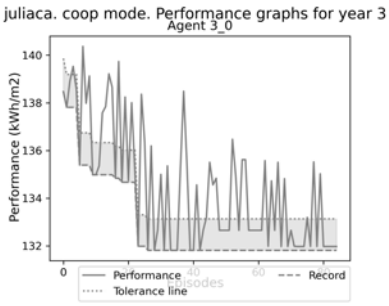
() Performance

() Rewards

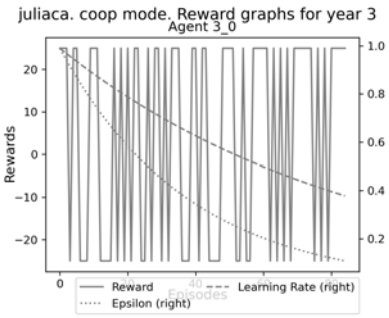


Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (12)

() Performance



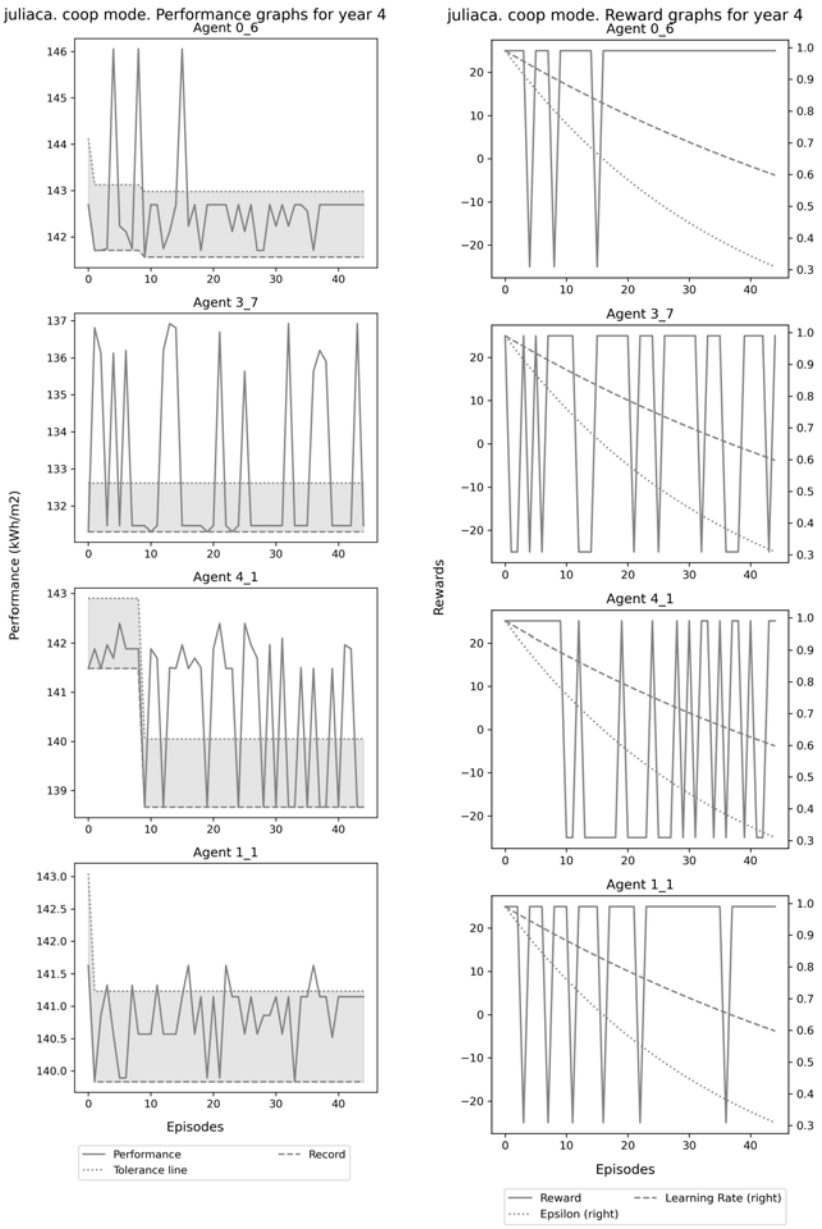
() Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (13)

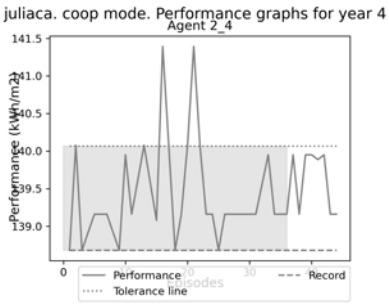
() Performance

() Rewards

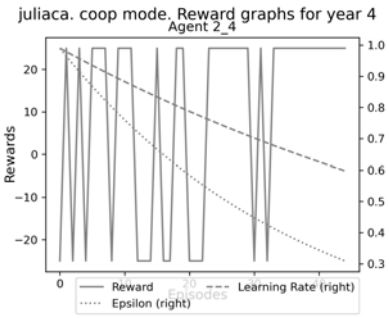


Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (14)

() Performance



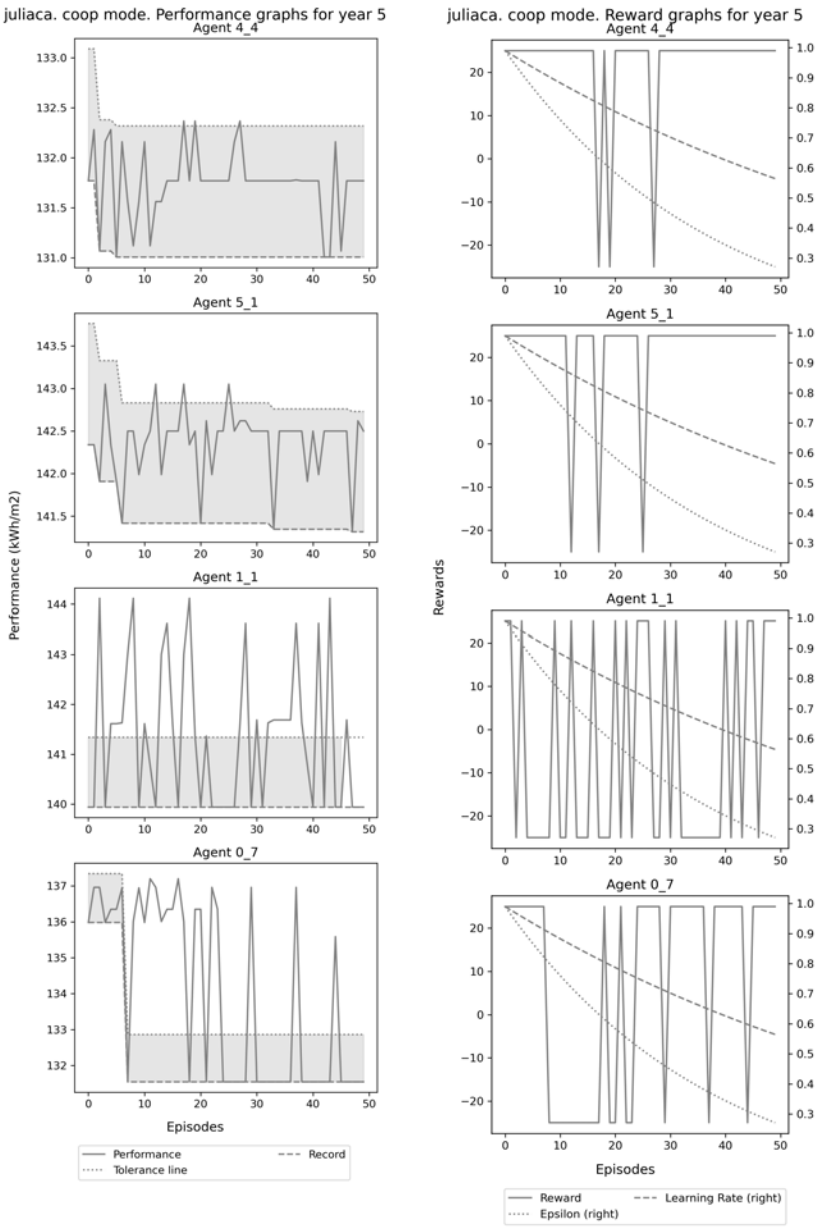
() Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (15)

() Performance

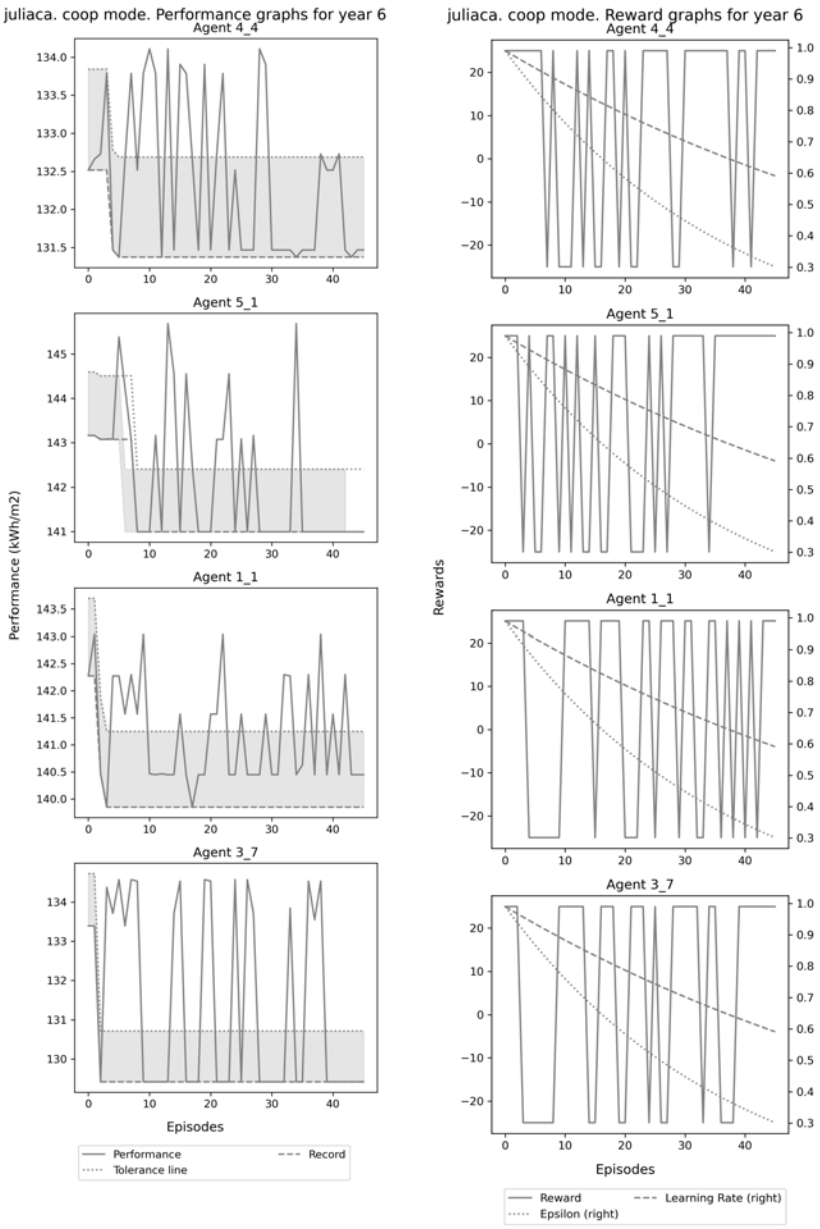
() Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (16)

() Performance

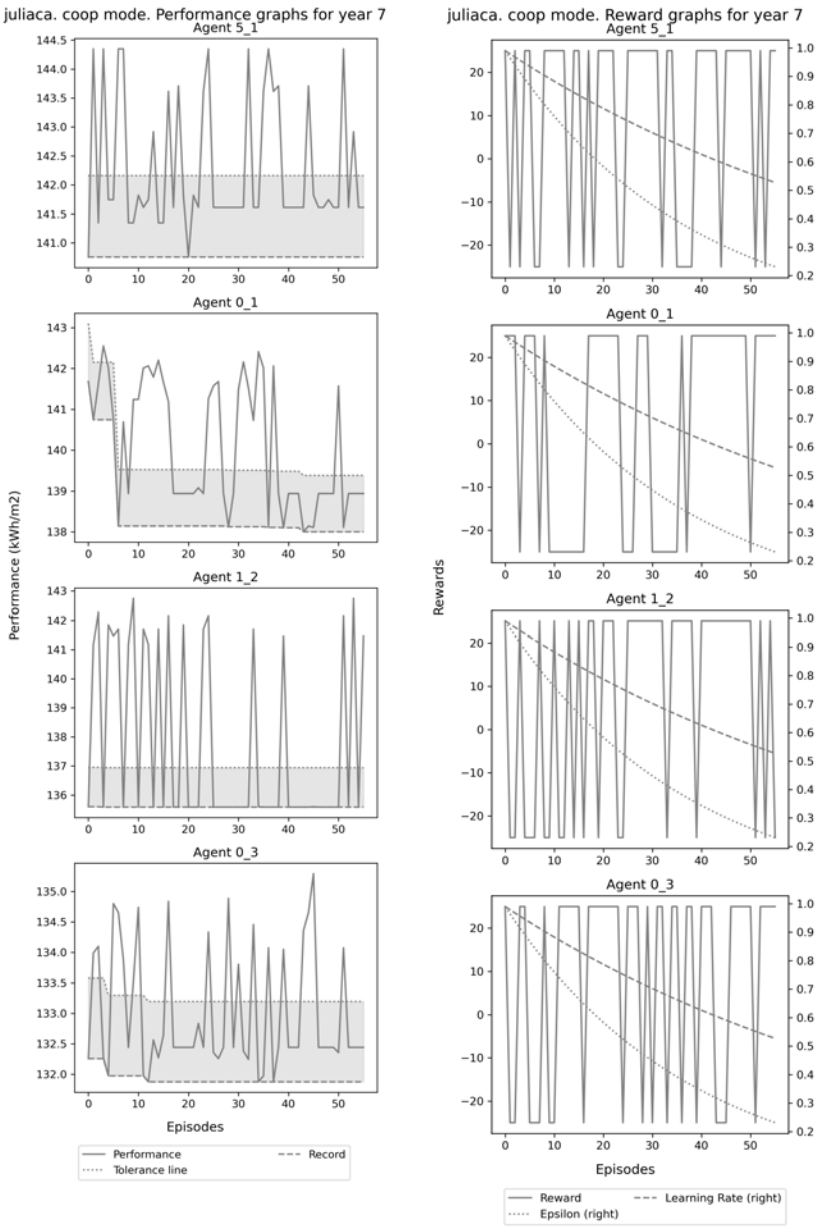
() Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (17)

() Performance

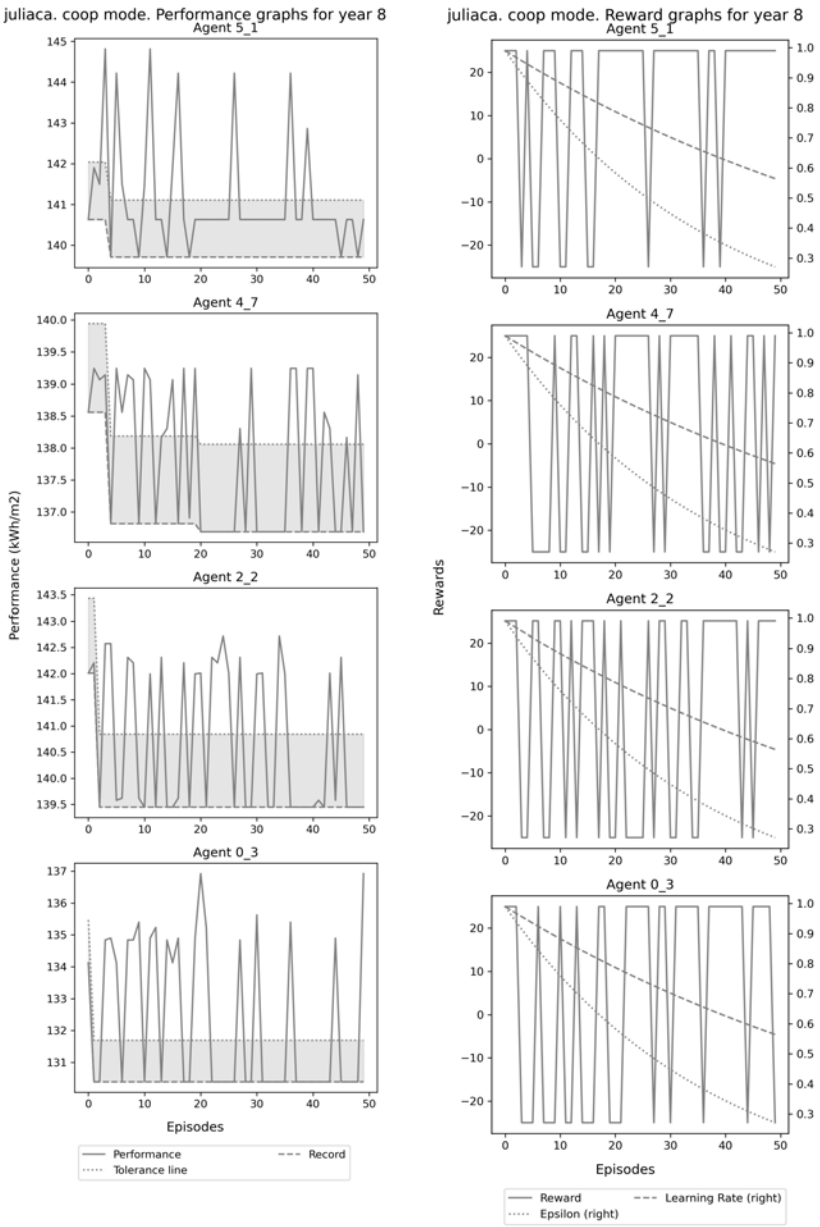
() Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (18)

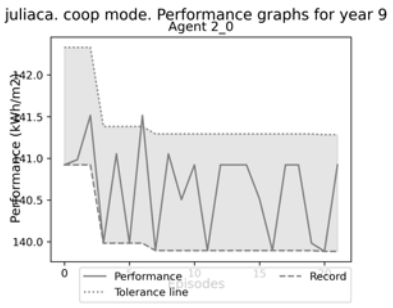
() Performance

() Rewards

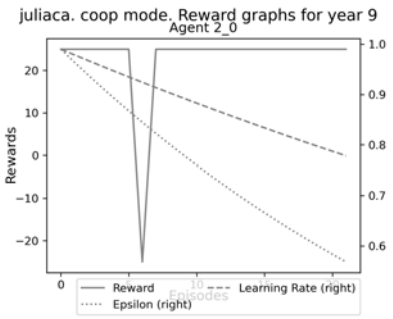


Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (19)

() Performance

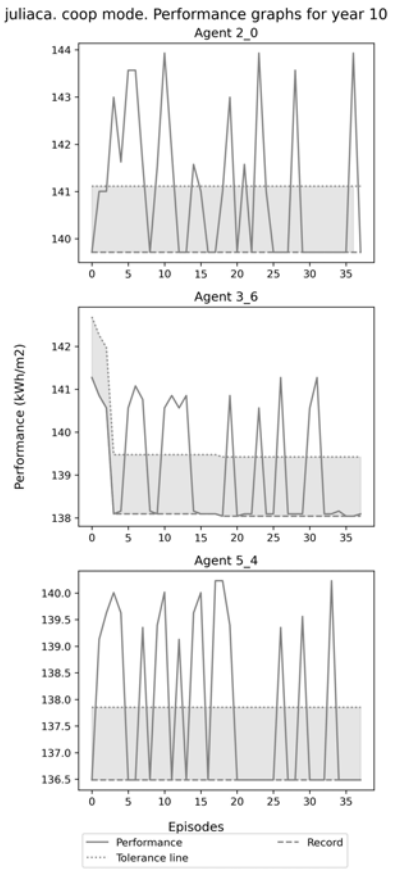


() Rewards

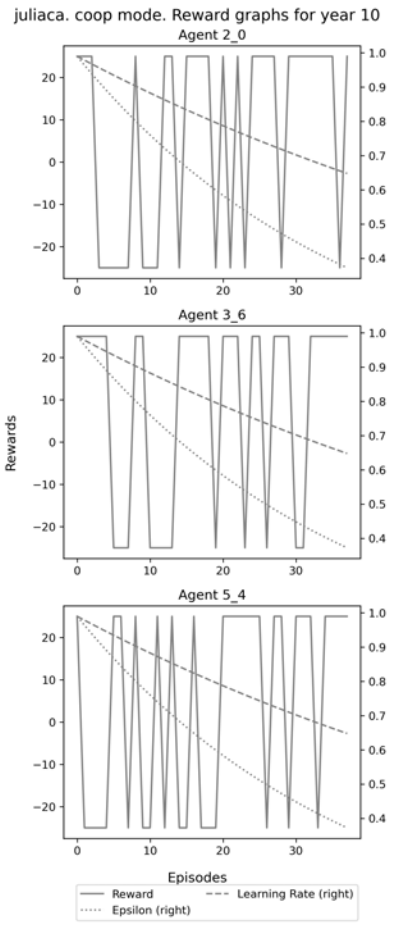


Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (20)

() Performance



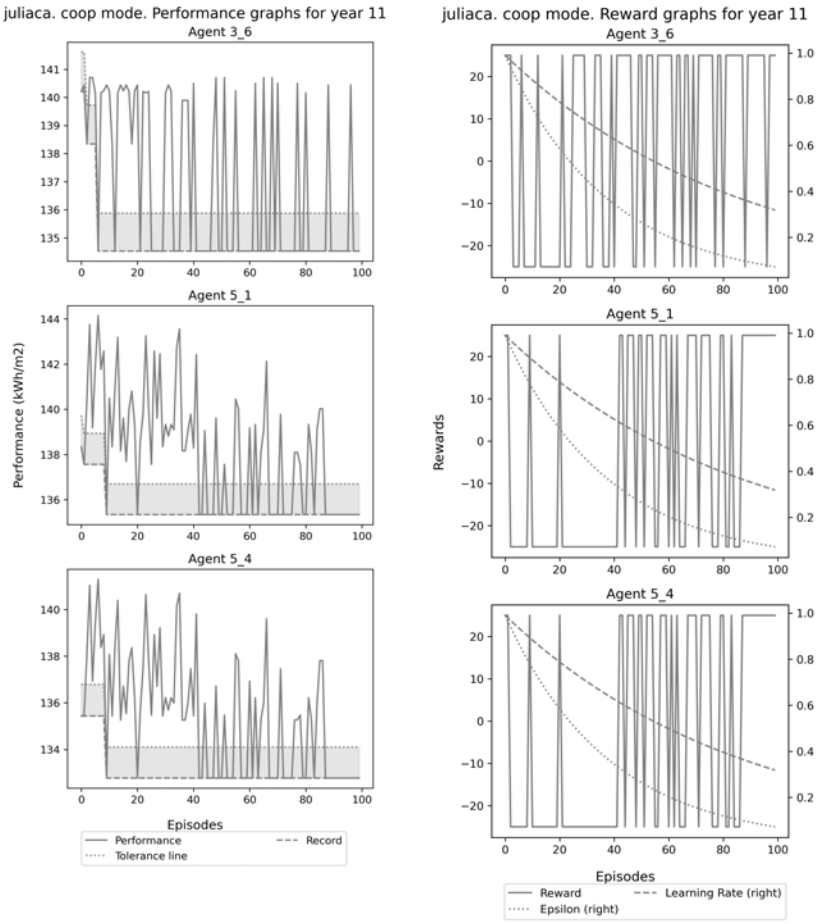
() Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (21)

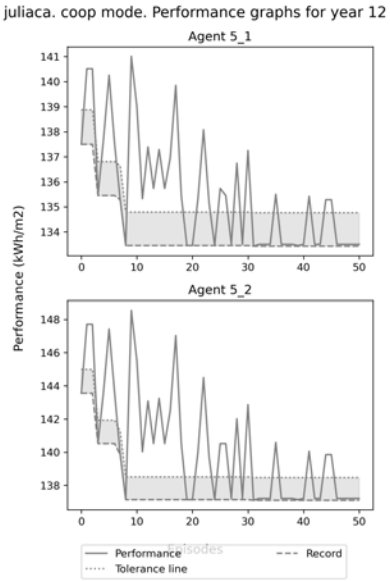
() Performance

() Rewards

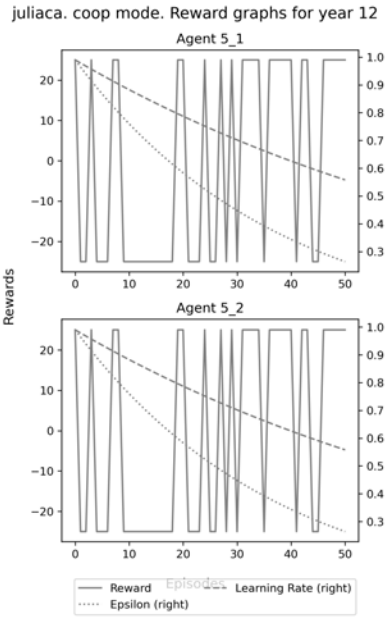


Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (22)

() Performance



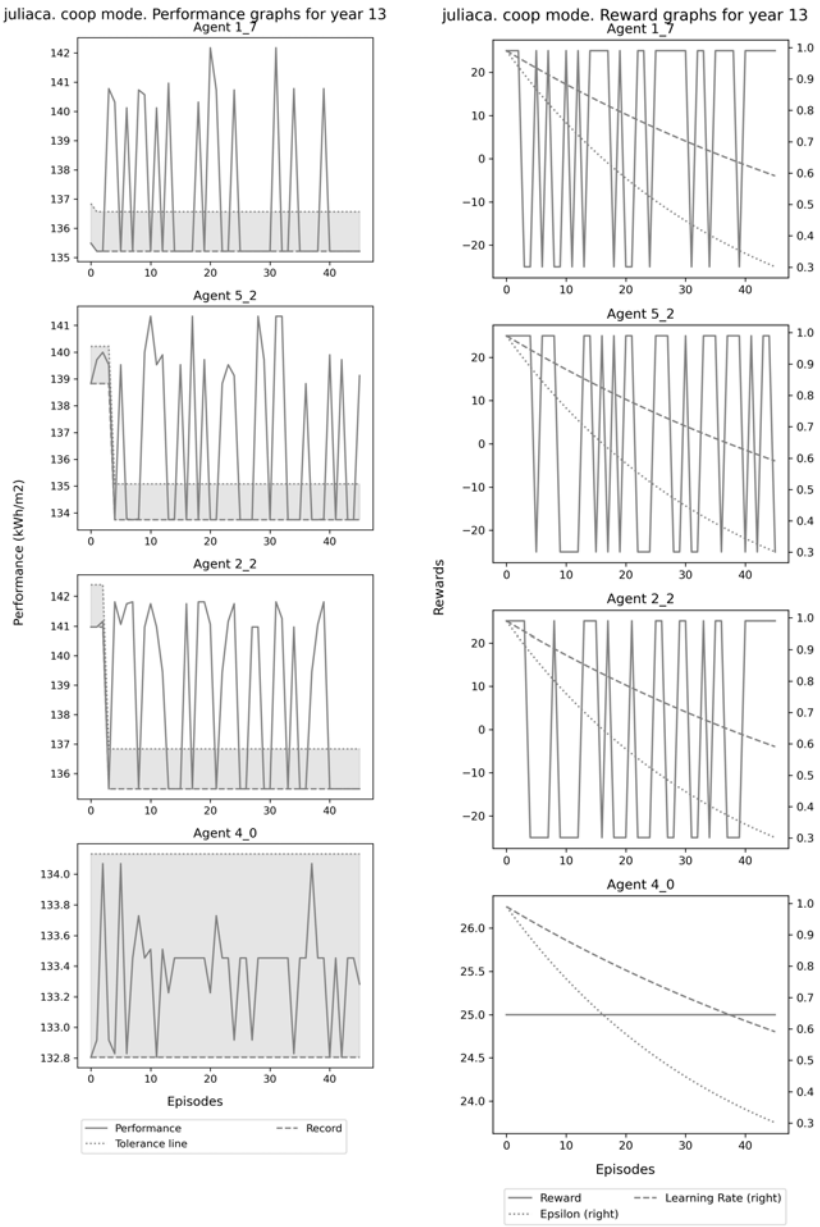
() Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (23)

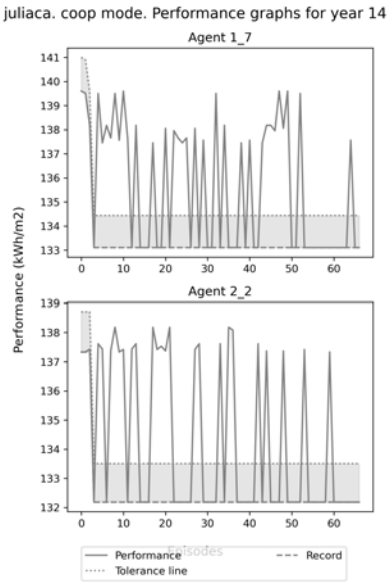
() Performance

() Rewards

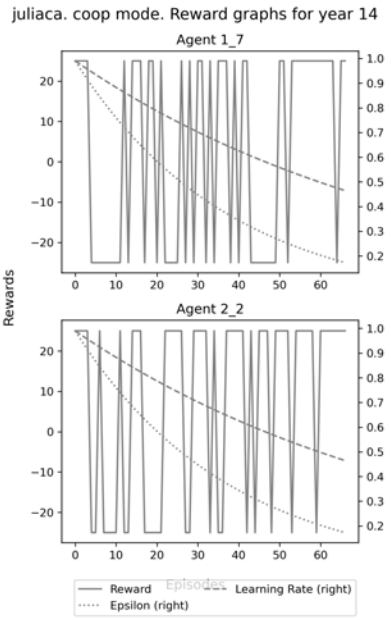


Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (24)

() Performance

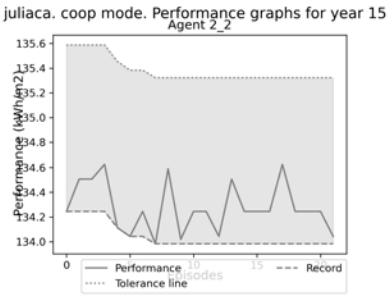


() Rewards

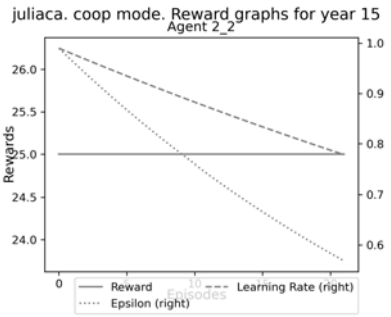


Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (25)

() Performance

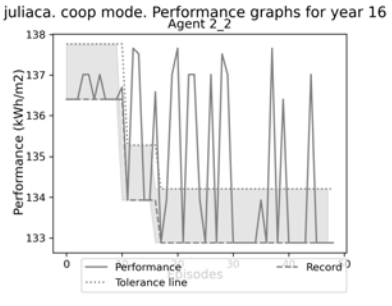


() Rewards

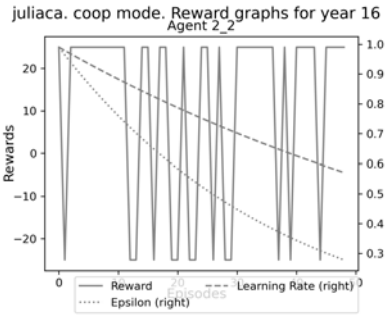


Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (26)

() Performance

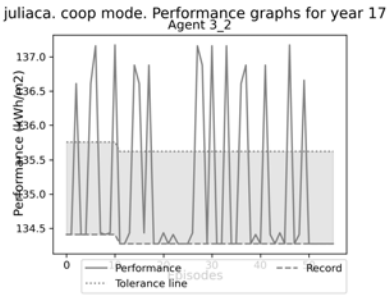


() Rewards

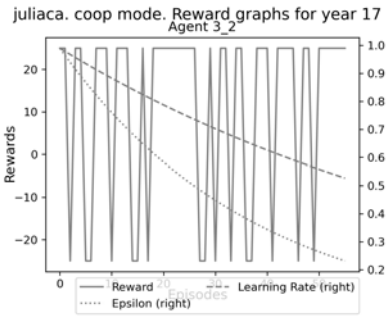


Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (27)

() Performance

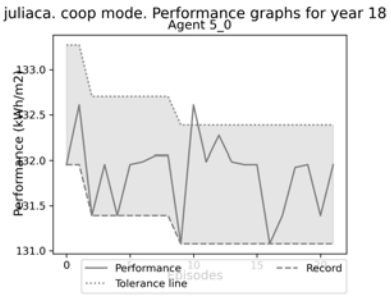


() Rewards

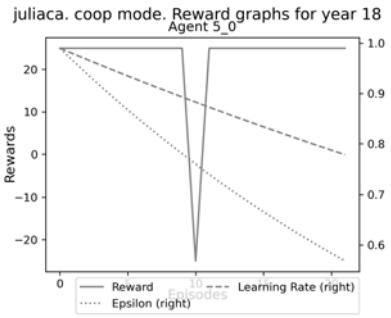


Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (28)

() Performance

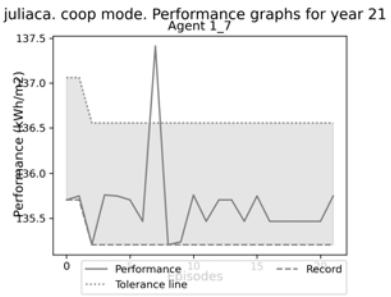


() Rewards

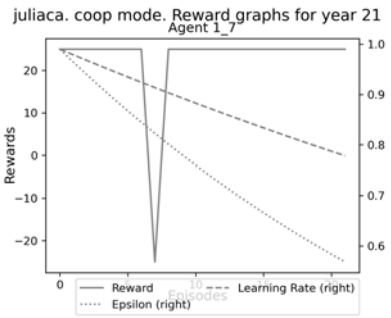


Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (29)

() Performance

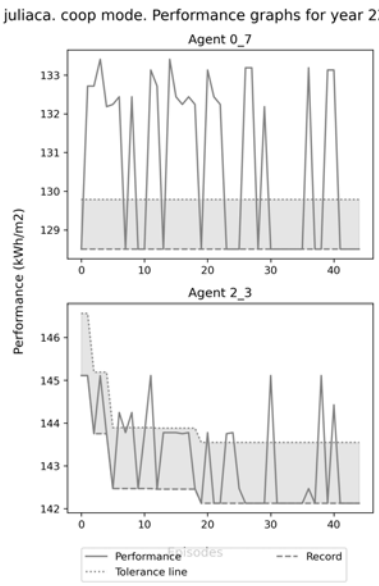


() Rewards

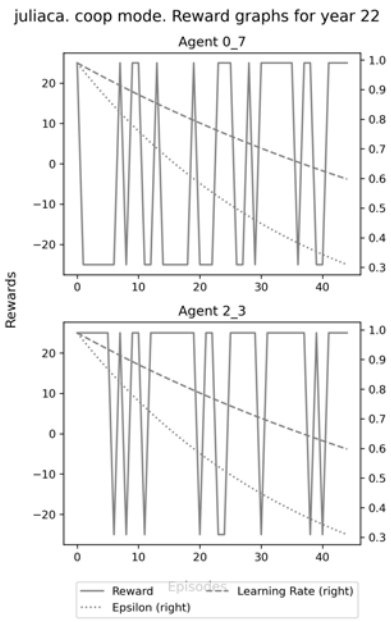


Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (30)

() Performance

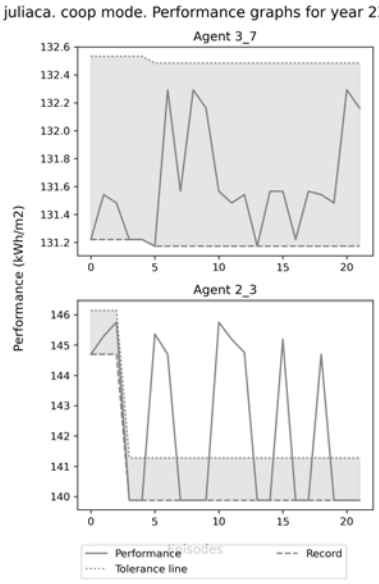


() Rewards

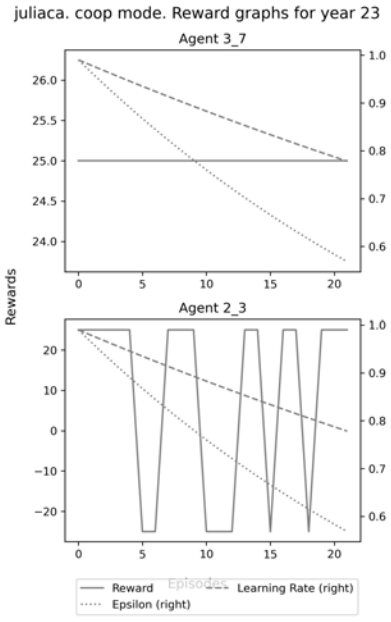


Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (31)

() Performance



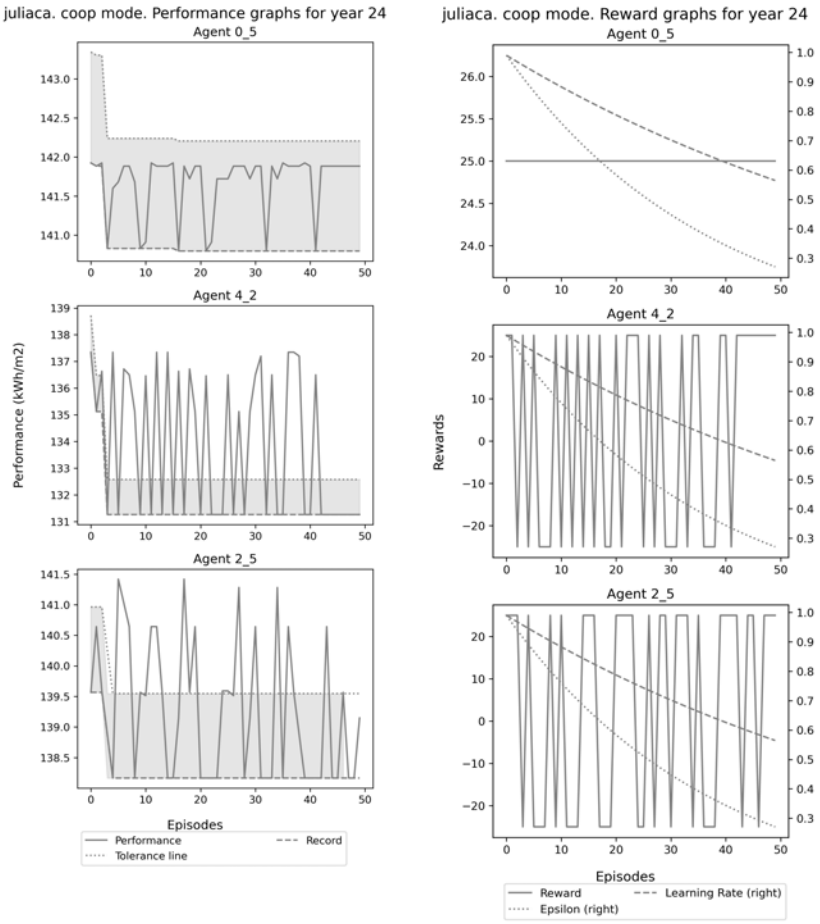
() Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (32)

() Performance

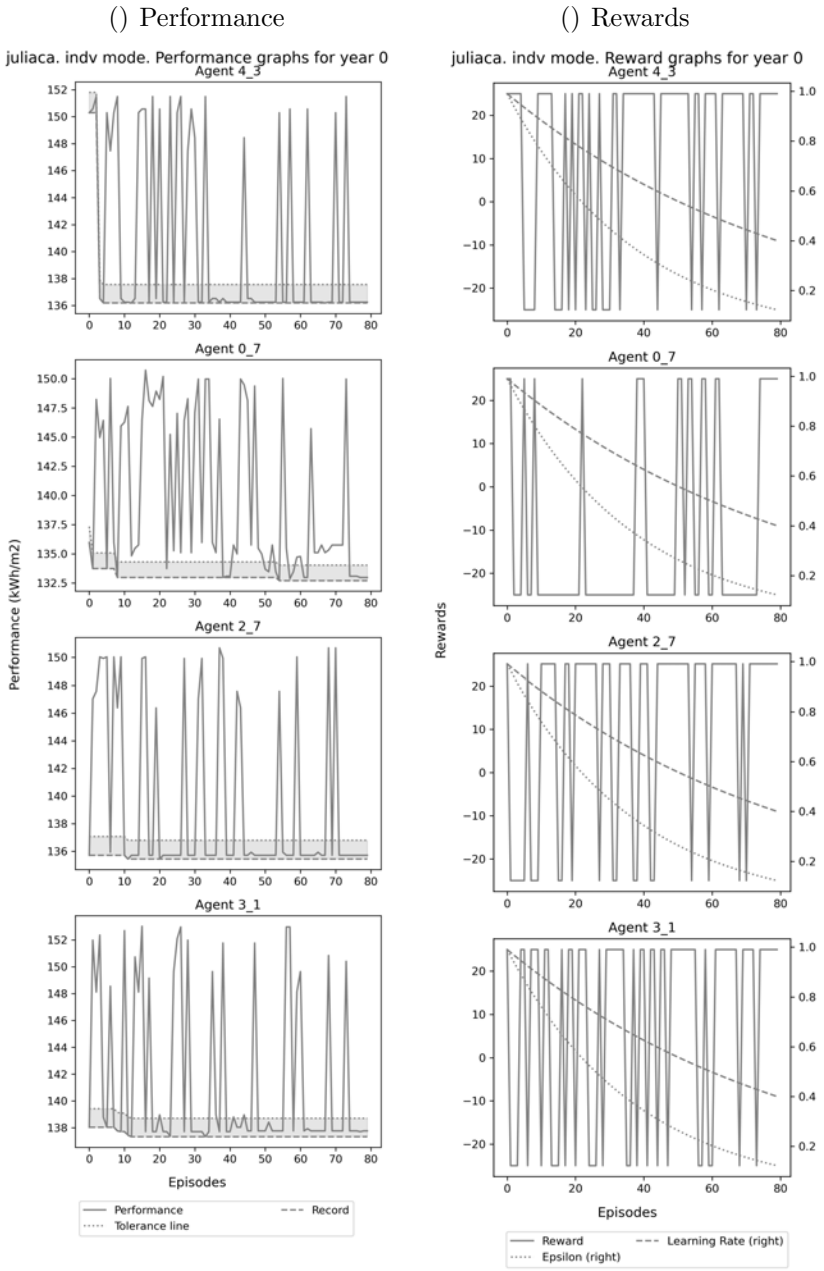
() Rewards



Learning process of all agents in Juliaca cooperative scenario for energy use Minimisation (33)

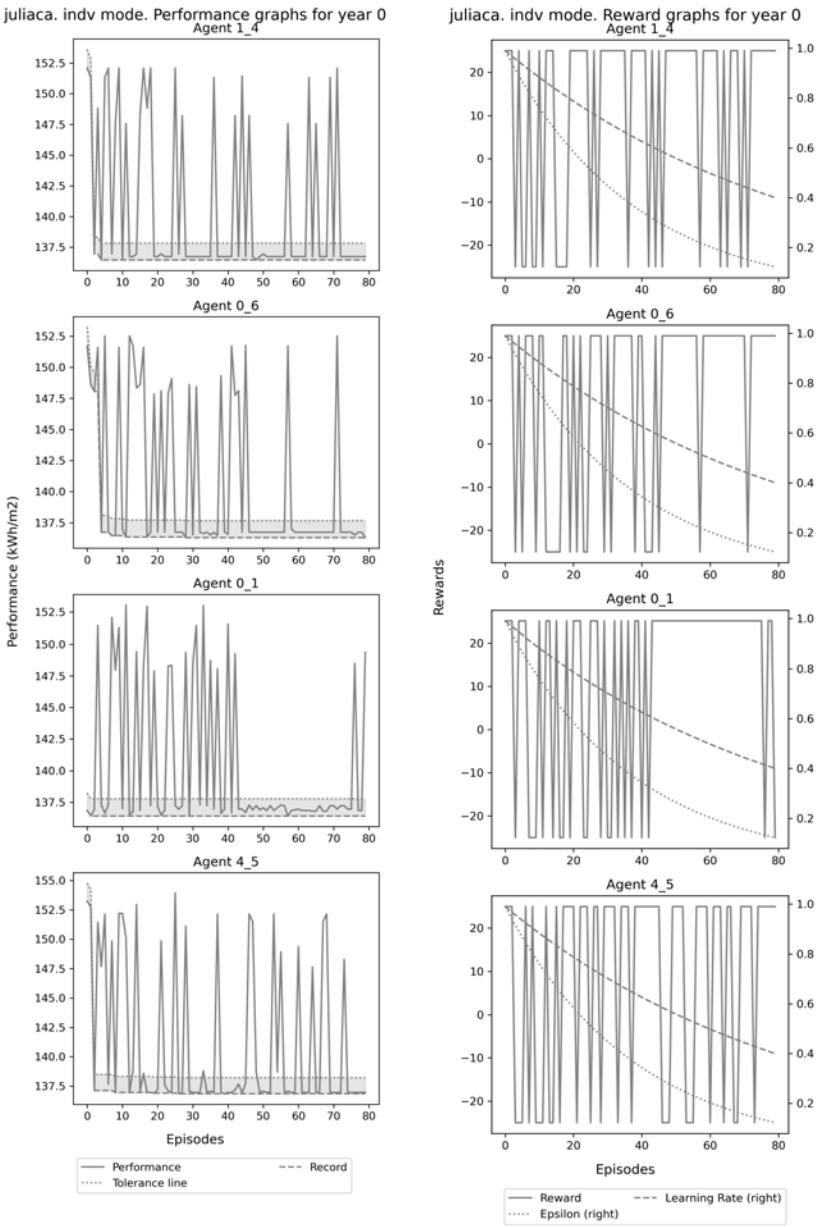
Appendix 19

Learning process by agent for each simulation year (Juliaca, Minimisation, competitive)



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (1)

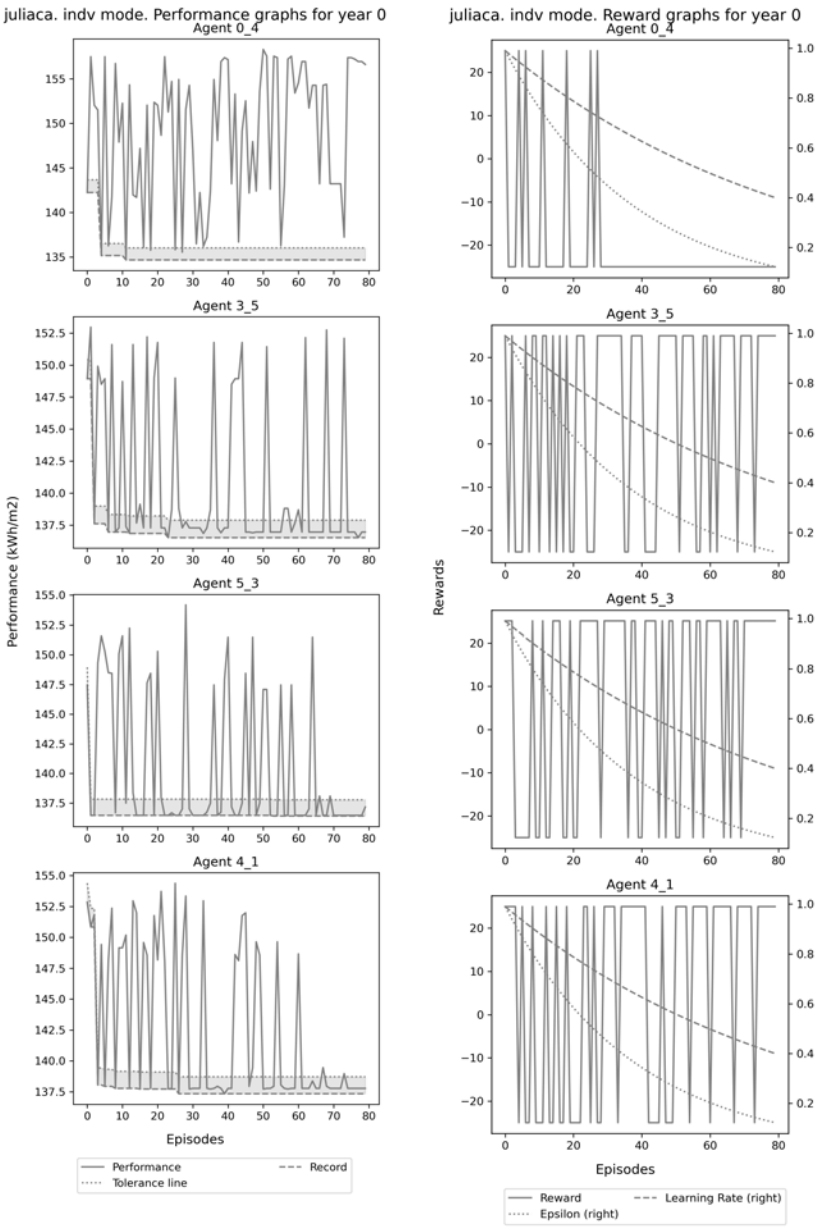
() Performance () Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (2)

() Performance

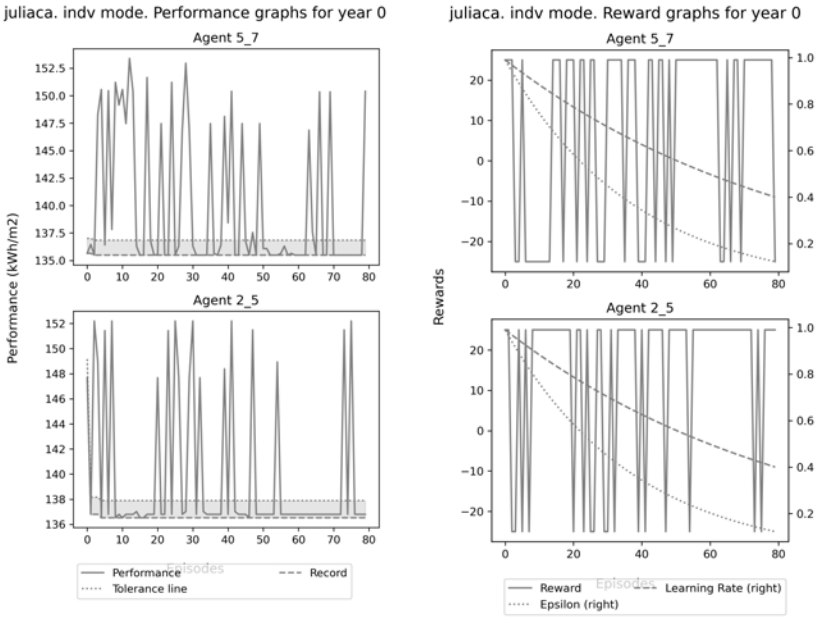
() Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (3)

() Performance

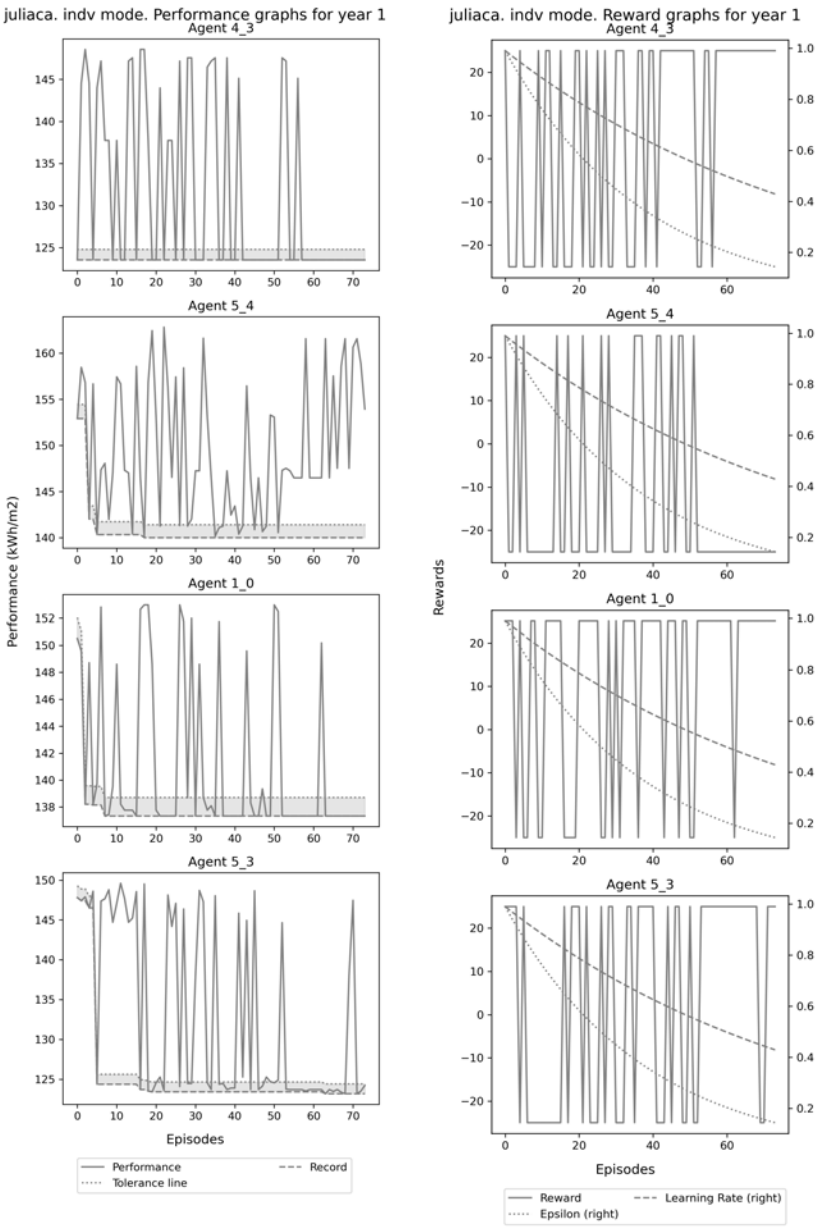
() Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (4)

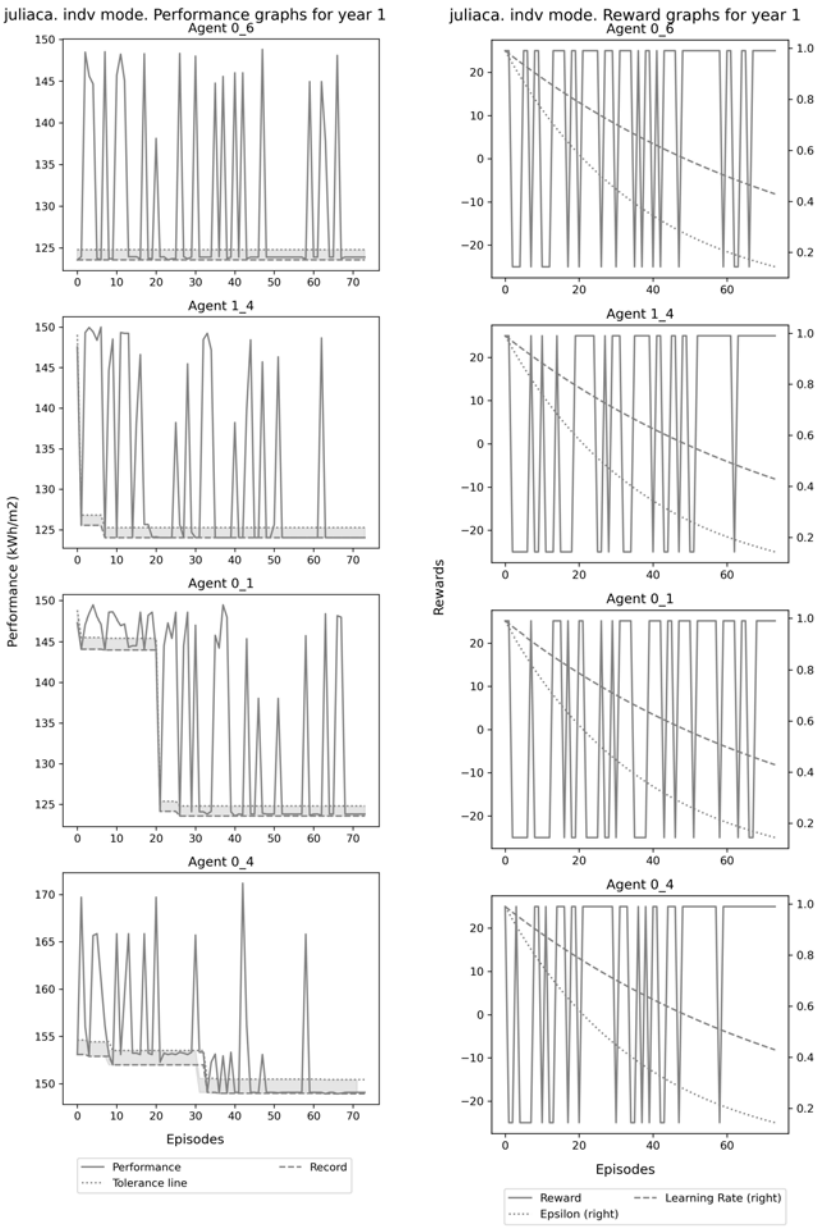
() Performance

() Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (5)

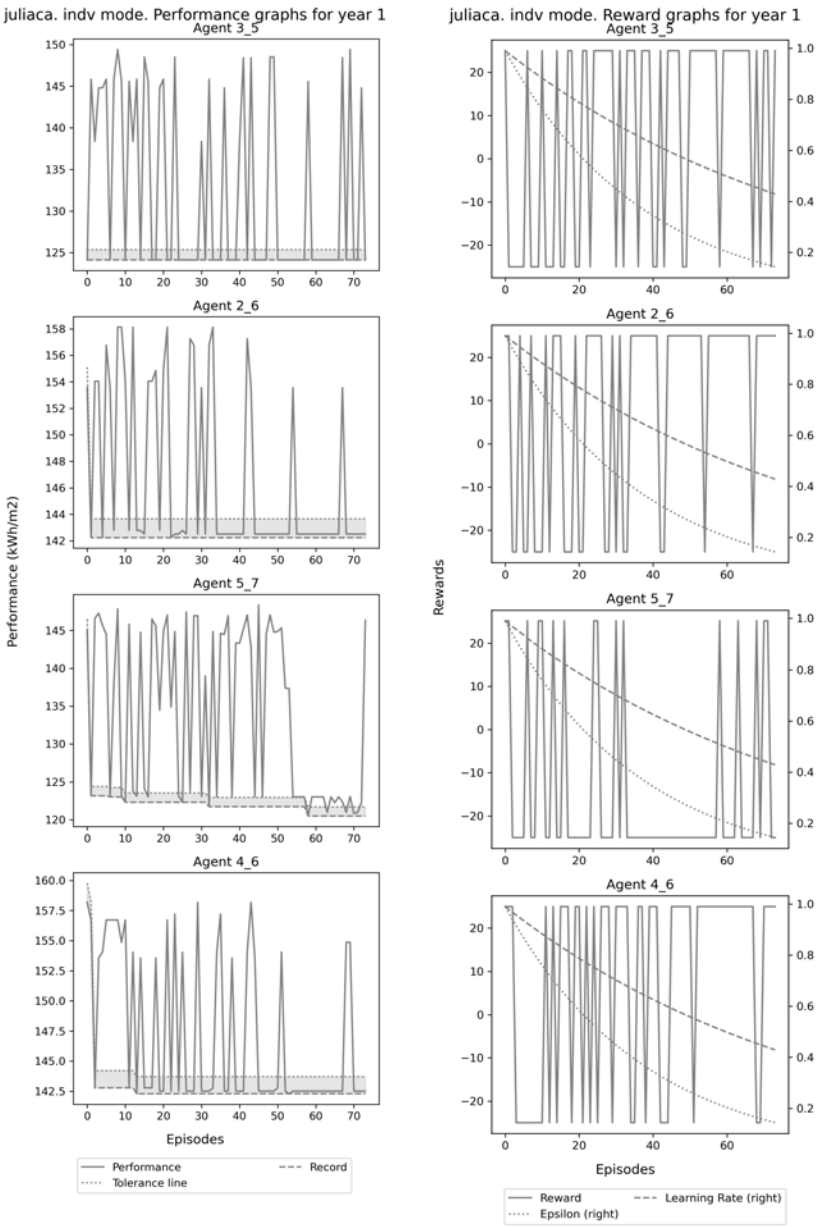
() Performance () Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (6)

() Performance

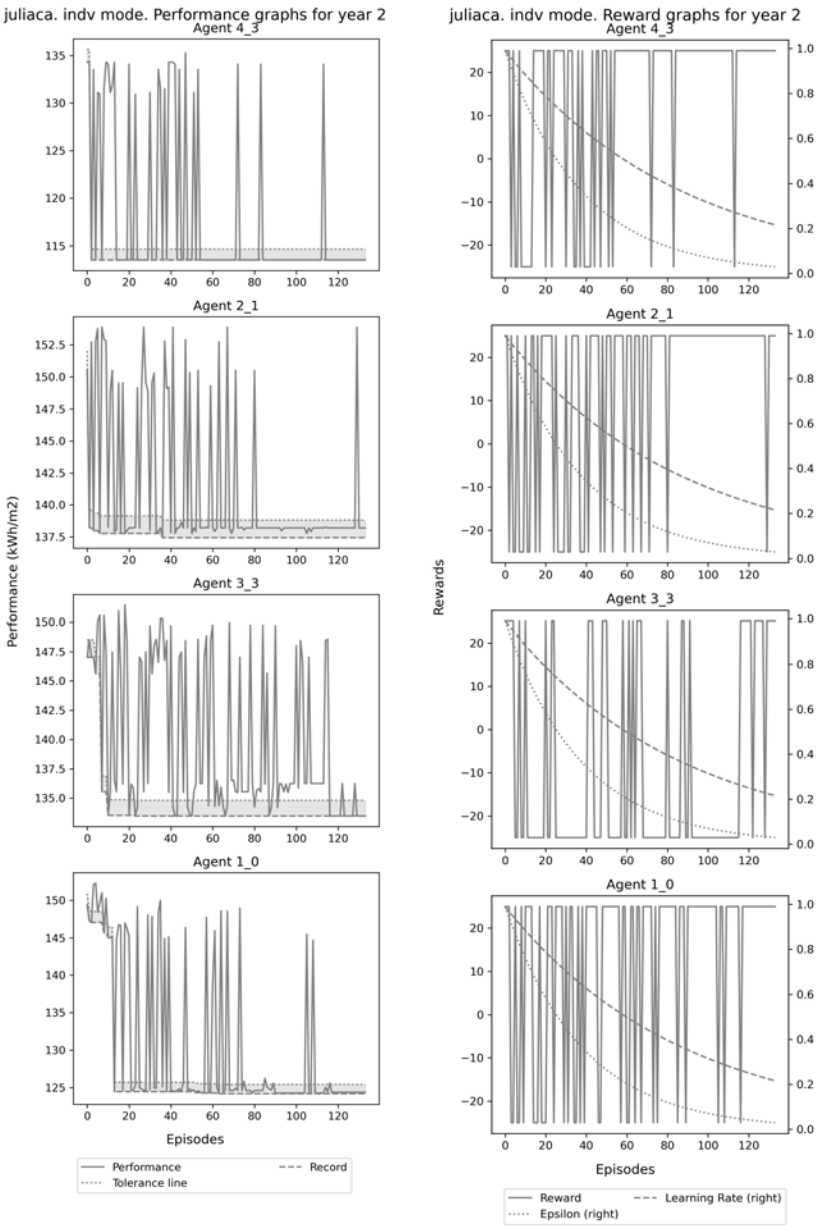
() Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (7)

() Performance

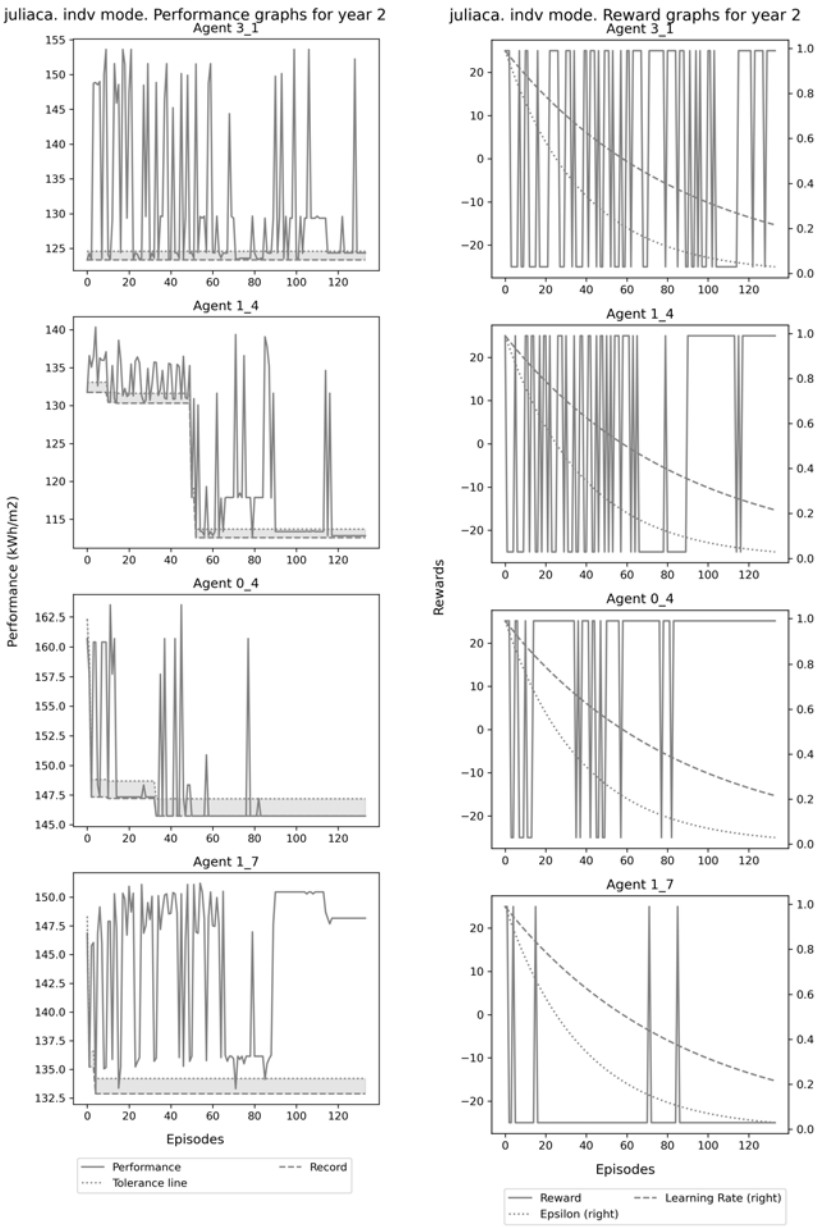
() Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (8)

() Performance

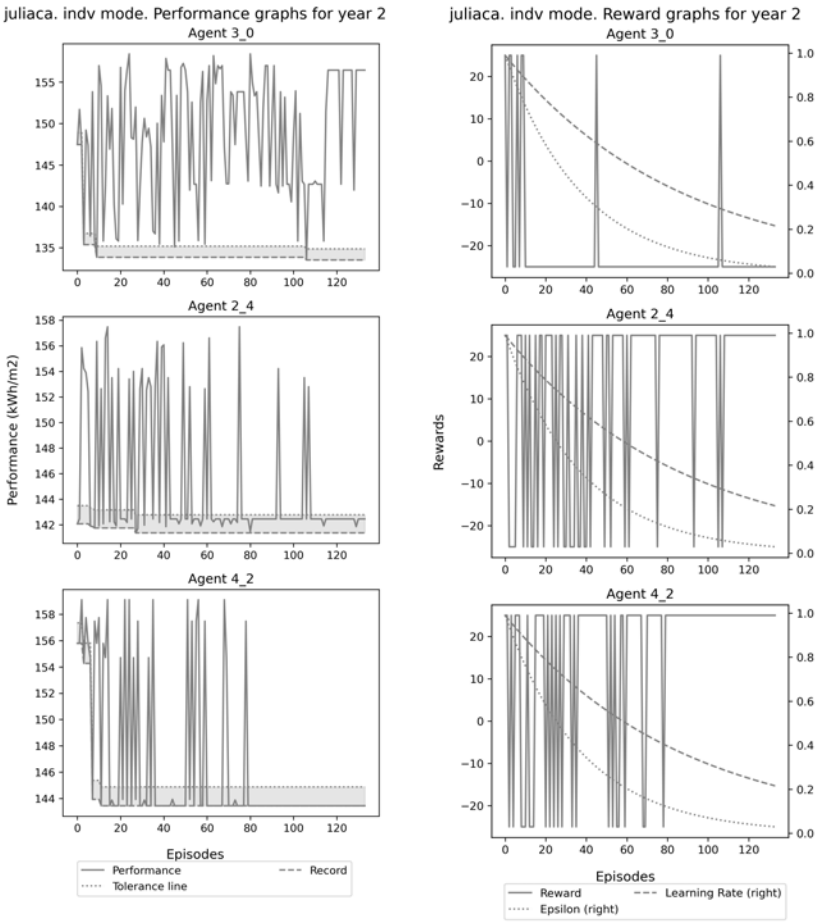
() Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (9)

() Performance

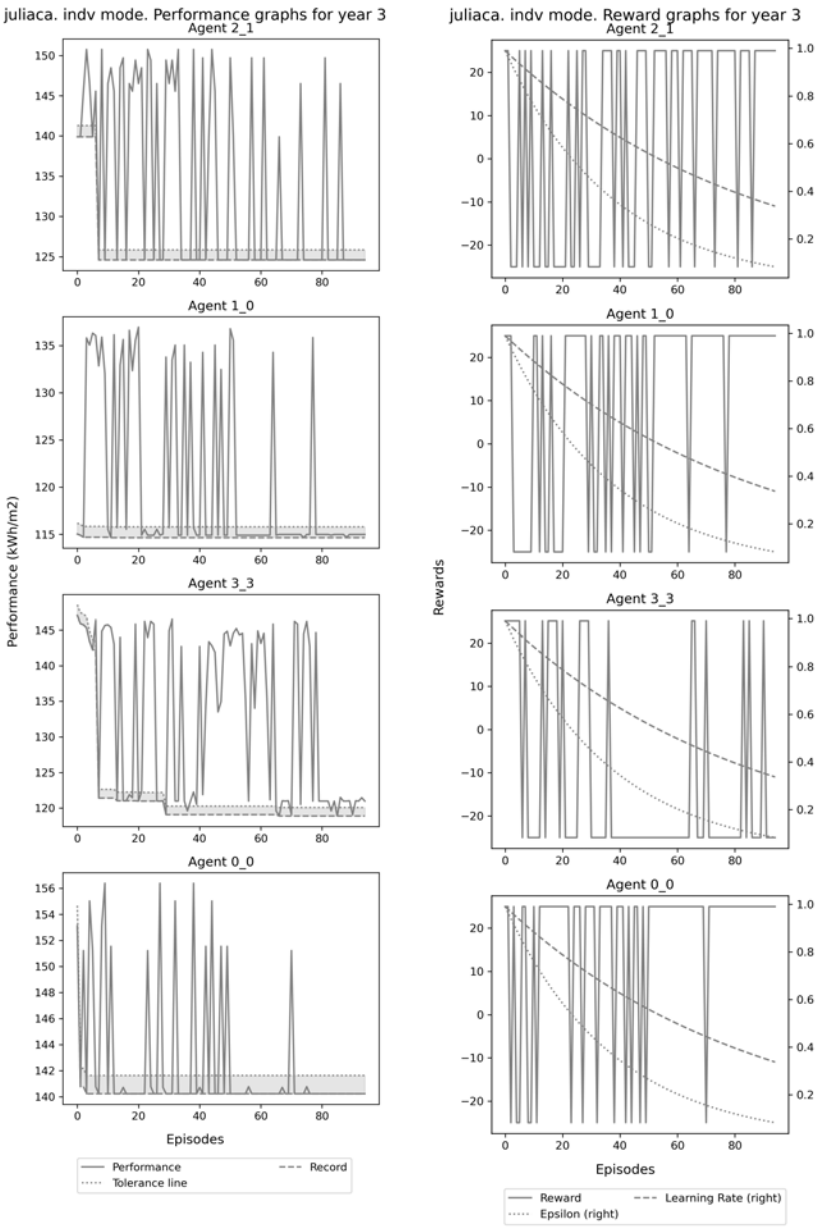
() Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (10)

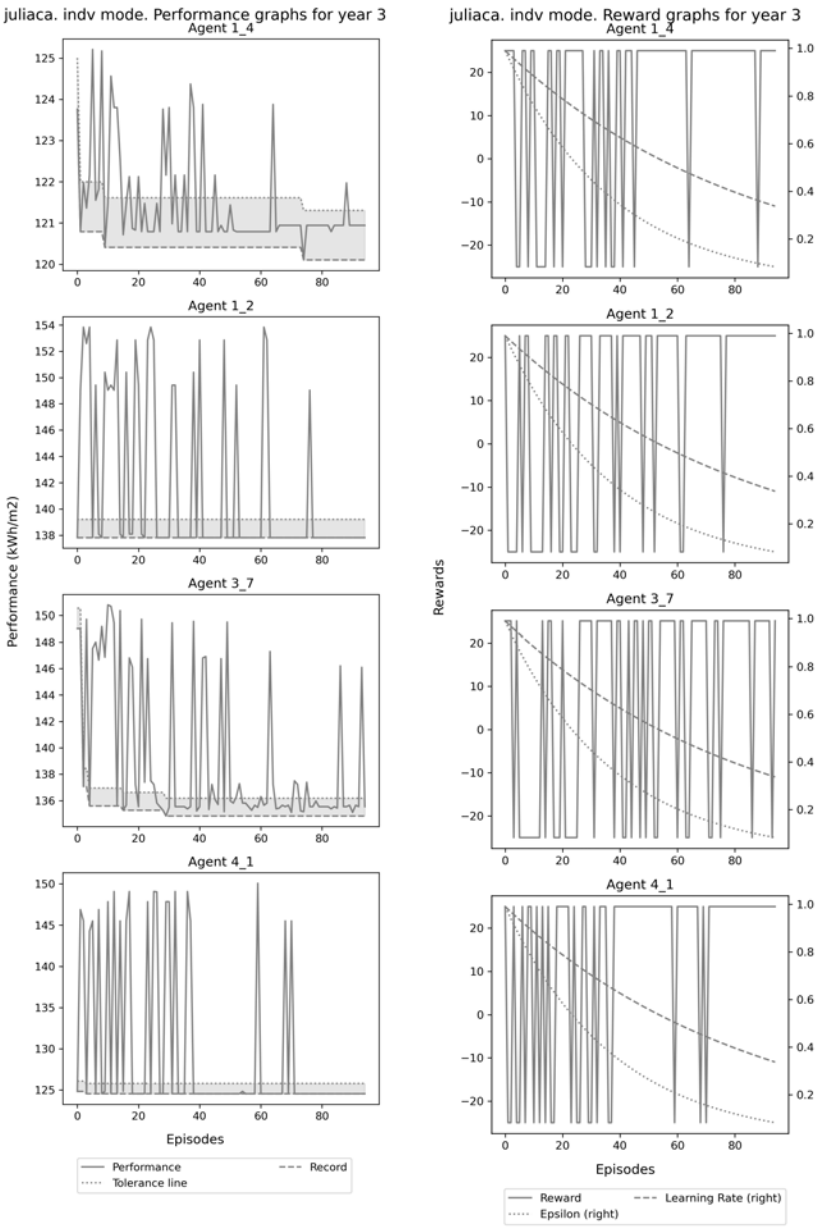
() Performance

() Rewards



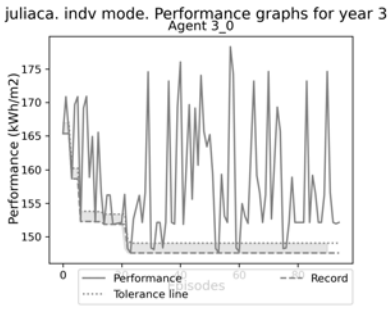
Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (11)

() Performance () Rewards

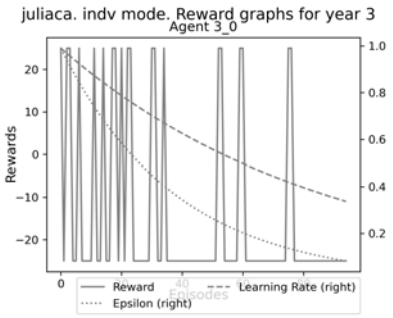


Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (12)

() Performance

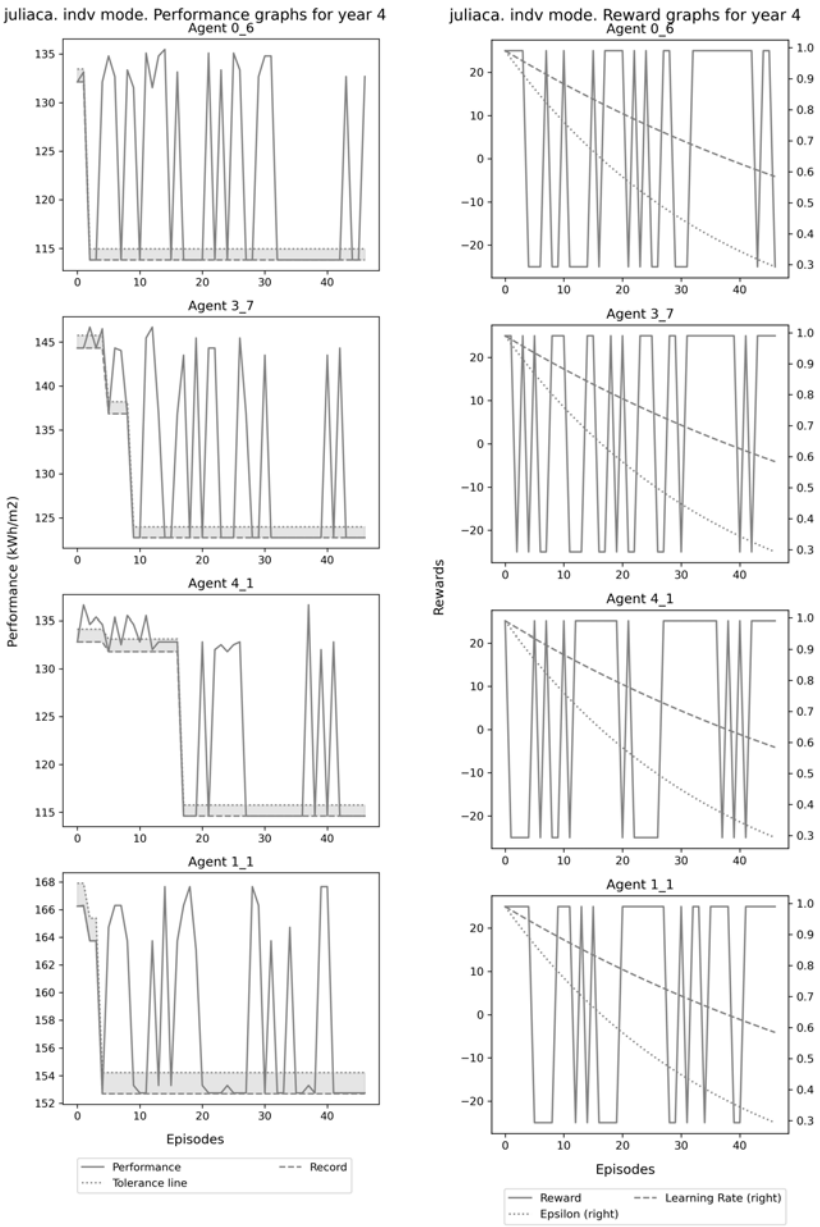


() Rewards



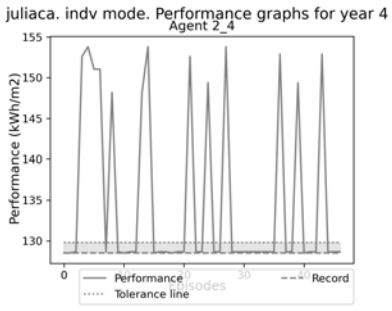
Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (13)

() Performance () Rewards

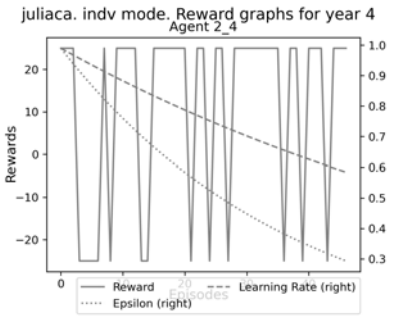


Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (14)

() Performance

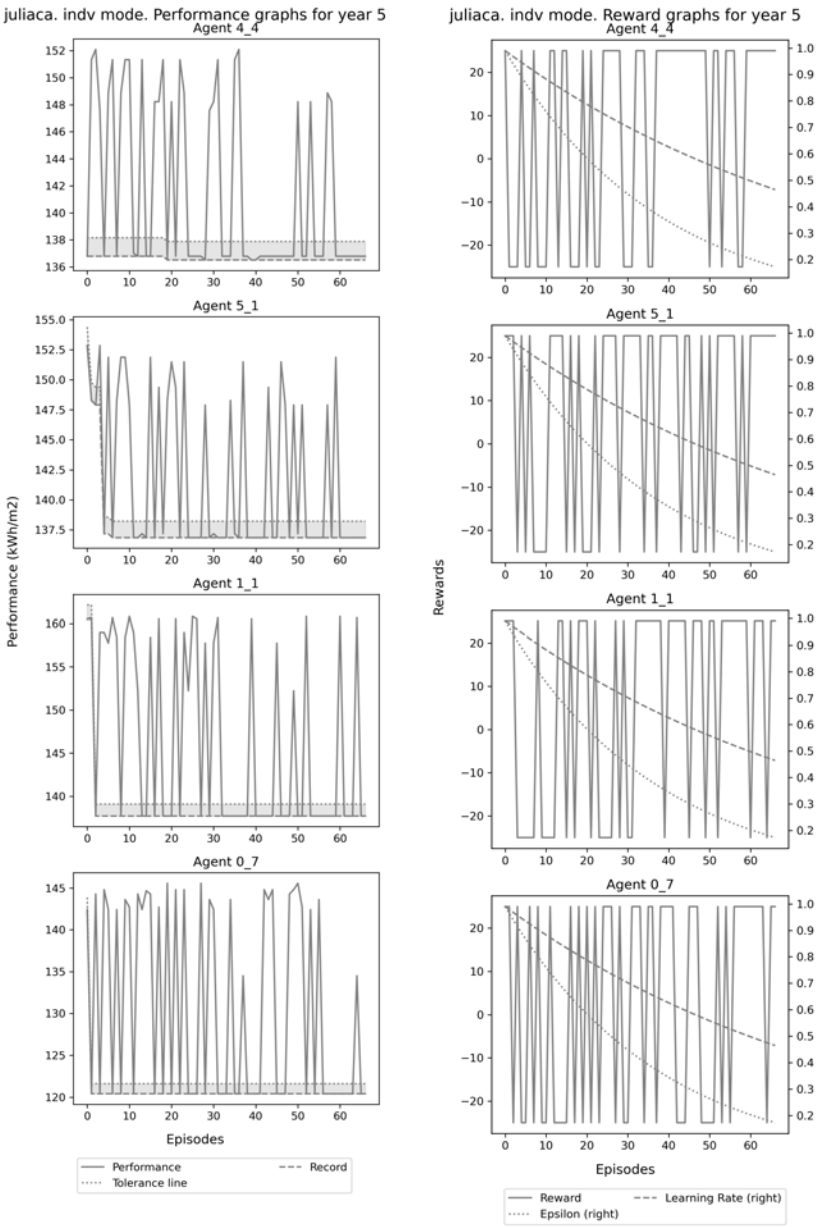


() Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (15)

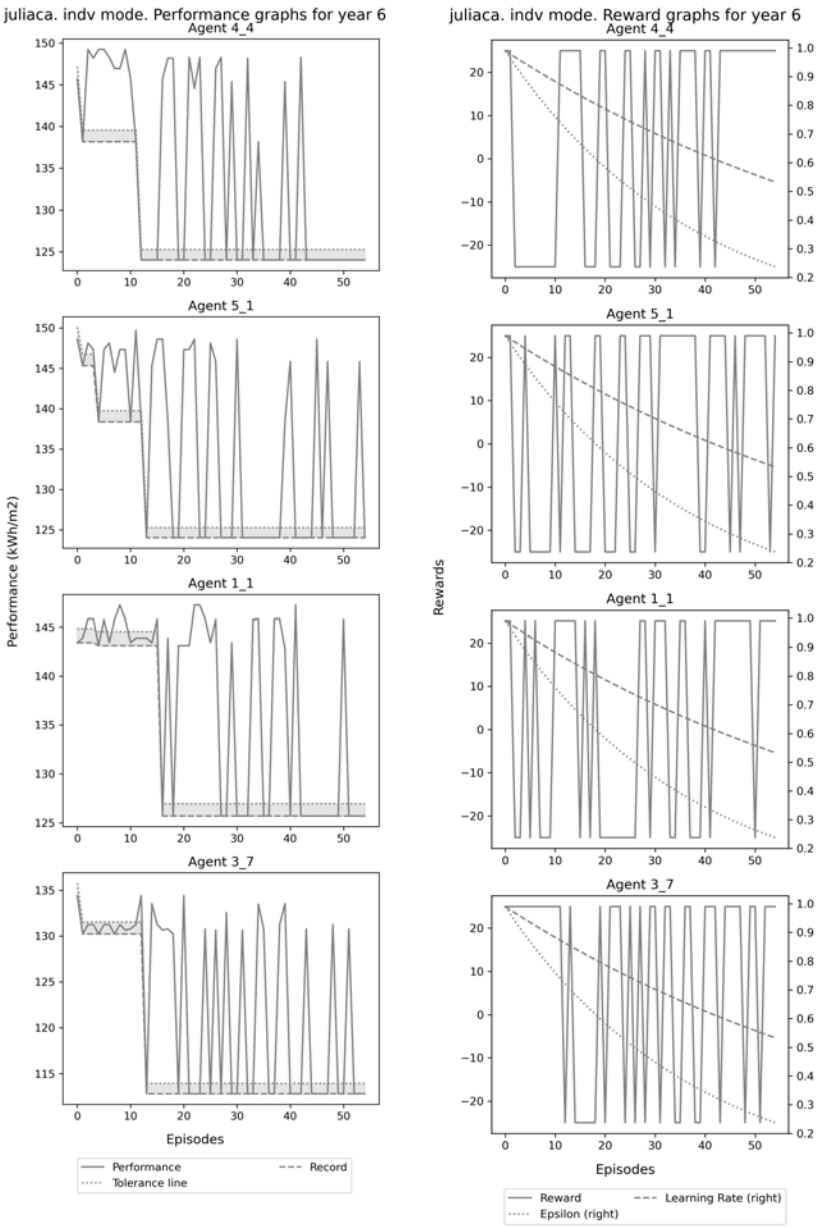
() Performance () Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (16)

() Performance

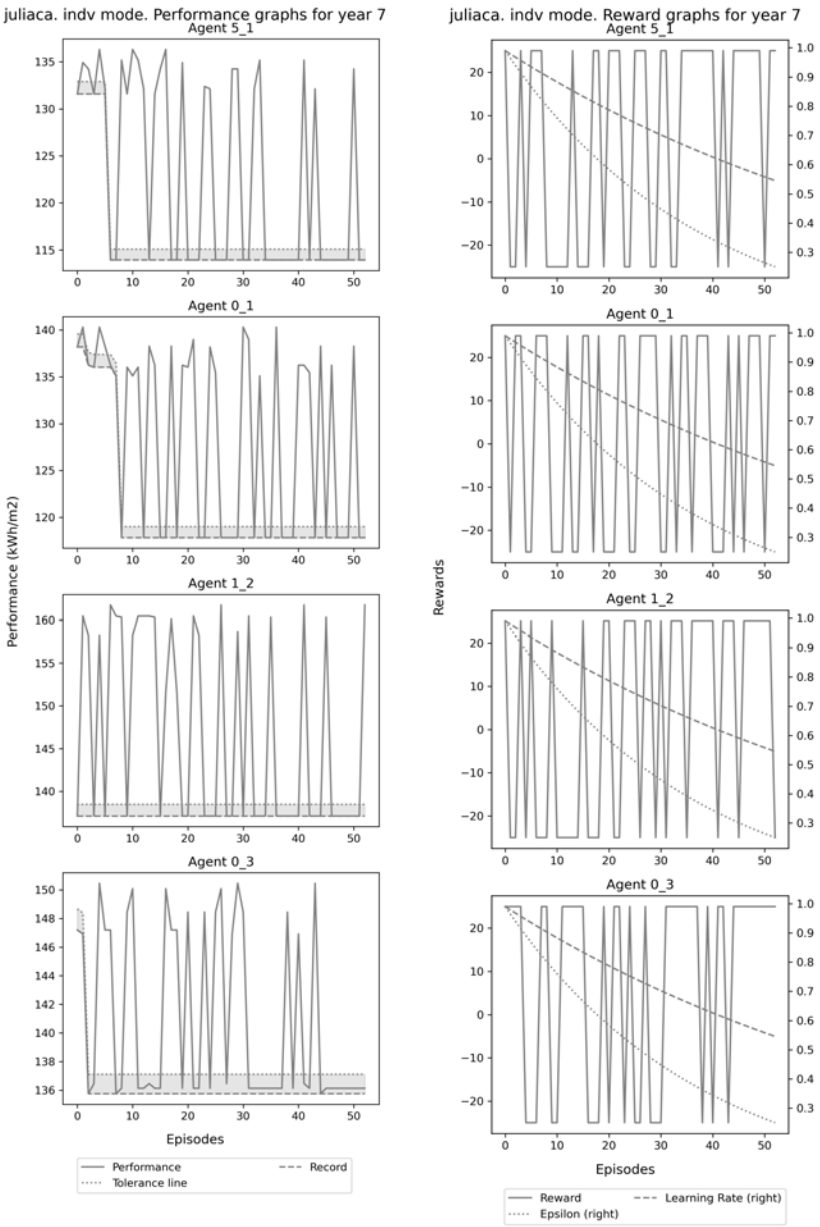
() Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (17)

() Performance

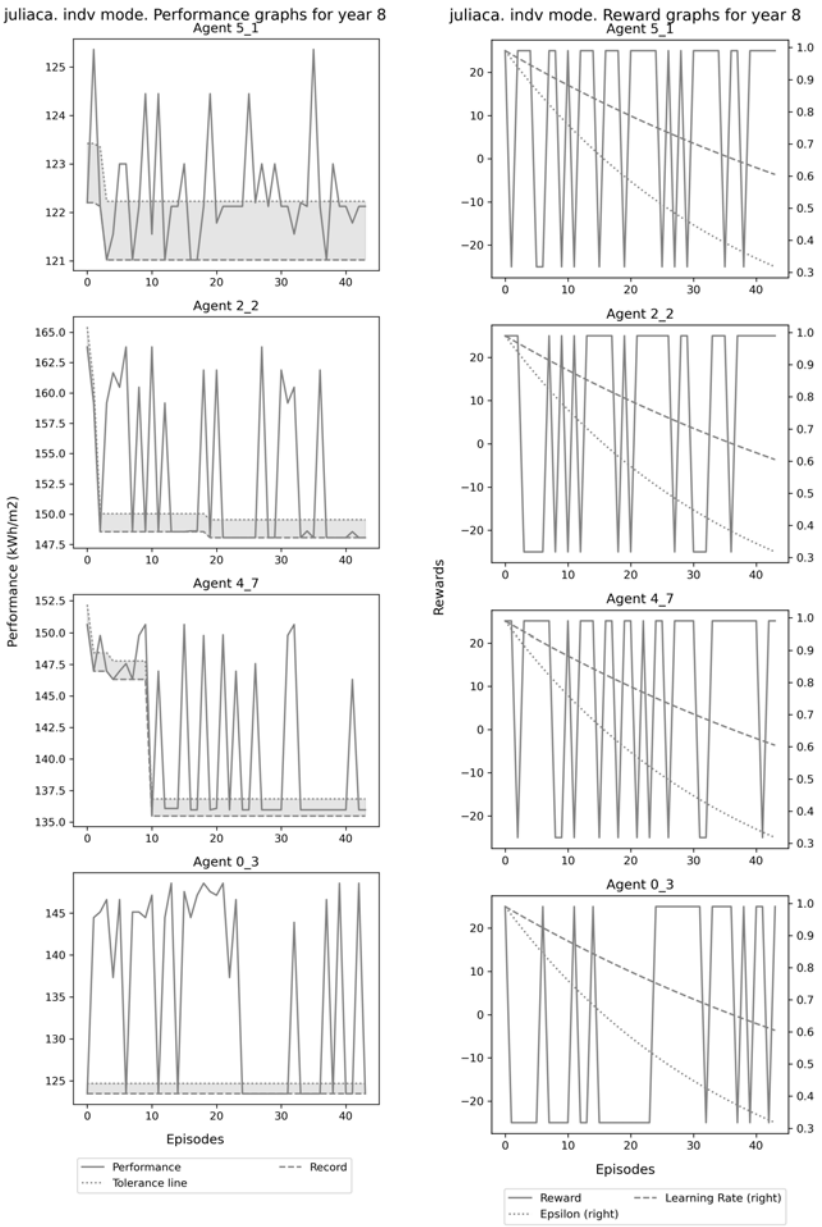
() Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (18)

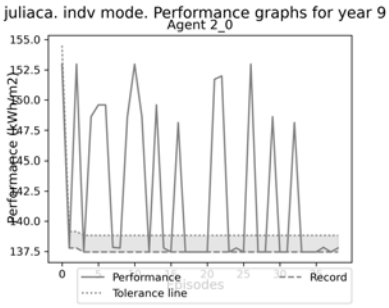
() Performance

() Rewards

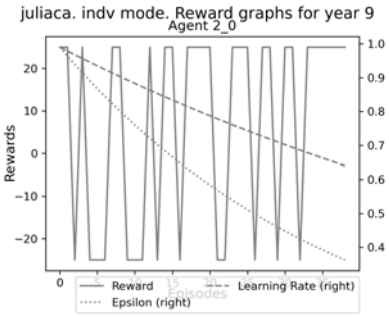


Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (19)

() Performance

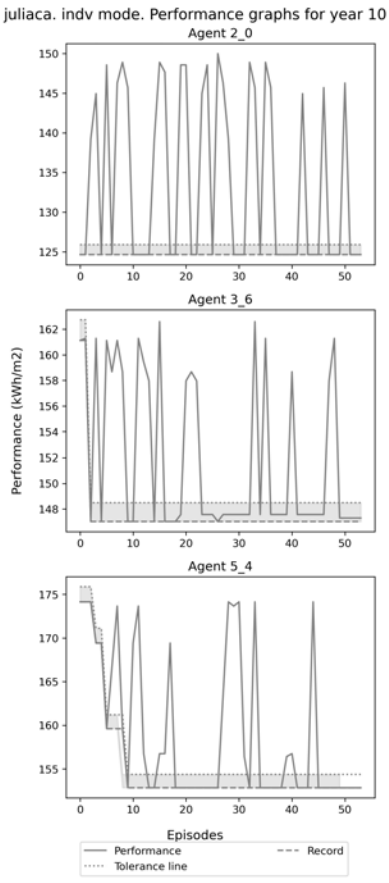


() Rewards

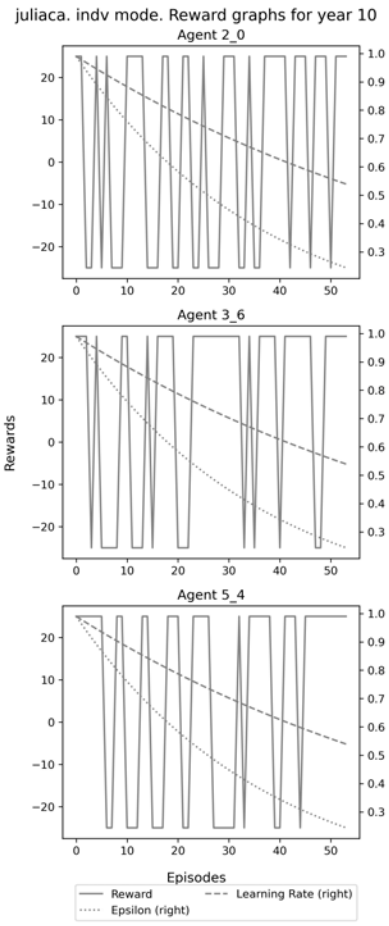


Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (20)

() Performance



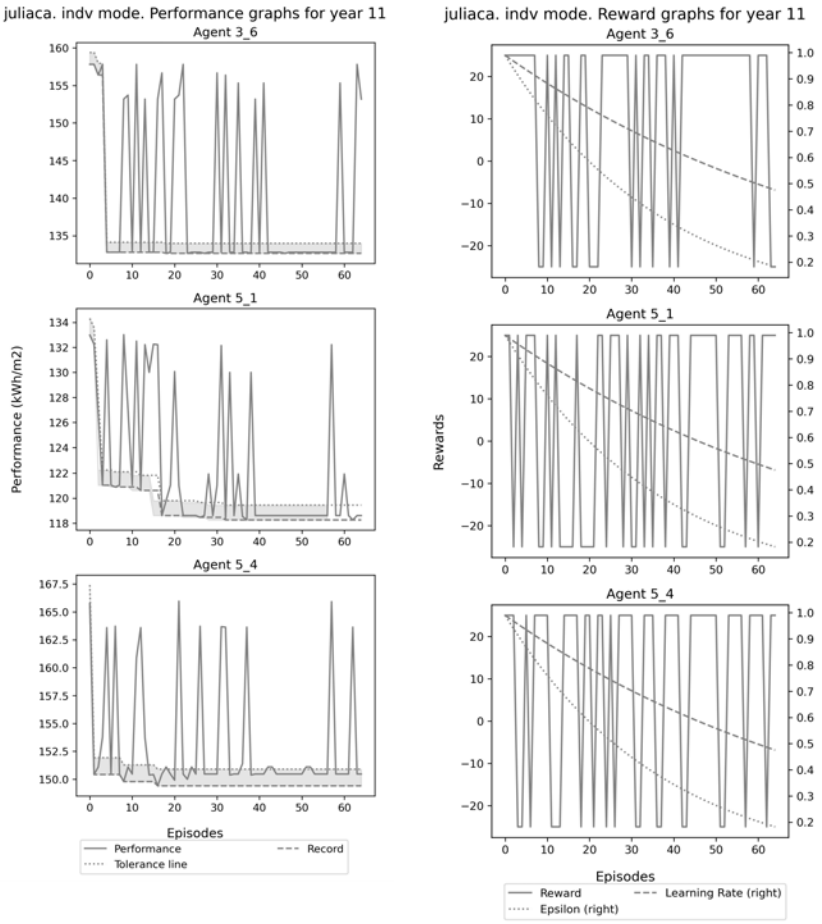
() Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (21)

() Performance

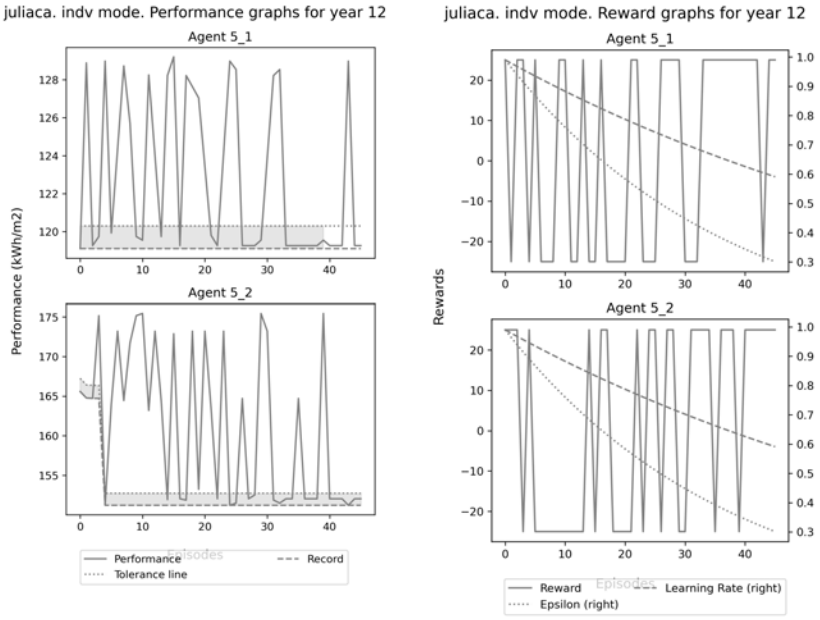
() Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (22)

() Performance

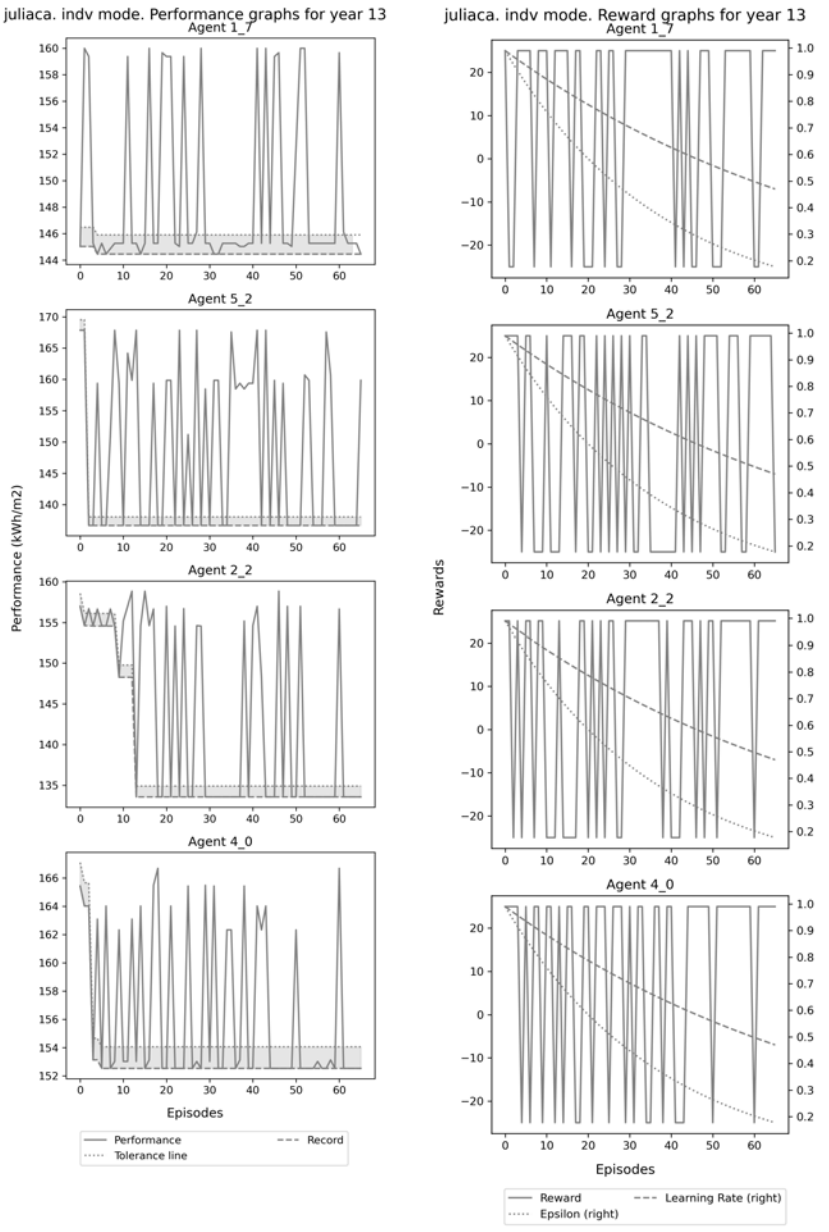
() Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (23)

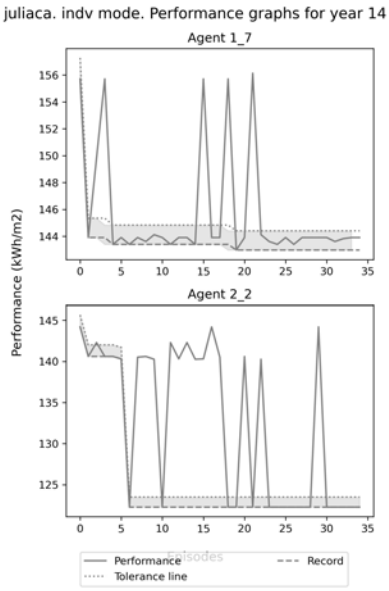
() Performance

() Rewards

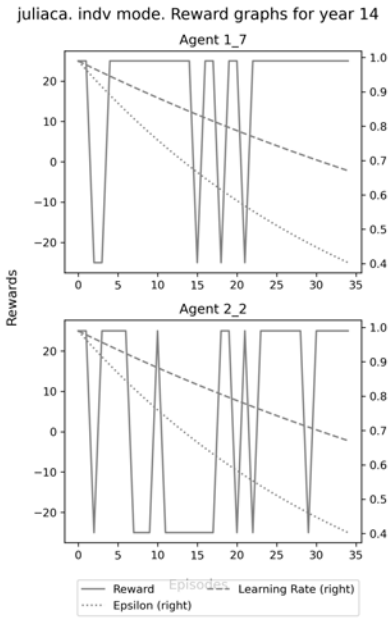


Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (24)

() Performance

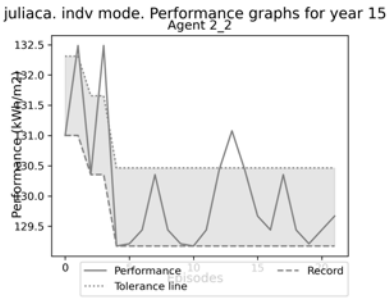


() Rewards

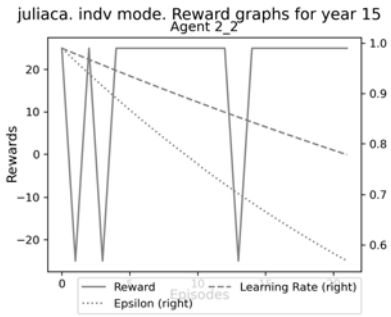


Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (25)

() Performance

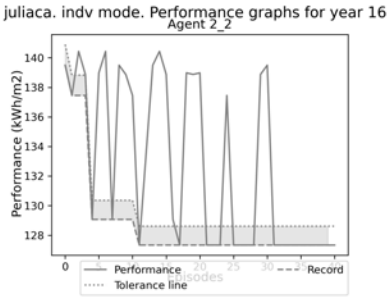


() Rewards

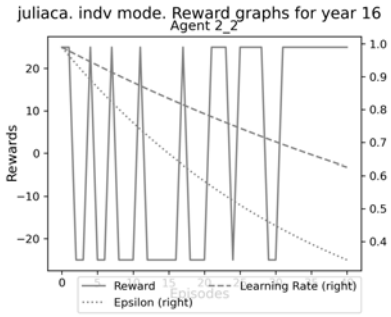


Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (26)

() Performance

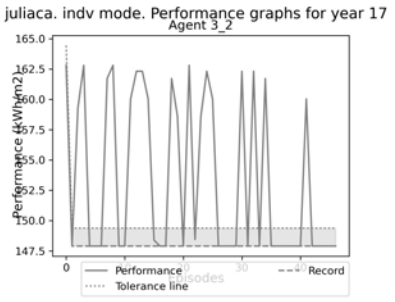


() Rewards

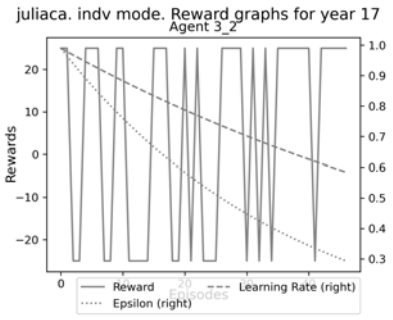


Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (27)

() Performance

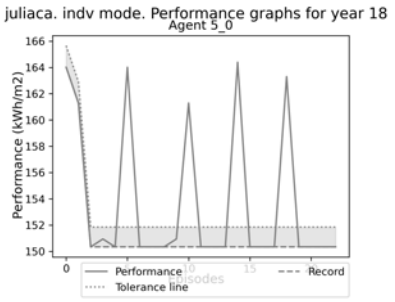


() Rewards

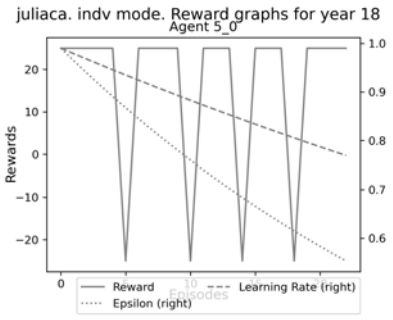


Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (28)

() Performance

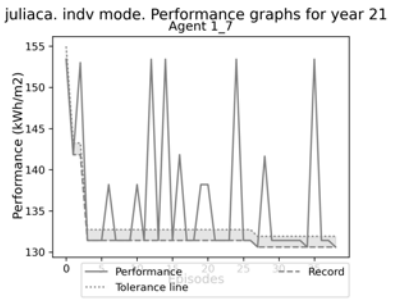


() Rewards

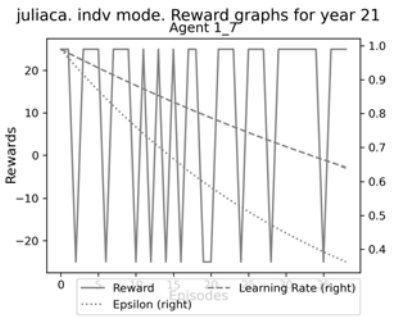


Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (29)

() Performance

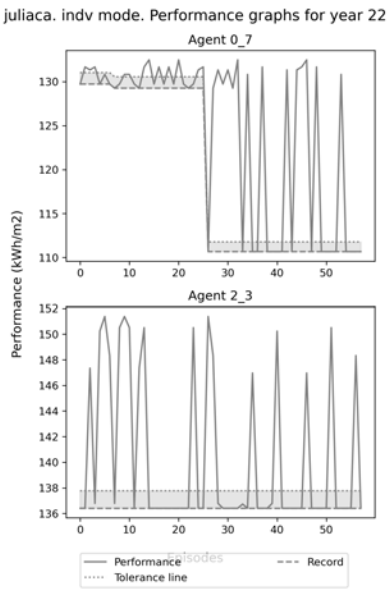


() Rewards

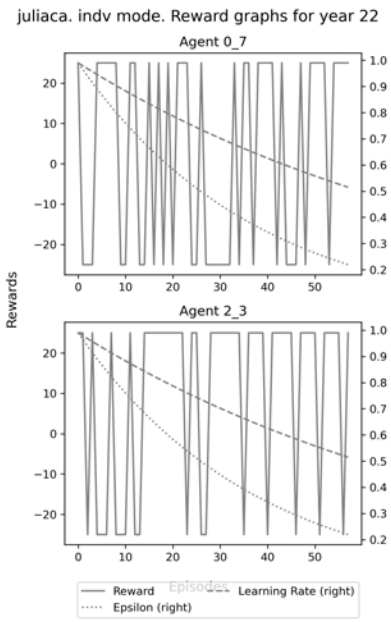


Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (30)

() Performance

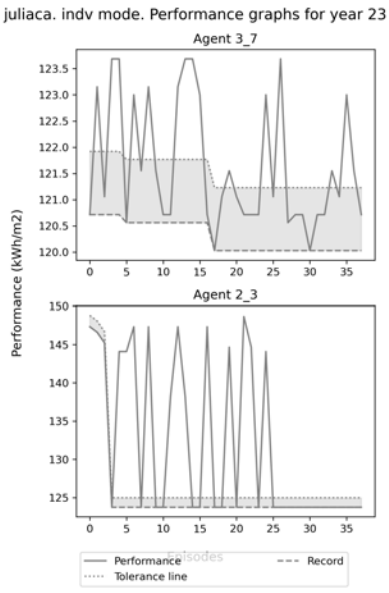


() Rewards

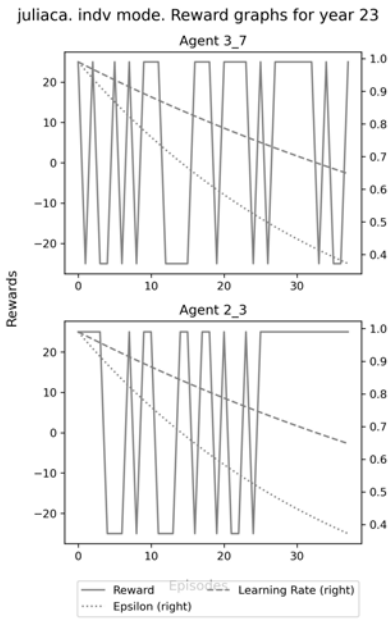


Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (31)

() Performance



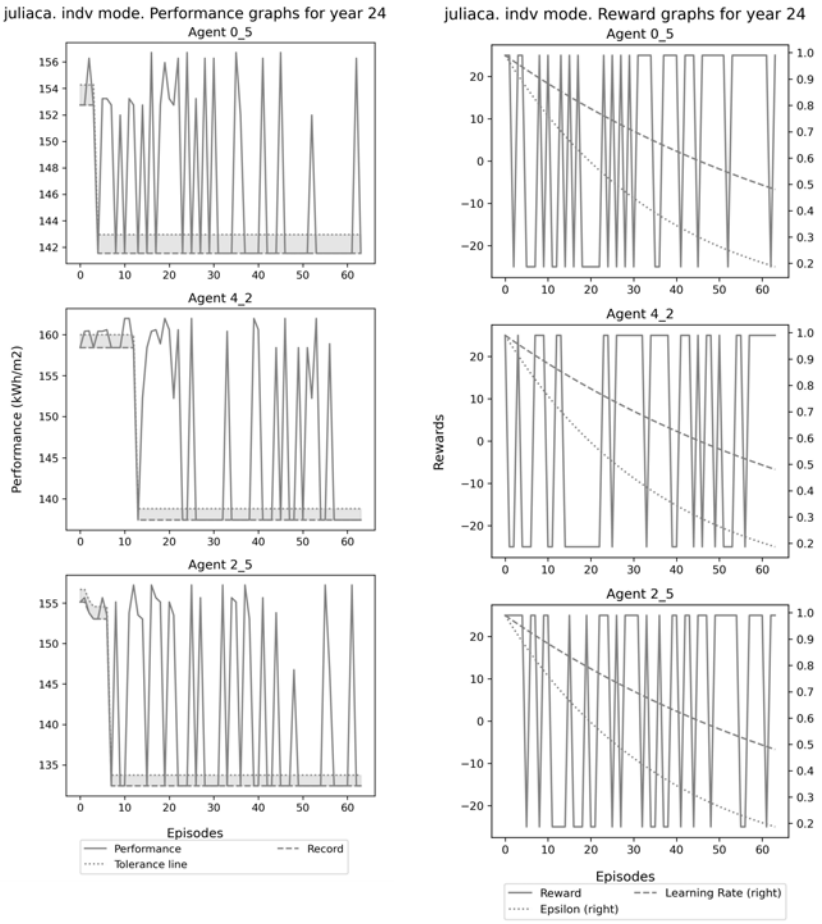
() Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (32)

() Performance

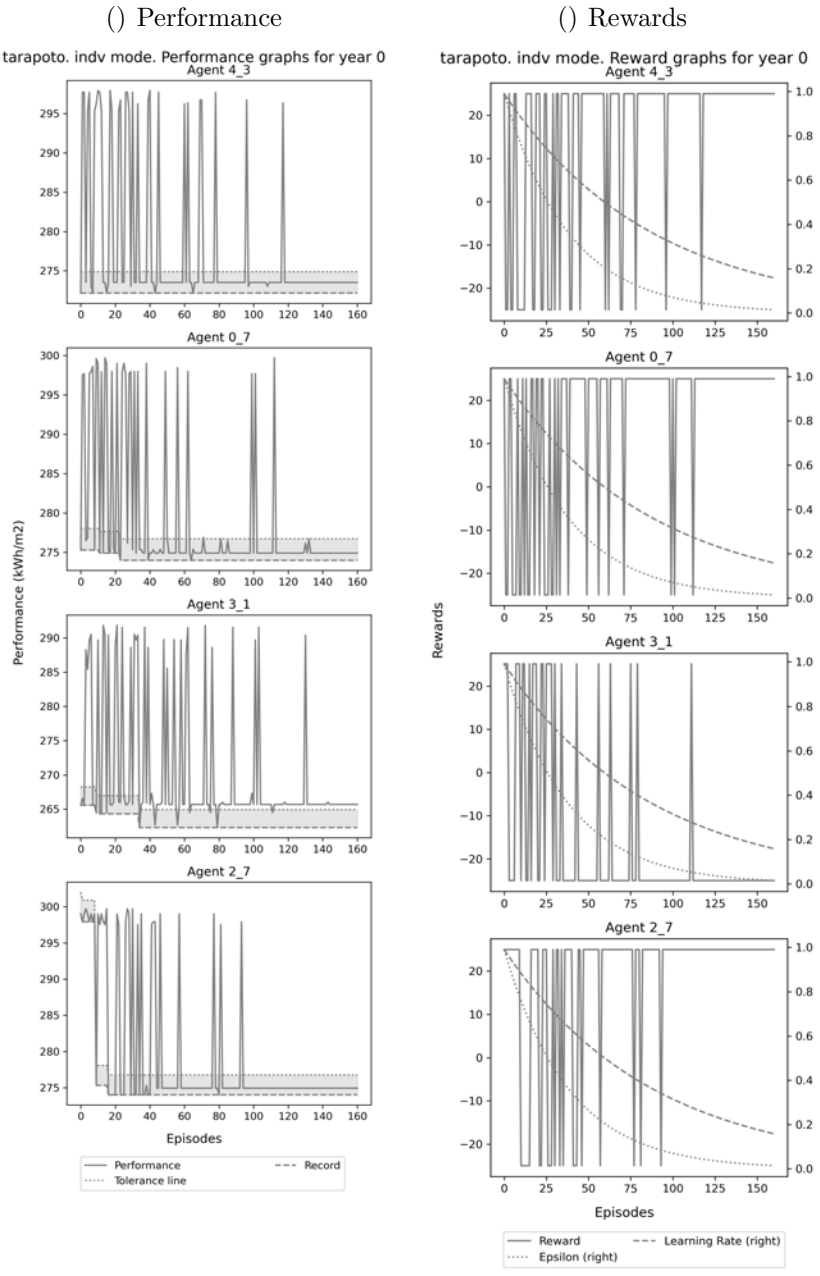
() Rewards



Learning process of all agents in Juliaca competitive scenario for energy use Minimisation (33)

Appendix 20

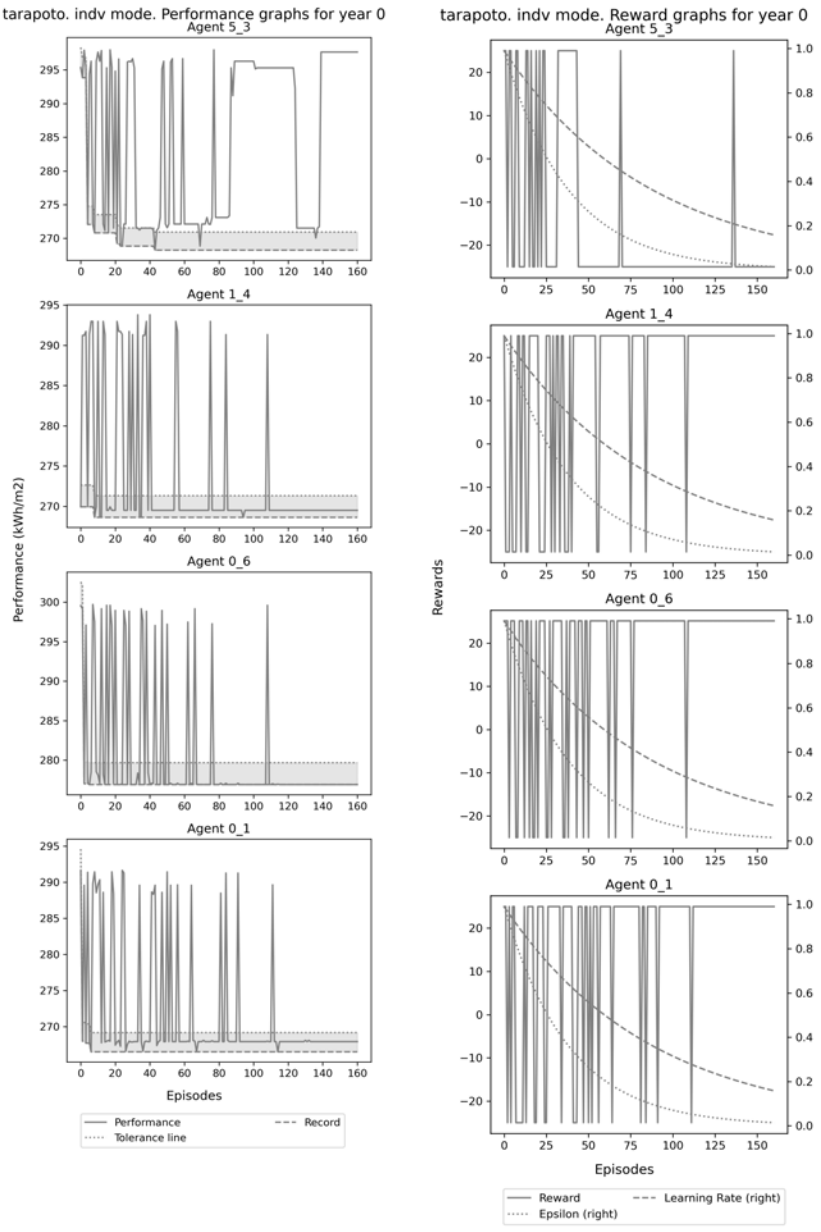
Learning process by agent for each simulation year (Tarapoto, Minimisation, competitive)



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (1)

() Performance

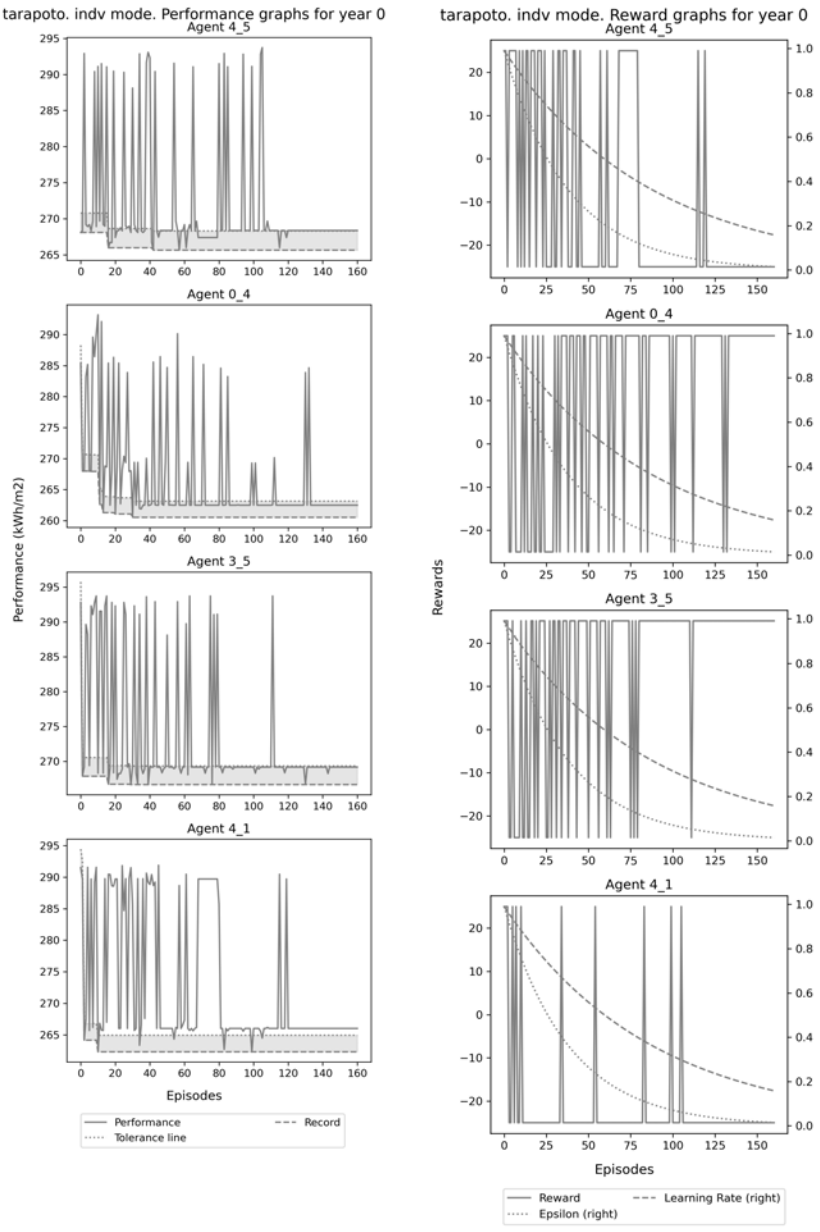
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (2)

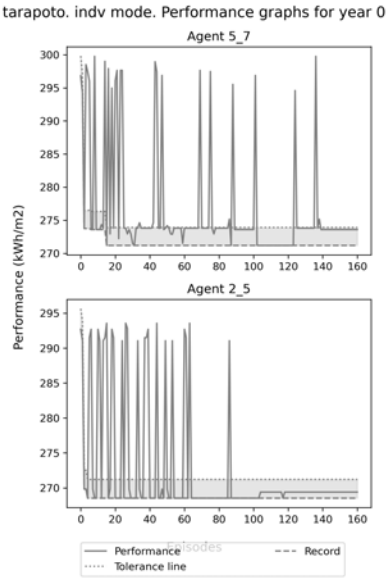
() Performance

() Rewards

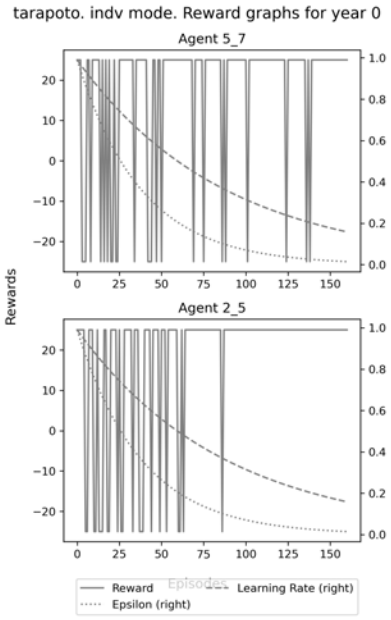


Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (3)

() Performance



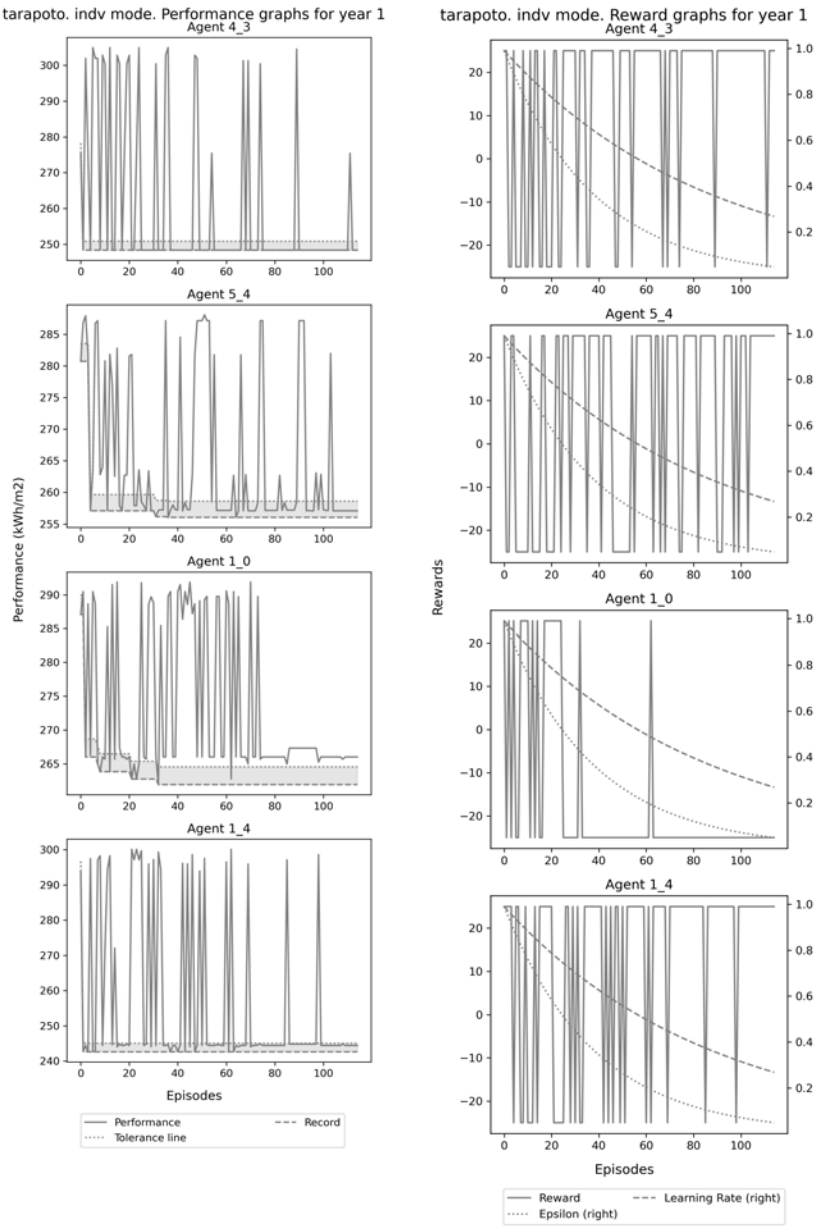
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (4)

() Performance

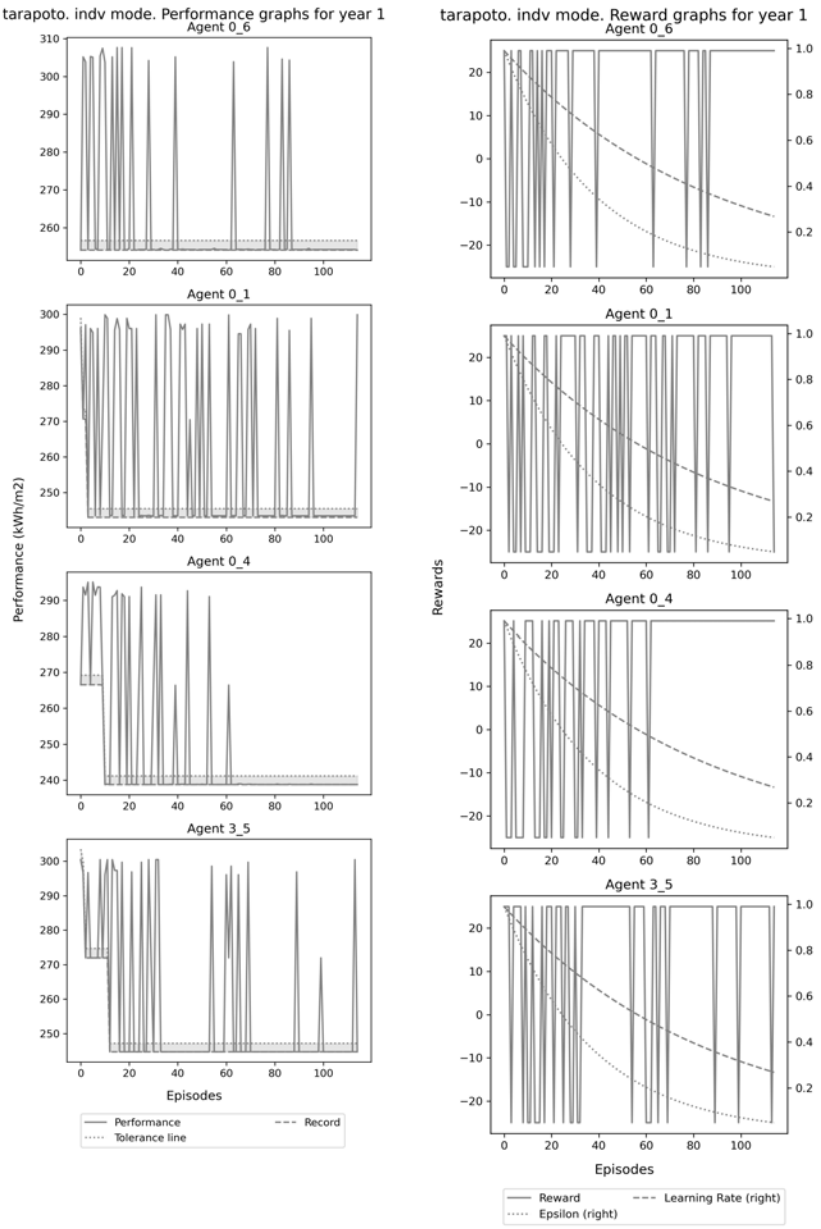
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (5)

() Performance

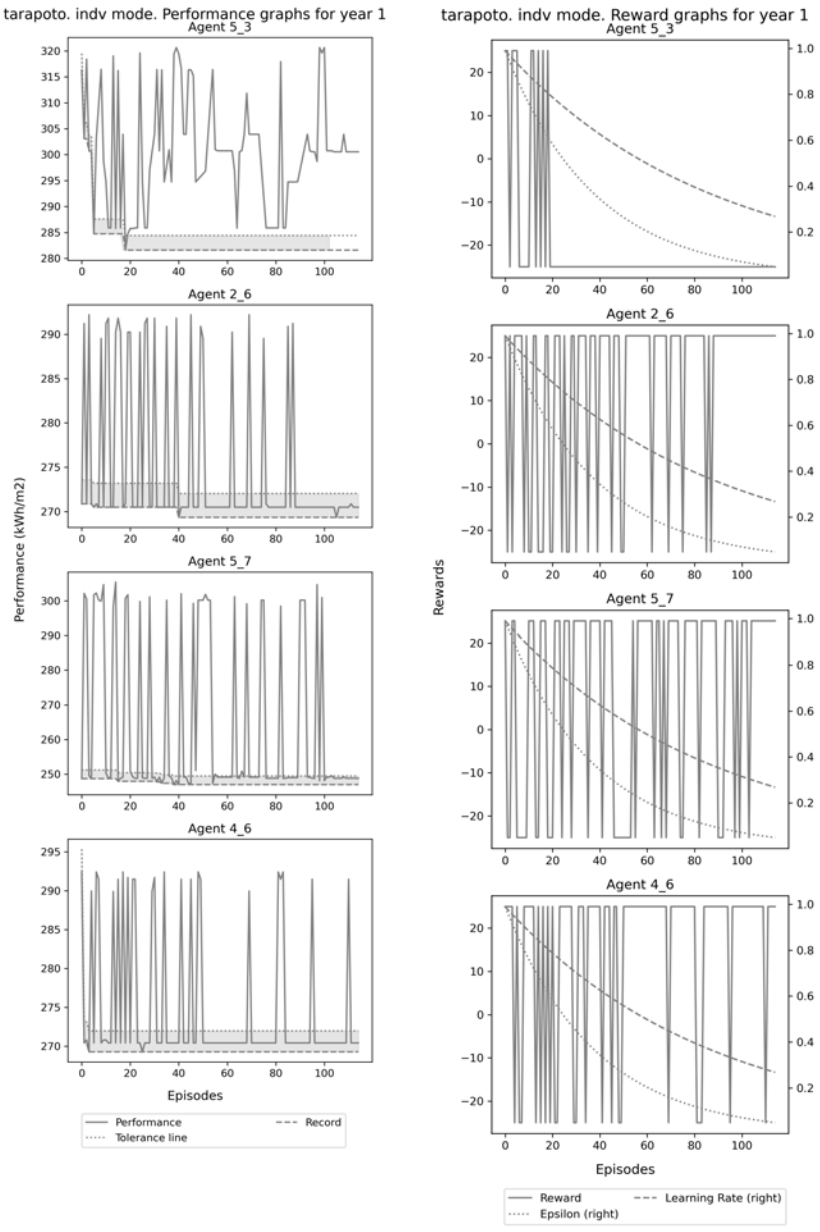
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (6)

() Performance

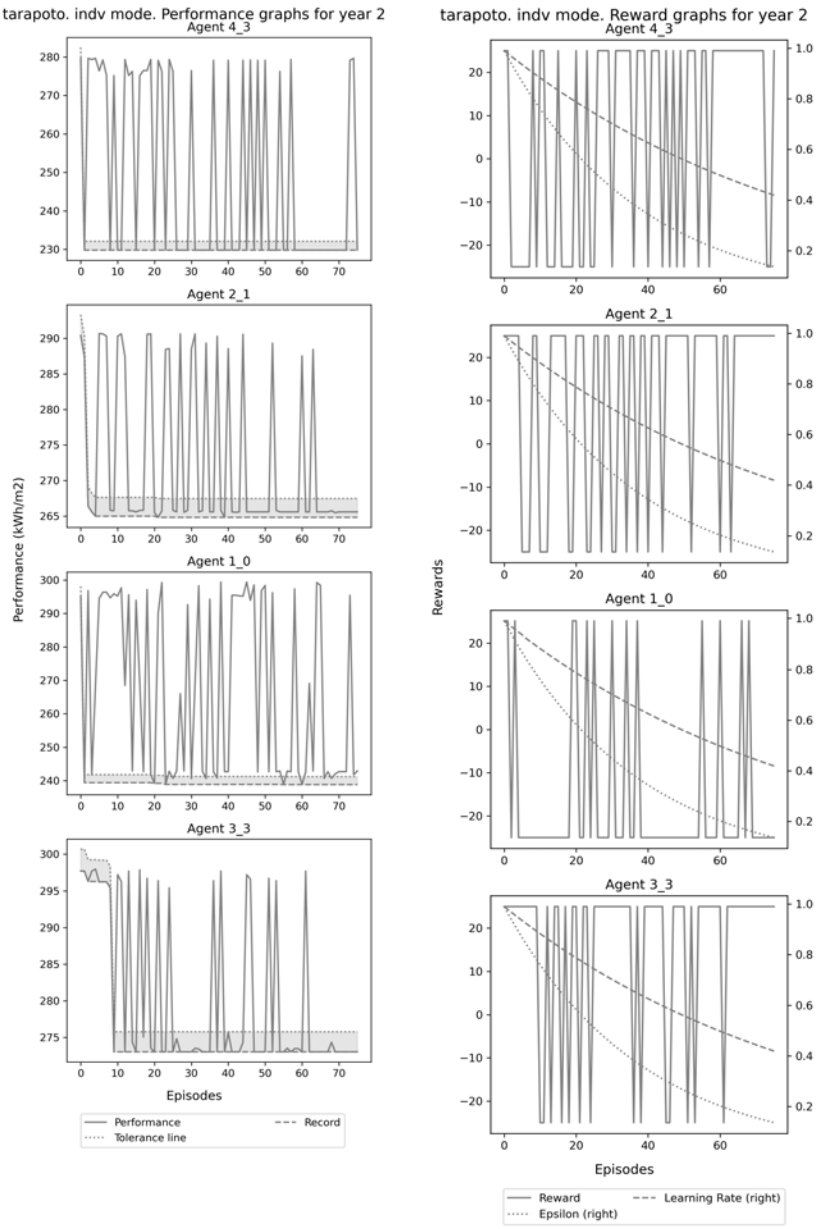
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (7)

() Performance

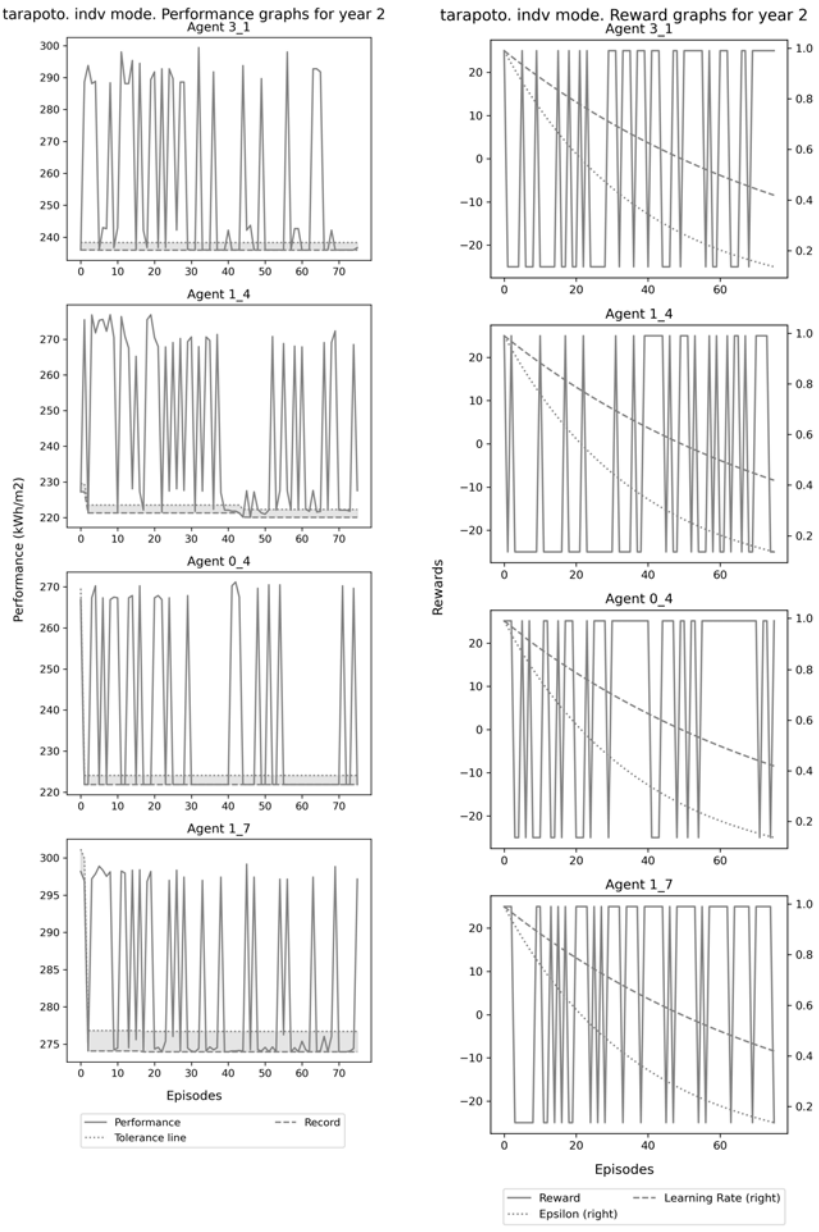
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (8)

() Performance

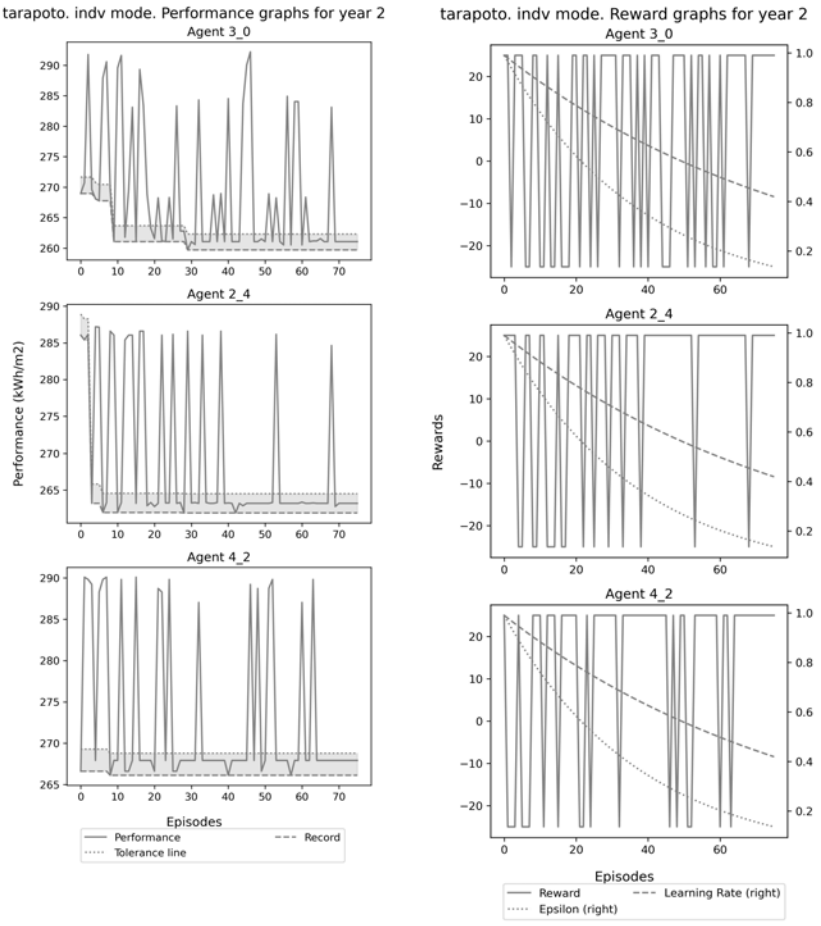
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (9)

() Performance

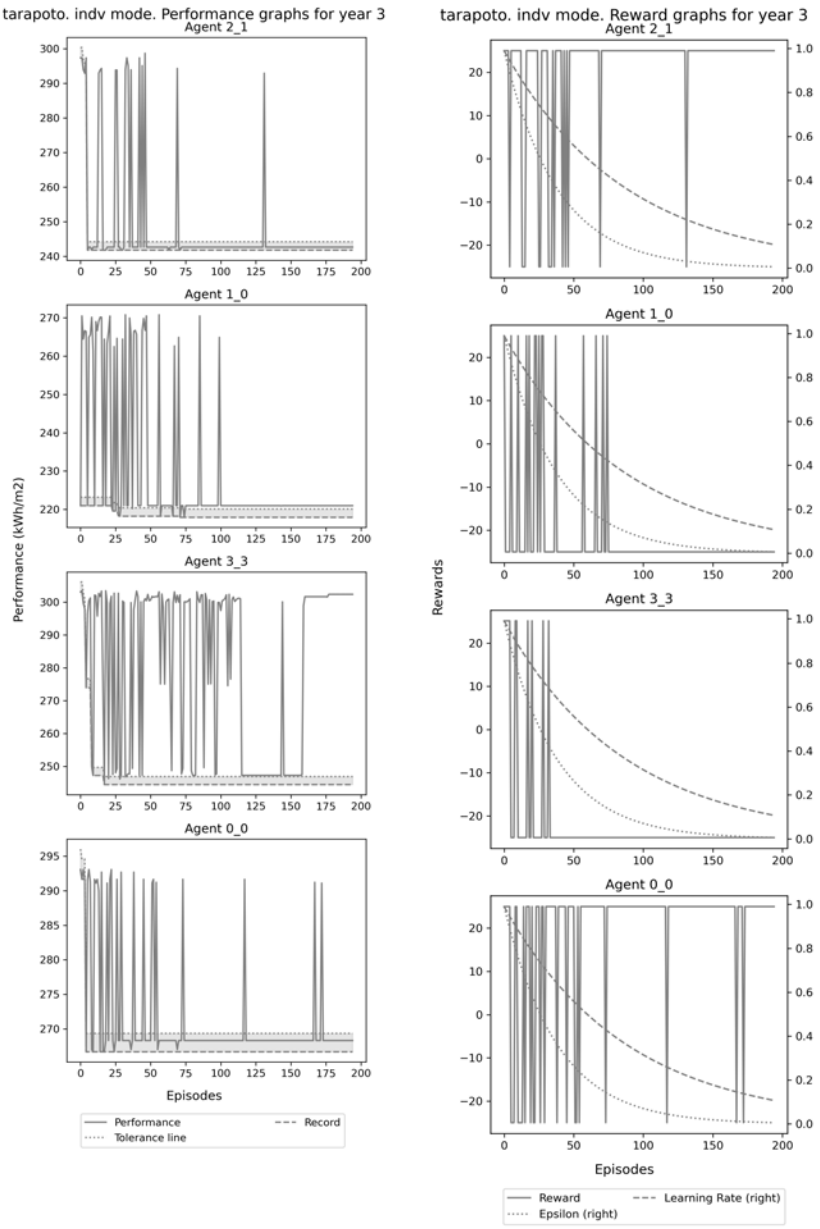
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (10)

() Performance

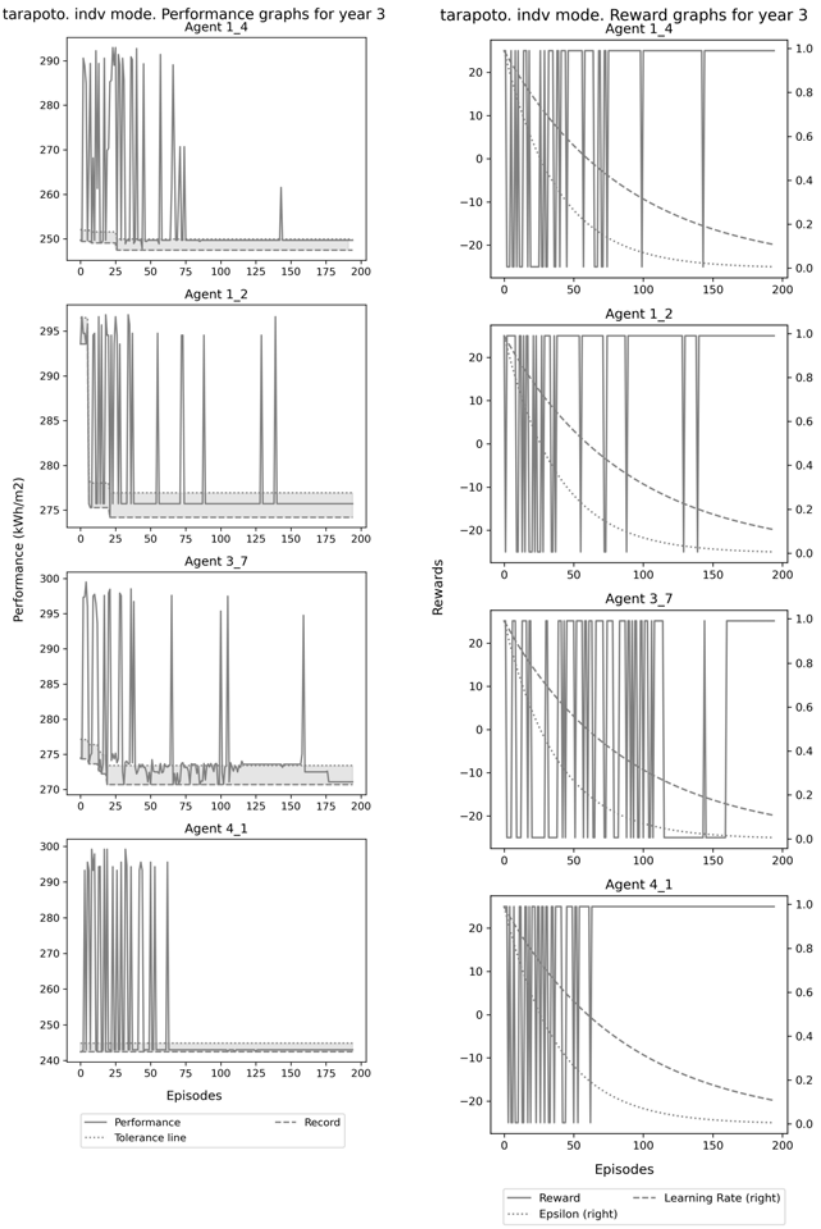
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (11)

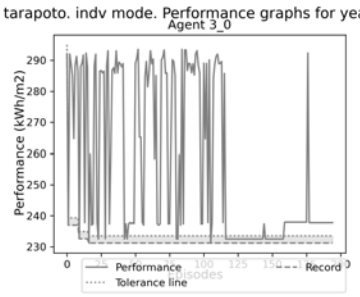
() Performance

() Rewards

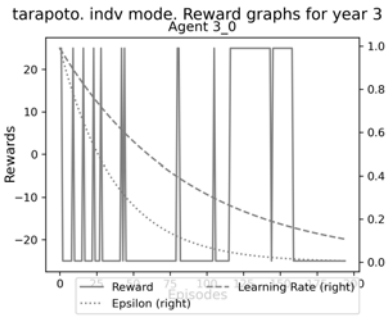


Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (12)

() Performance



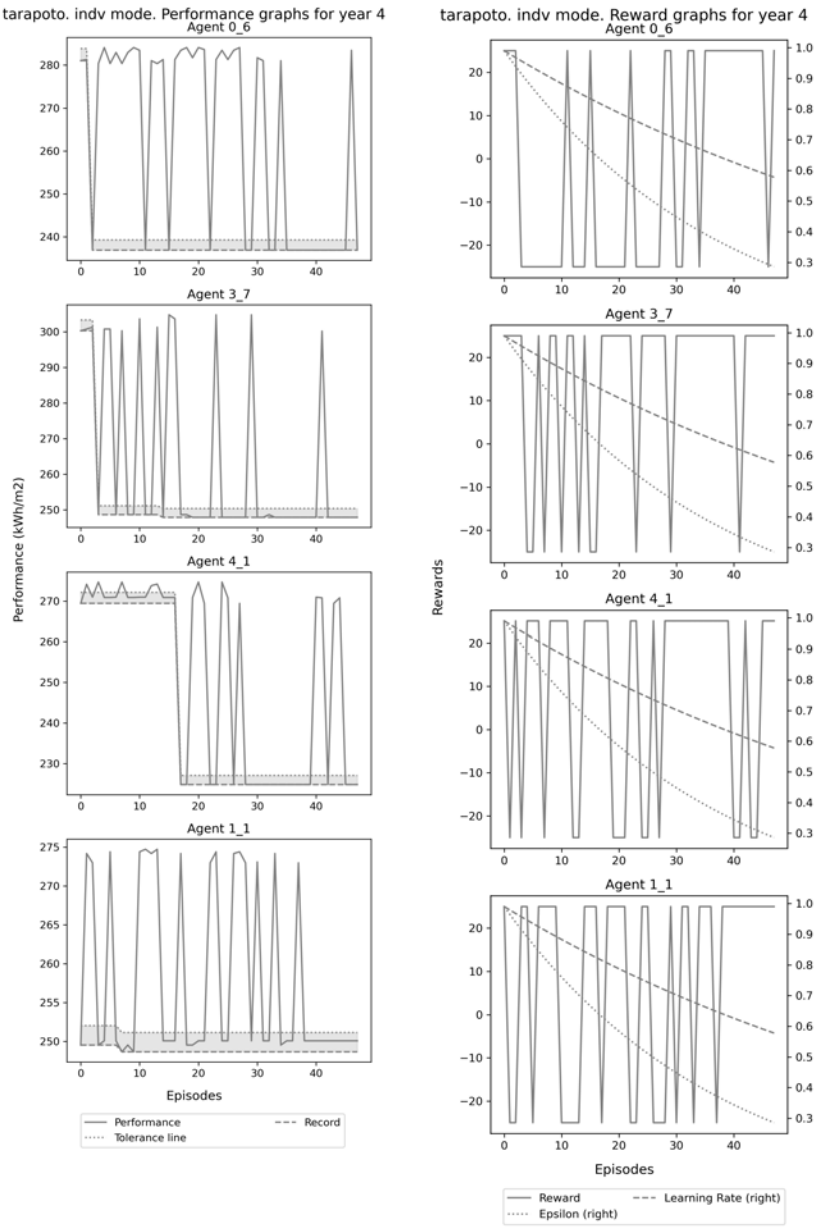
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (13)

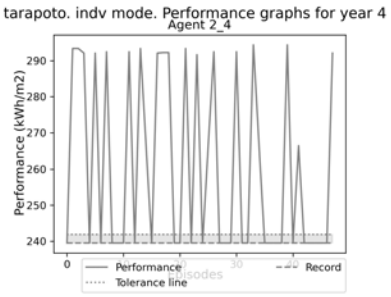
() Performance

() Rewards

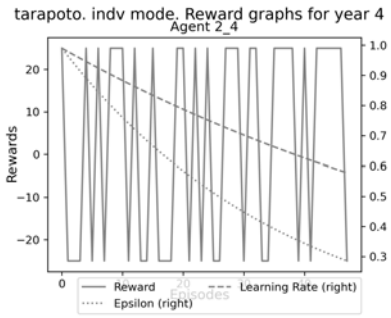


Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (14)

() Performance



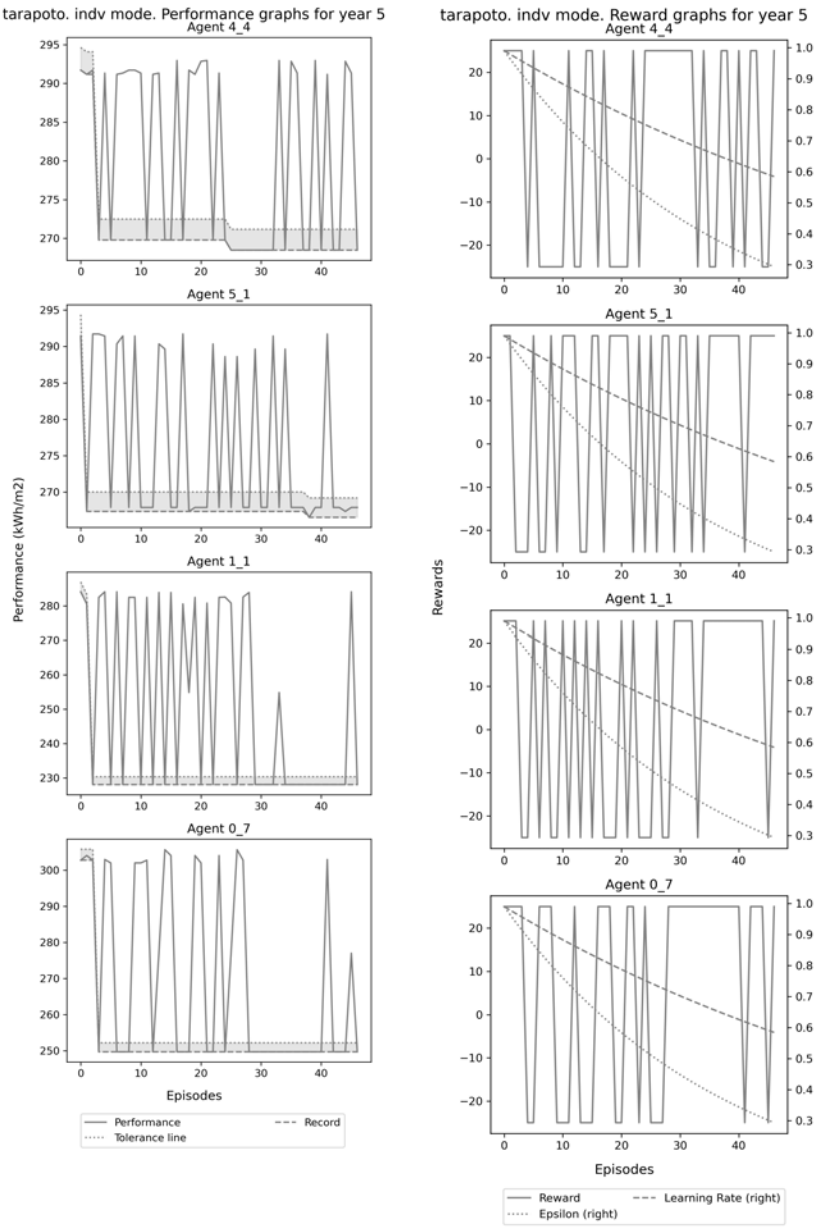
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (15)

() Performance

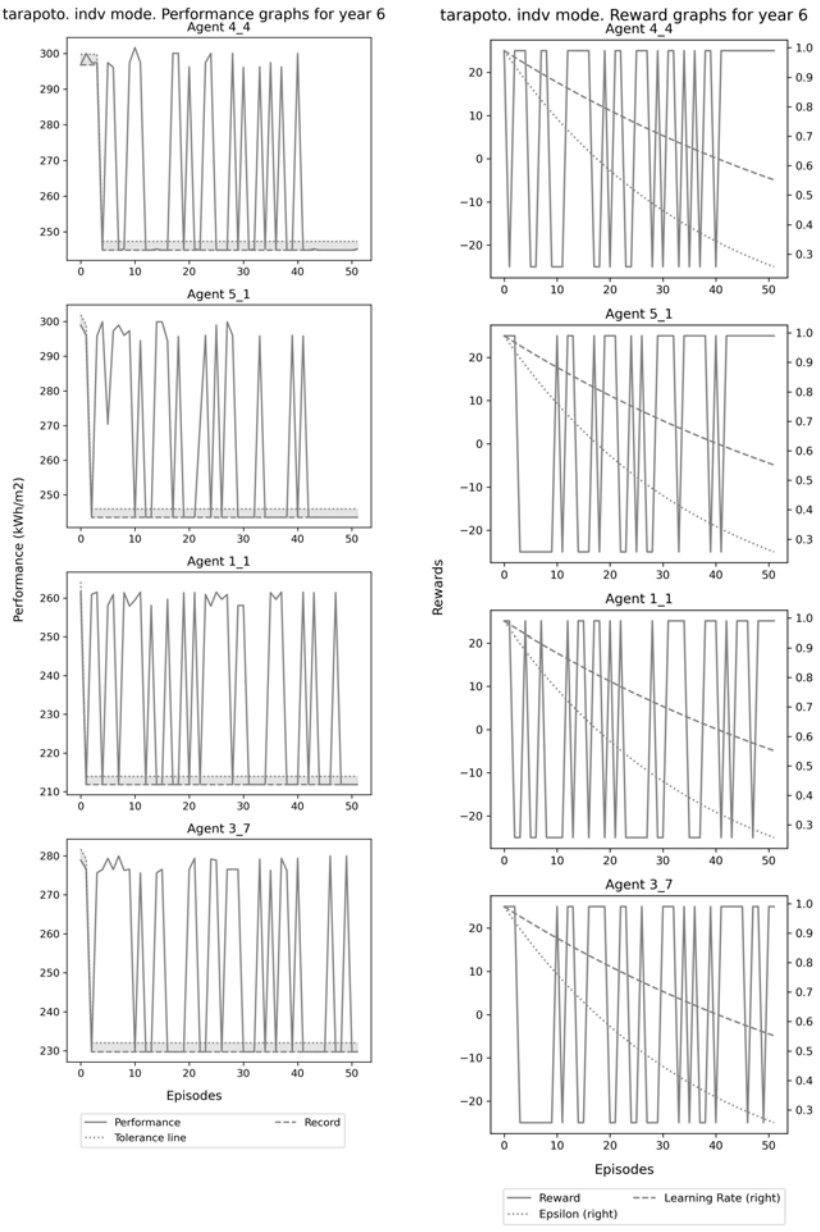
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (16)

() Performance

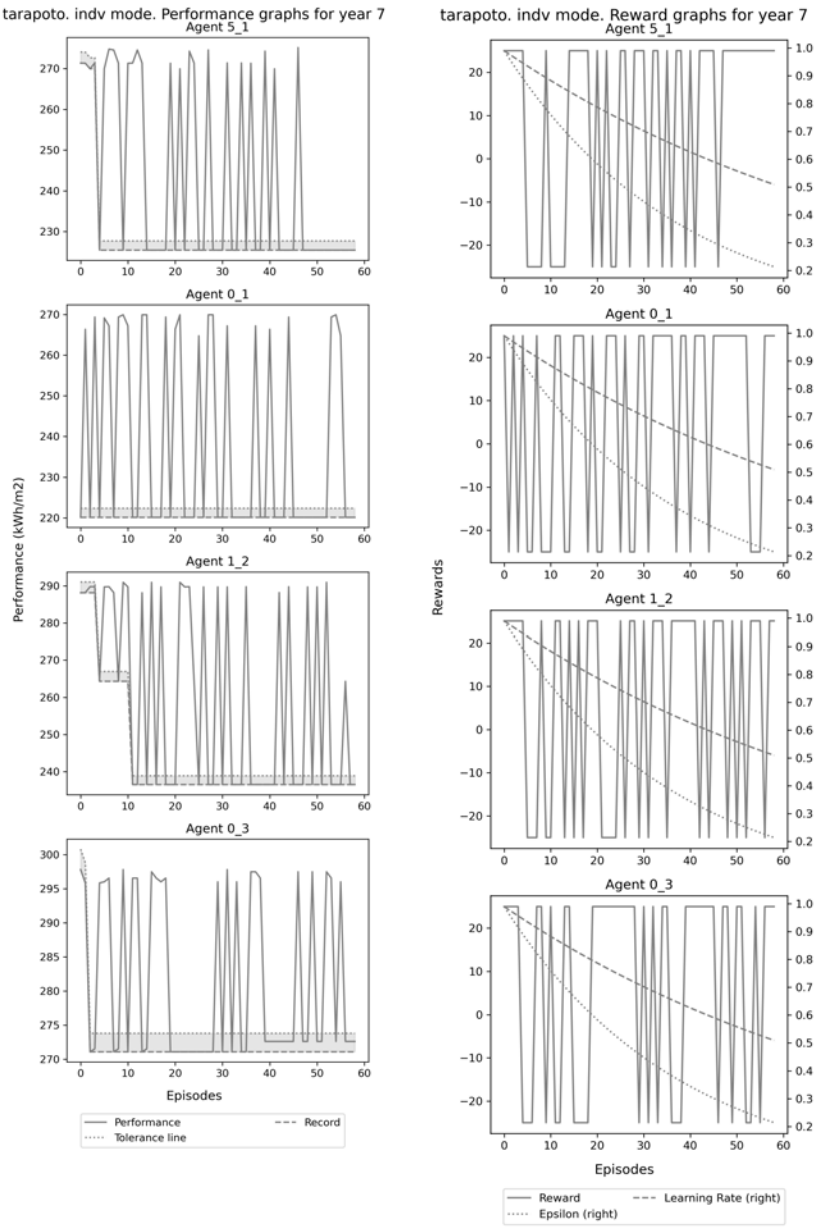
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (17)

() Performance

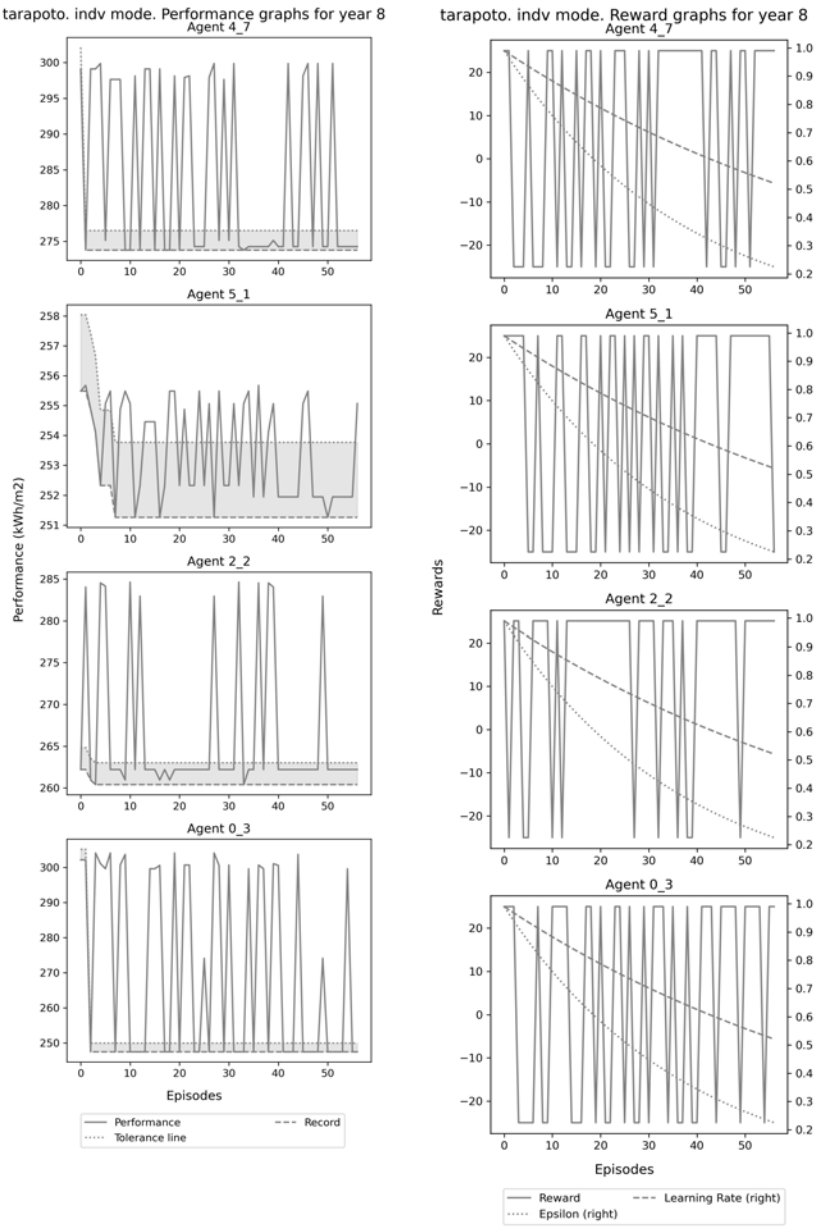
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (18)

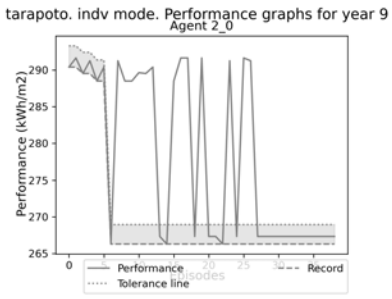
() Performance

() Rewards

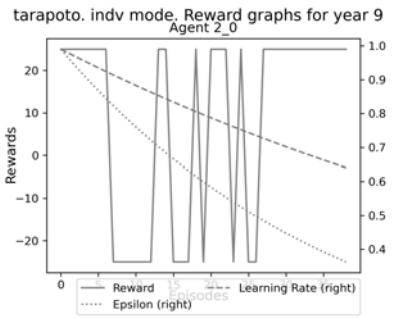


Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (19)

() Performance

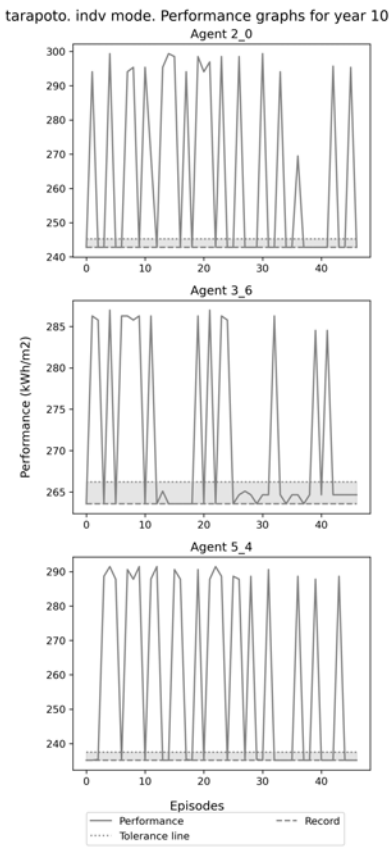


() Rewards

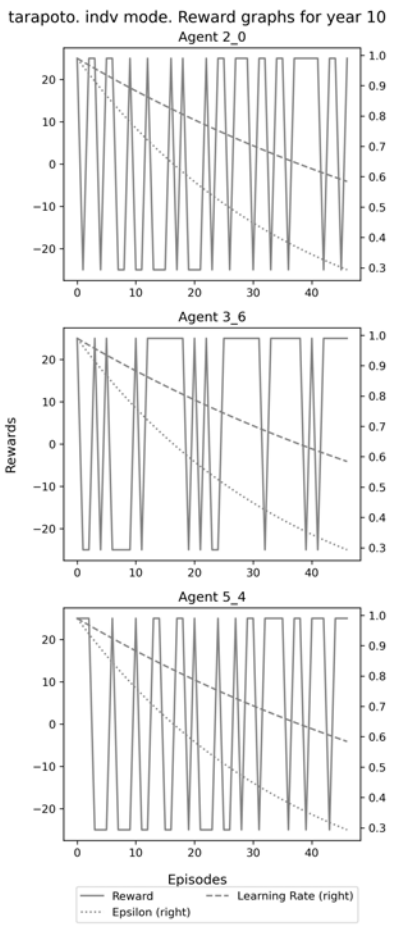


Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (20)

() Performance



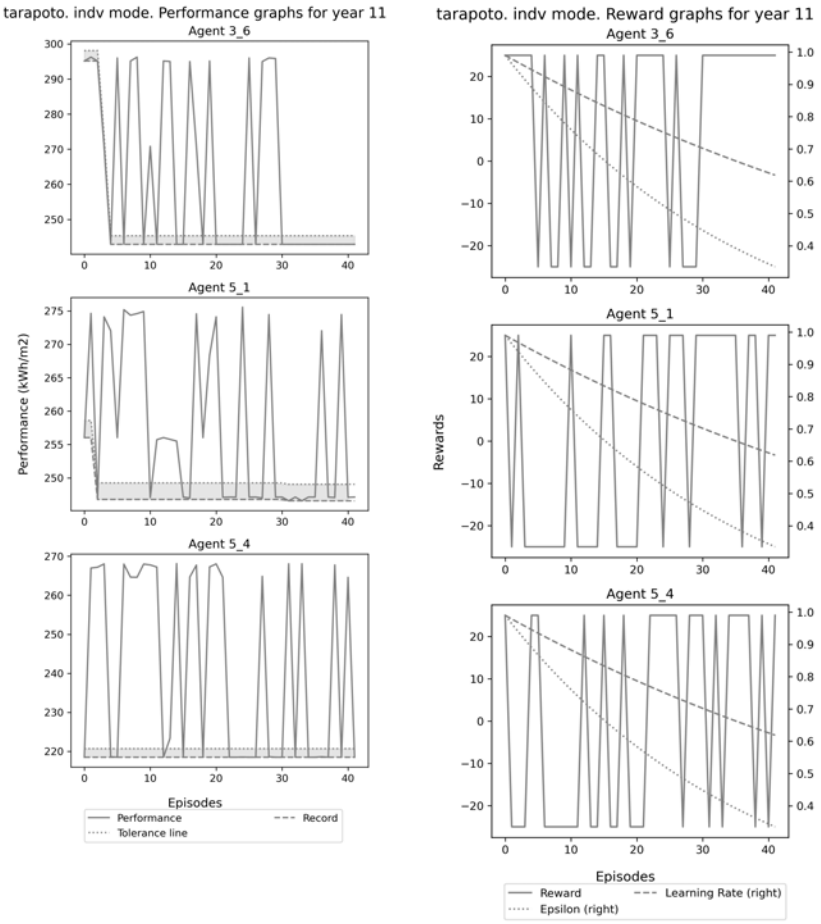
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (21)

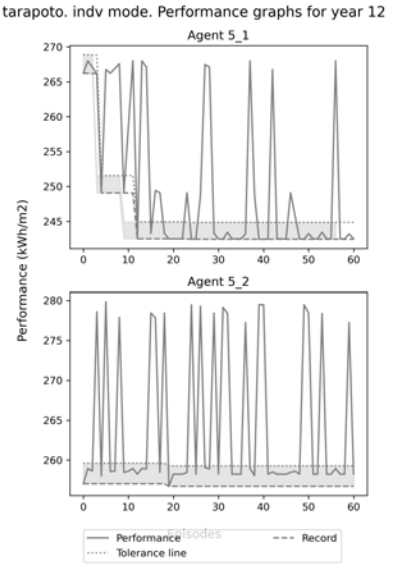
() Performance

() Rewards

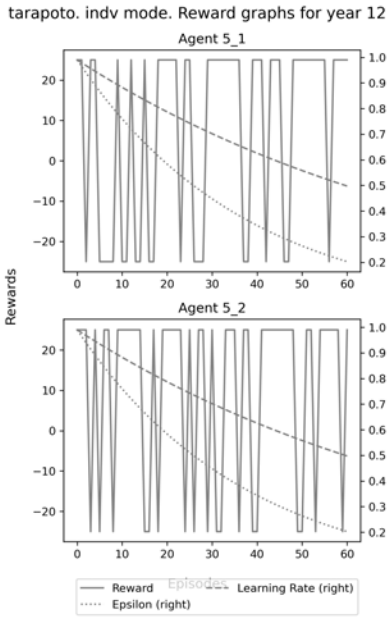


Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (22)

() Performance



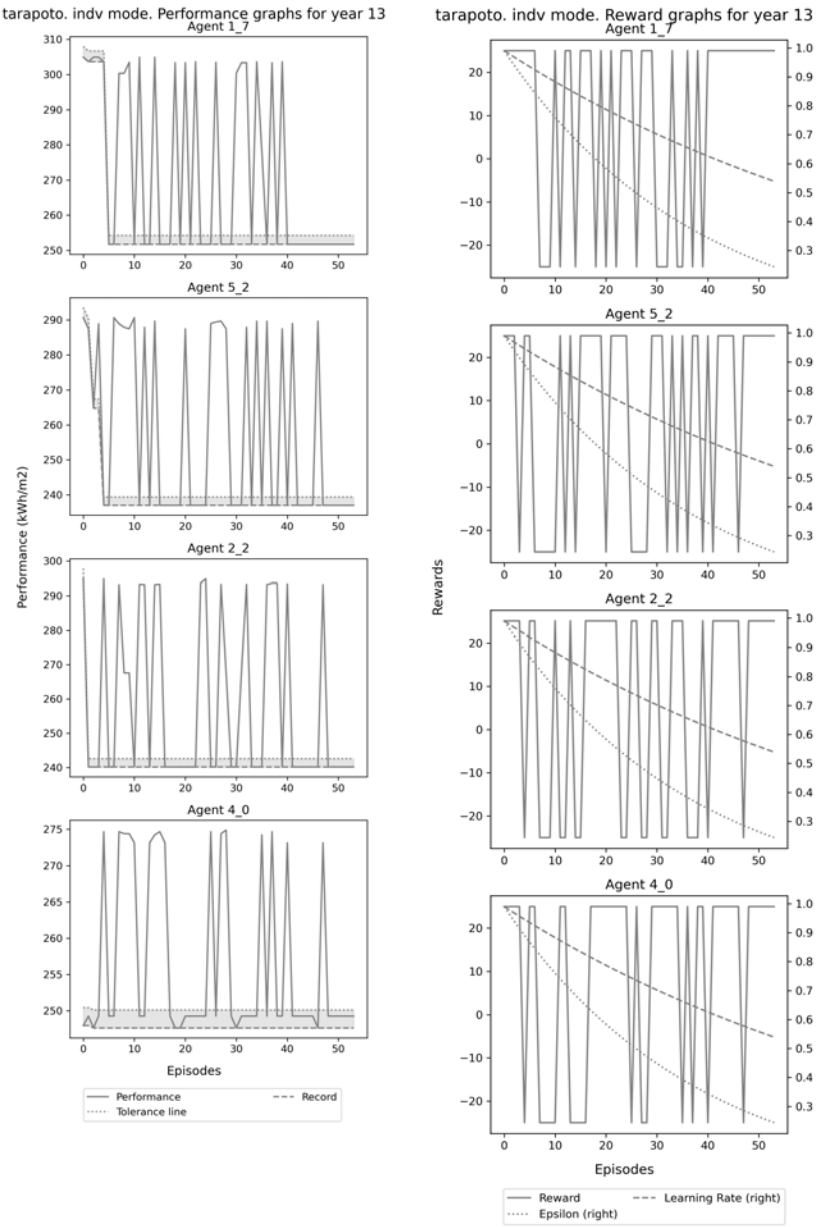
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (23)

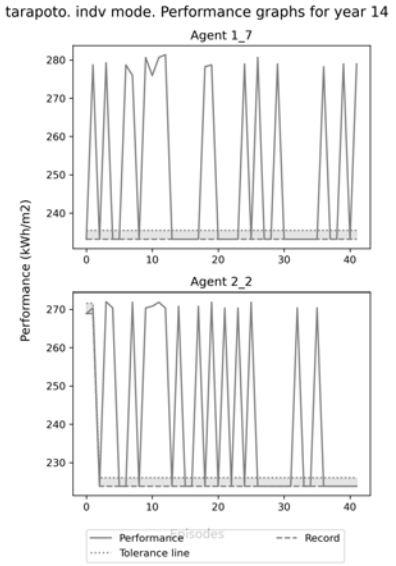
() Performance

() Rewards

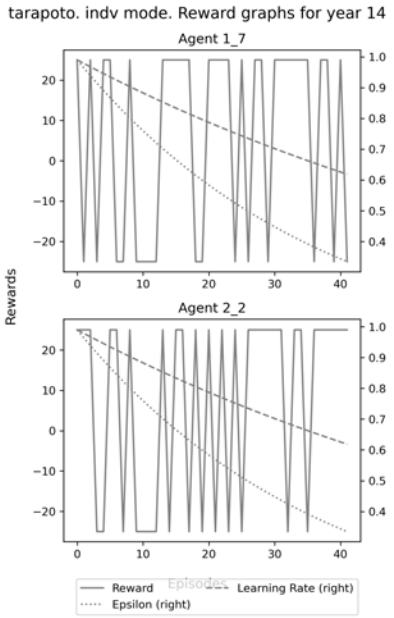


Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (24)

() Performance

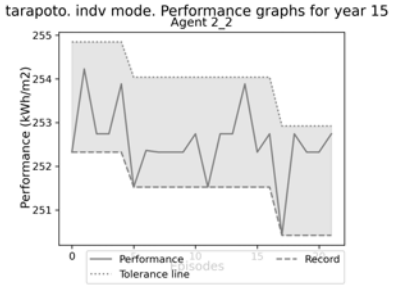


() Rewards

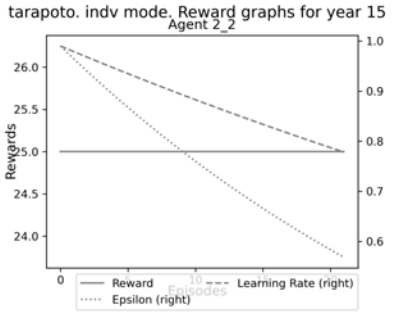


Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (25)

() Performance

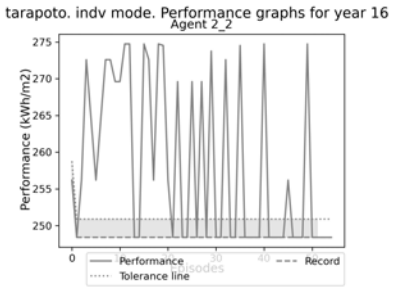


() Rewards

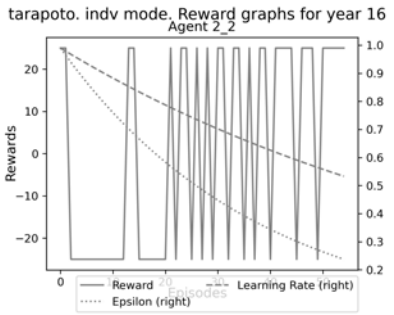


Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (26)

() Performance

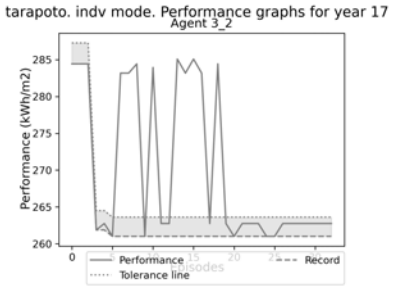


() Rewards

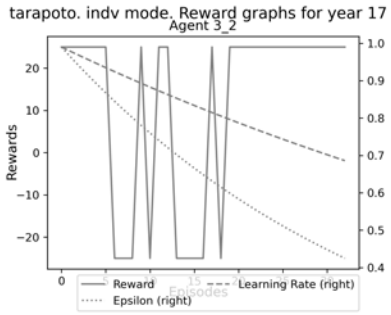


Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (27)

() Performance

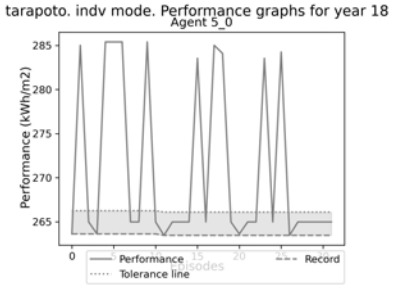


() Rewards

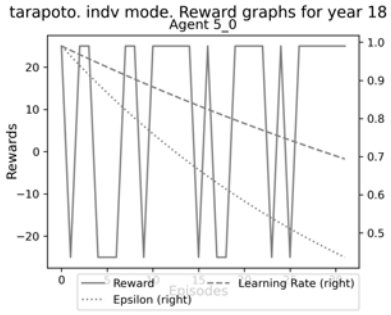


Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (28)

() Performance

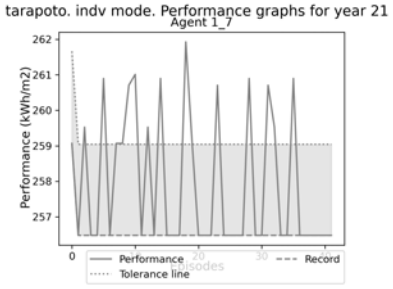


() Rewards

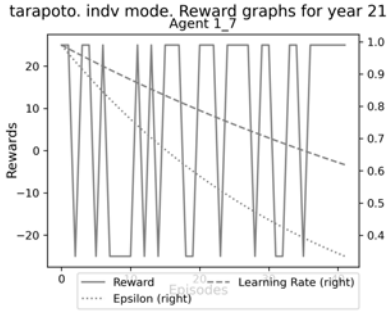


Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (29)

() Performance



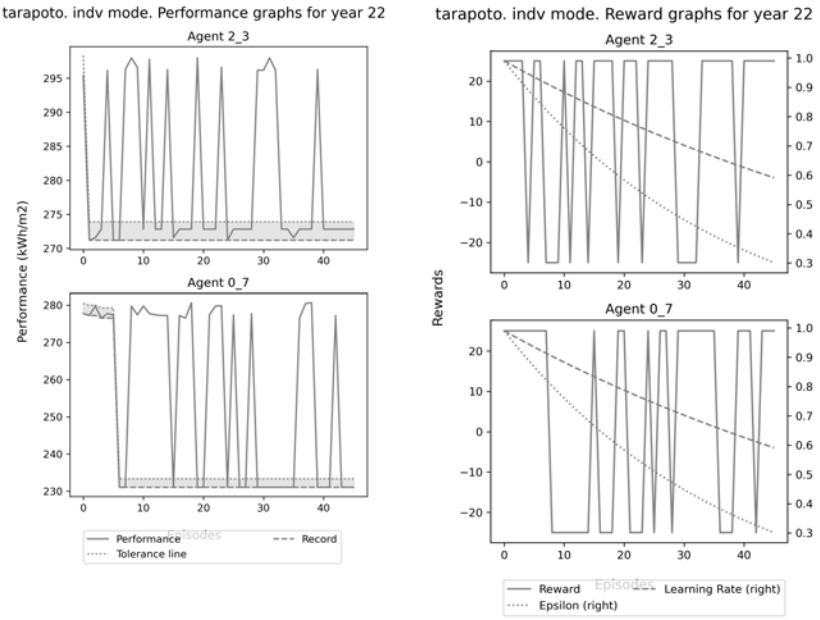
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (30)

() Performance

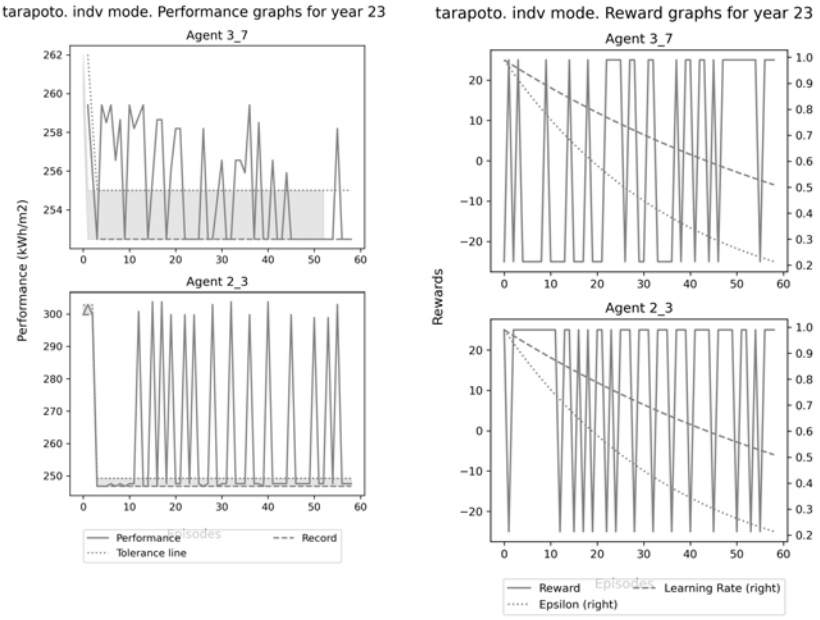
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (31)

() Performance

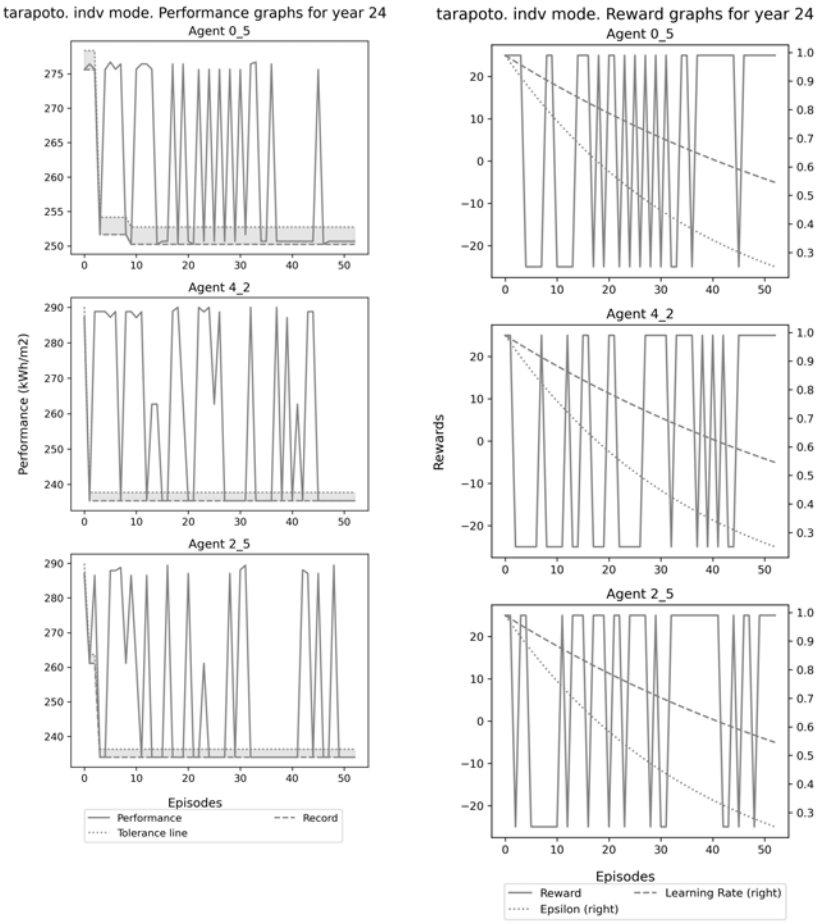
() Rewards



Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (32)

() Performance

() Rewards

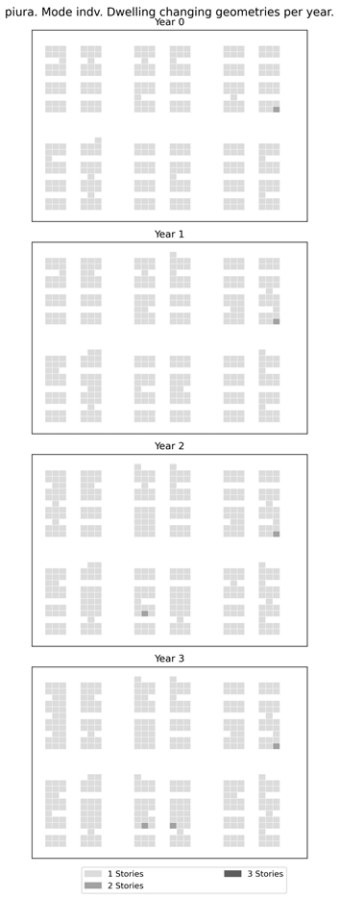


Learning process of all agents in Tarapoto competitive scenario for energy use Minimisation (33)

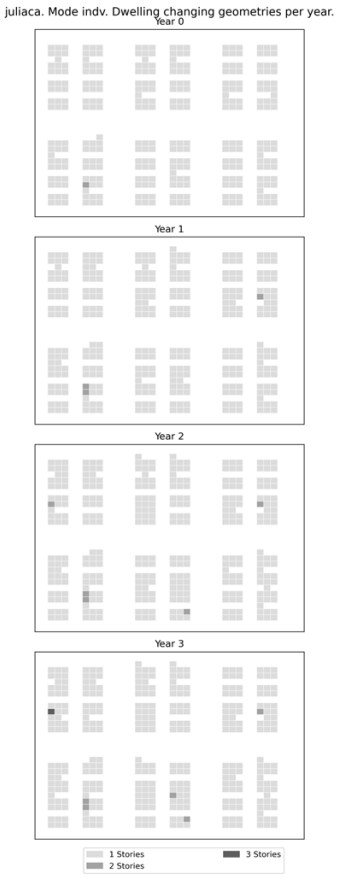
Appendix 21

Spatial outcomes comparison by location

() Piura



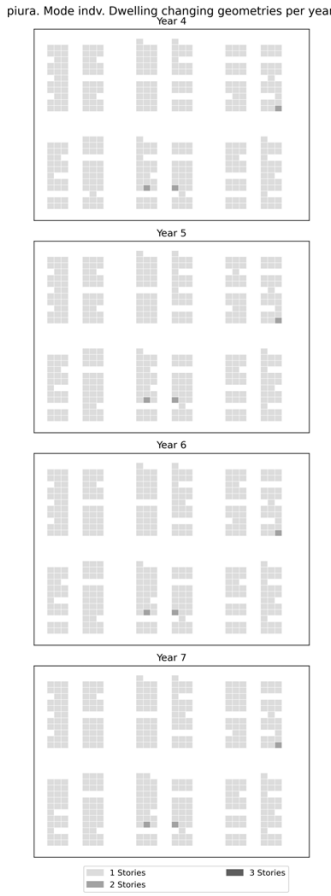
() Juliaca



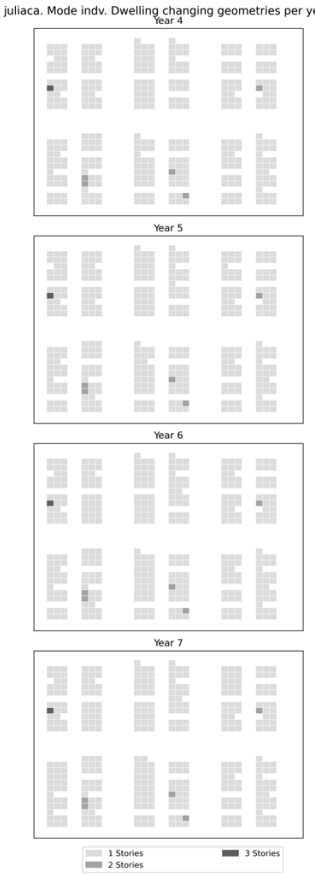
() Tarapoto

Emerging Spatial outcomes by year and location (1)

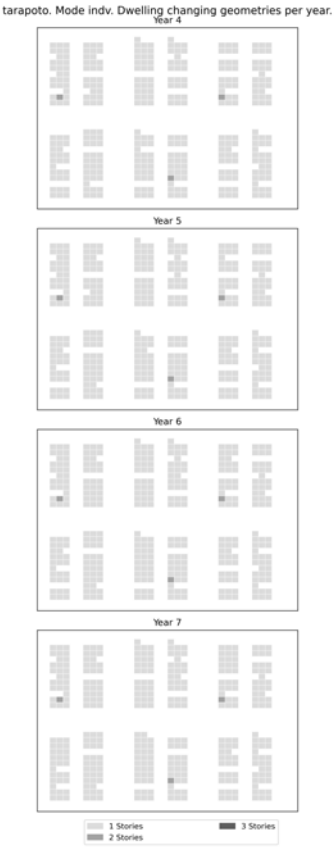
() Piura



() Juliaca

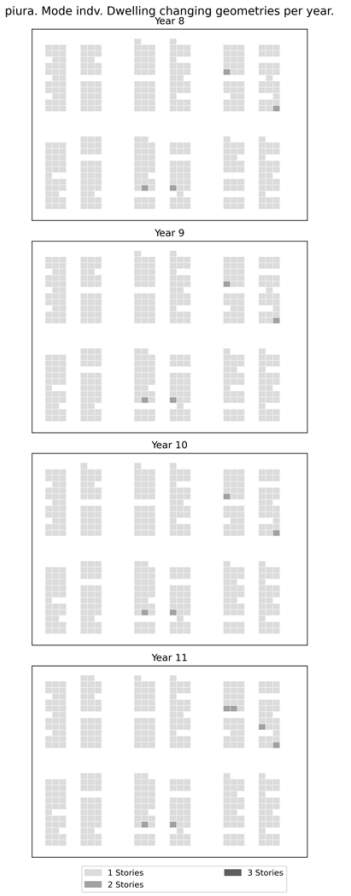


() Tarapoto



Emerging Spatial outcomes by year and location (2)

() Piura

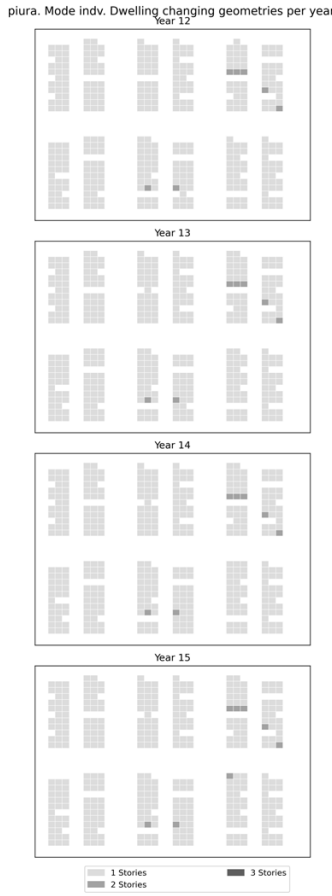


() Juliaca

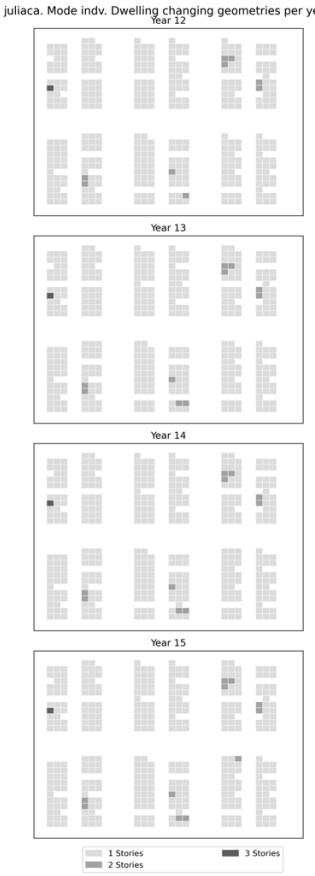
() Tarapoto

Emerging Spatial outcomes by year and location (3)

() Piura



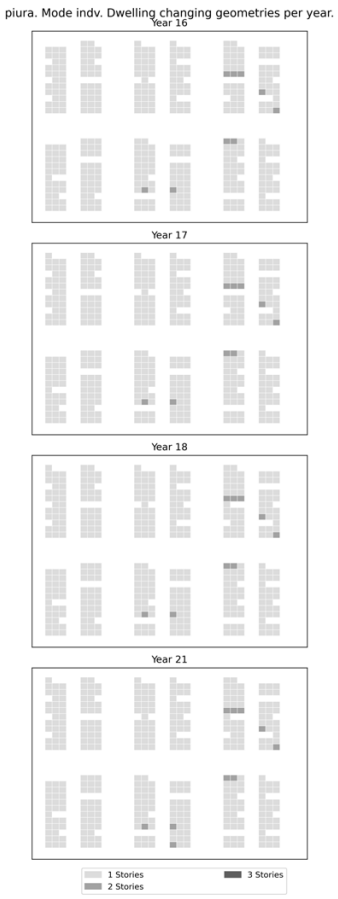
() Juliaca



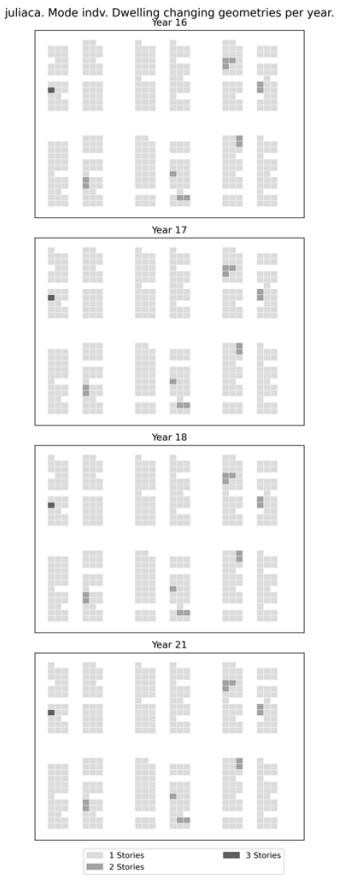
() Tarapoto

Emerging Spatial outcomes by year and location (4)

() Piura



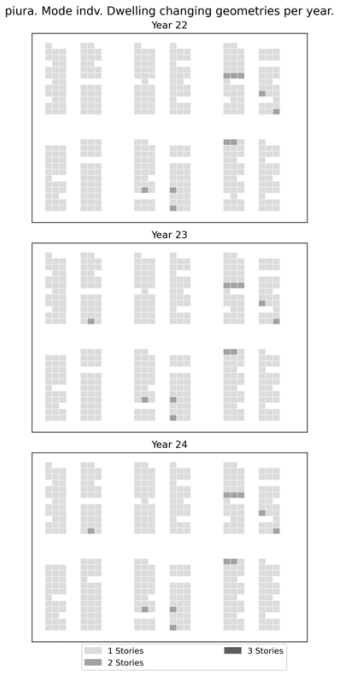
() Juliaca



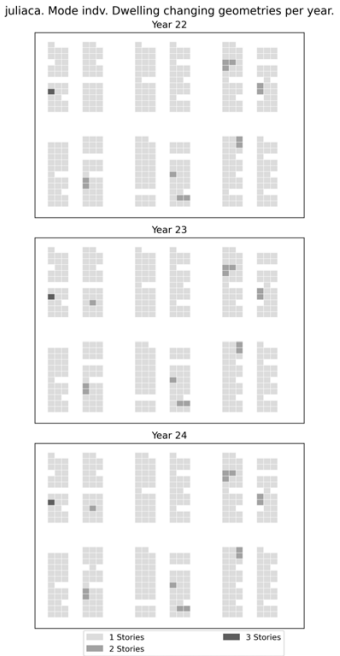
() Tarapoto

Emerging Spatial outcomes by year and location (5)

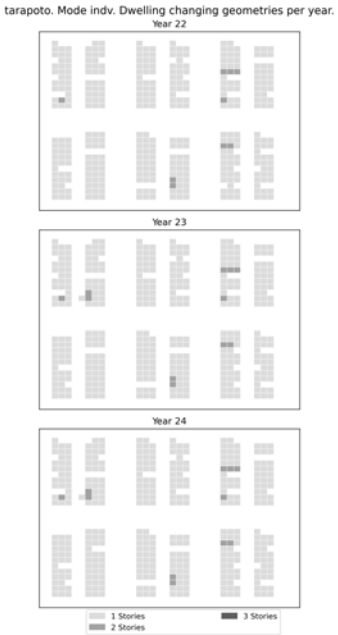
() Piura



() Juliaca



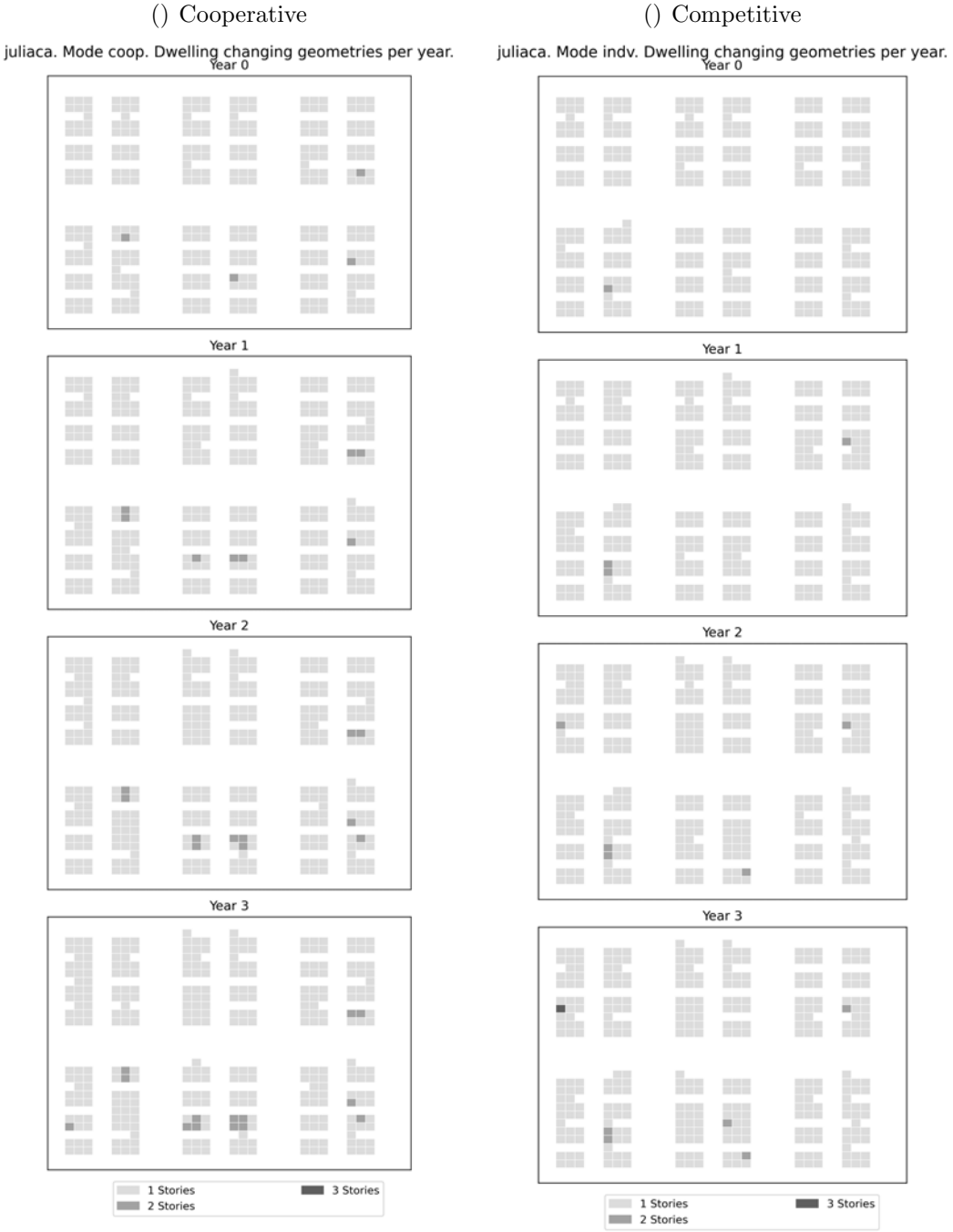
() Tarapoto



Emerging Spatial outcomes by year and location (6)

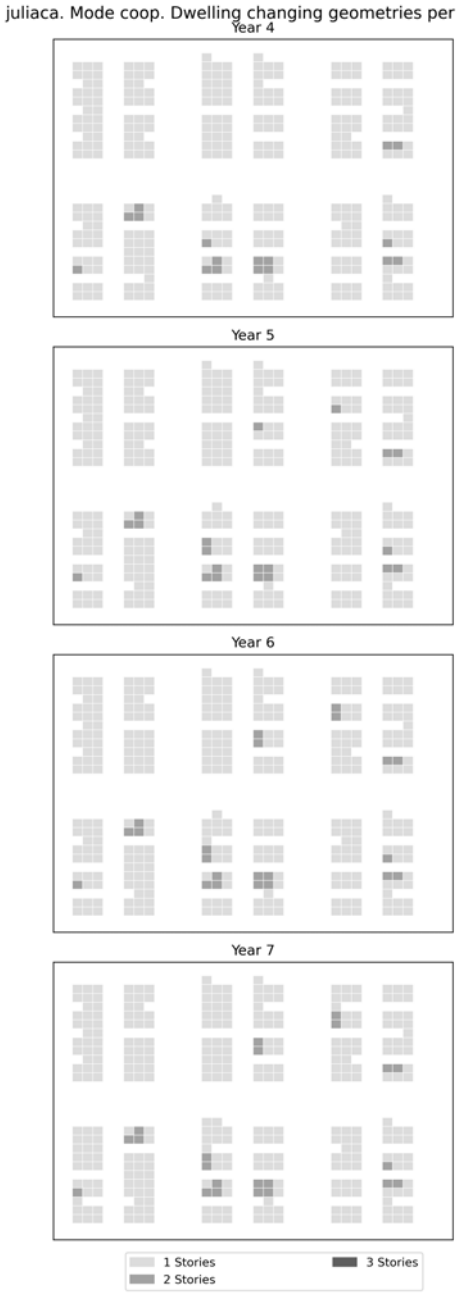
Appendix 22

Spatial outcomes comparison by scenario, competitive and cooperative

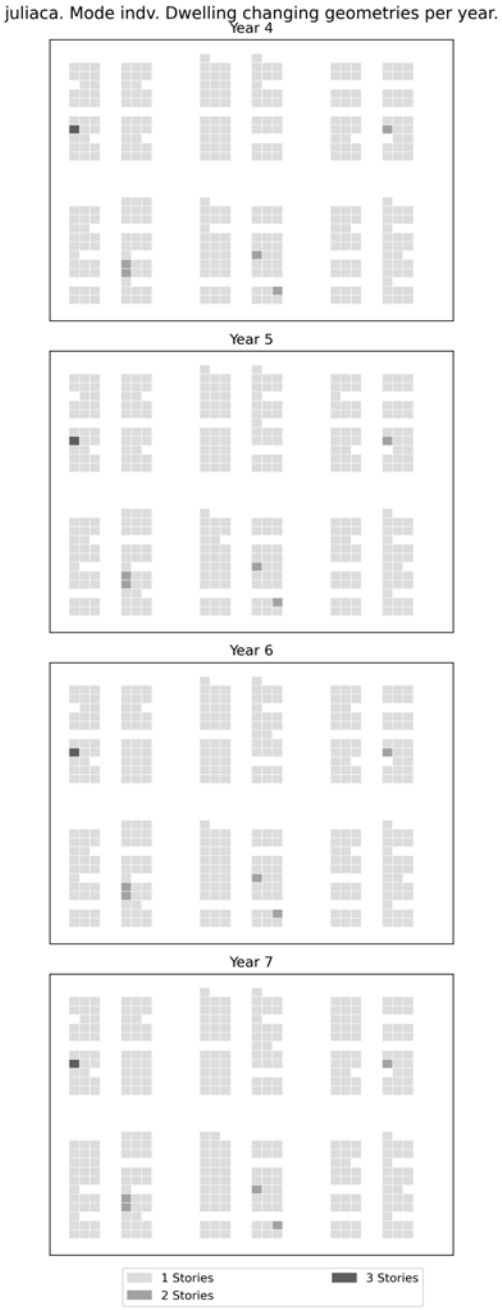


Emerging Spatial outcomes by year and scenarios cooperative and competitve (1)

() Cooperative



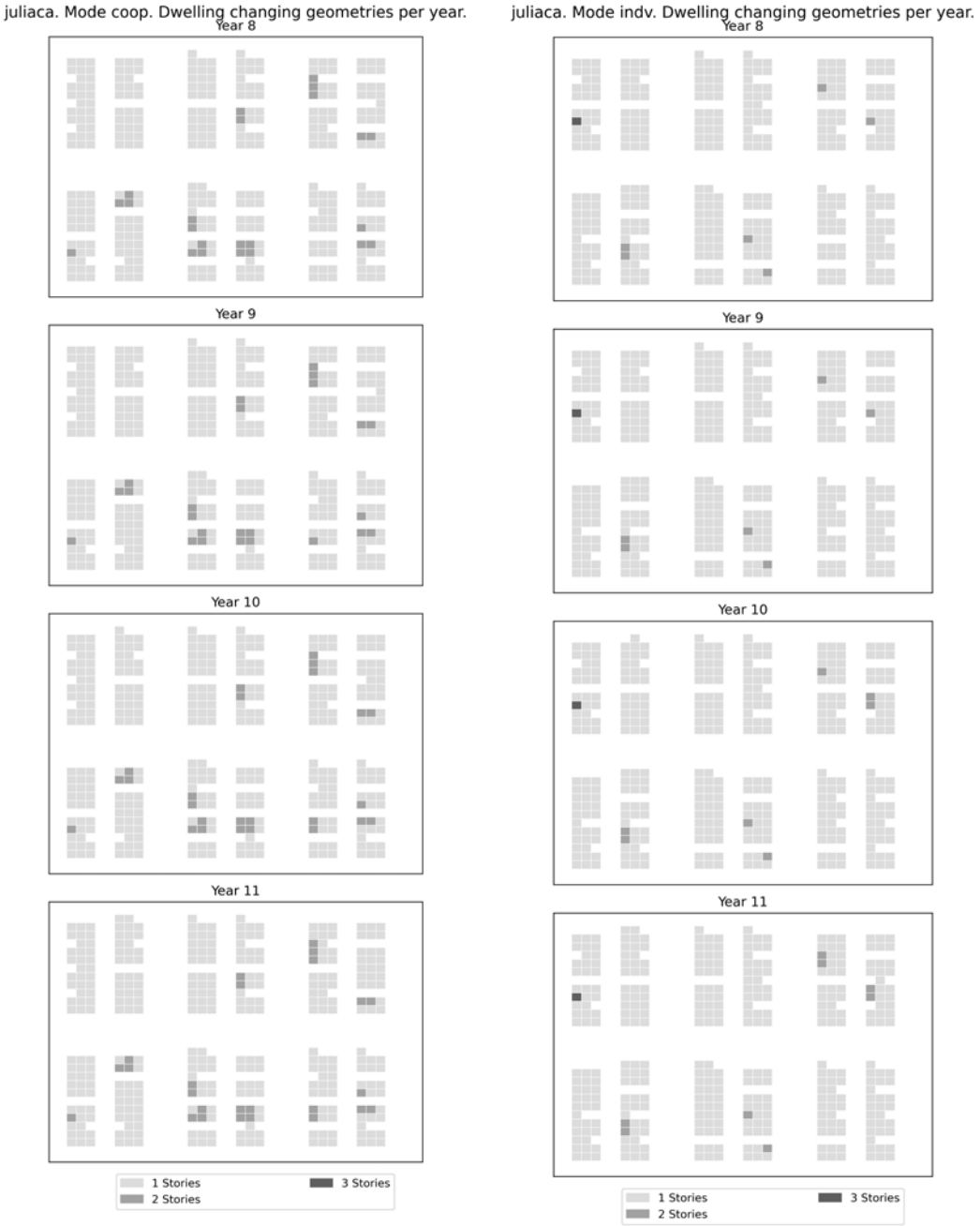
() Competitive



Emerging Spatial outcomes by year and scenarios cooperative and competitve (2)

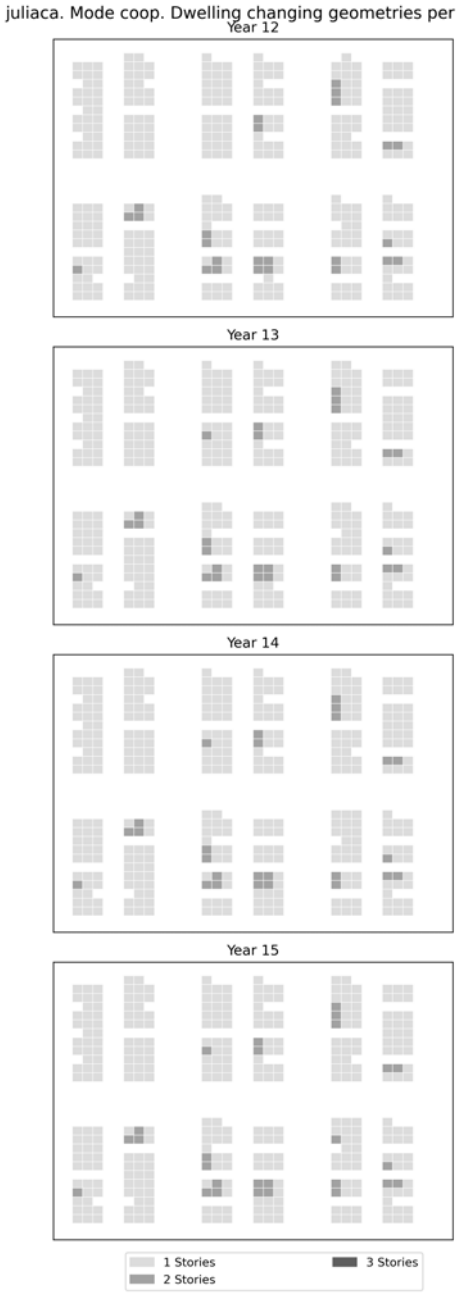
() Cooperative

() Competitive

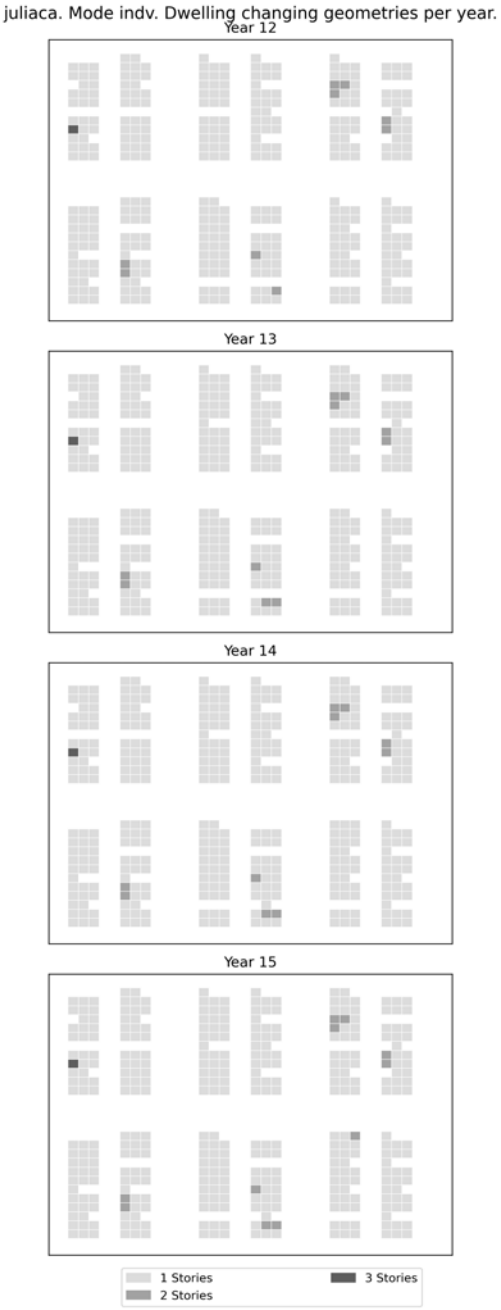


Emerging Spatial outcomes by year and scenarios cooperative and competitve (3)

() Cooperative



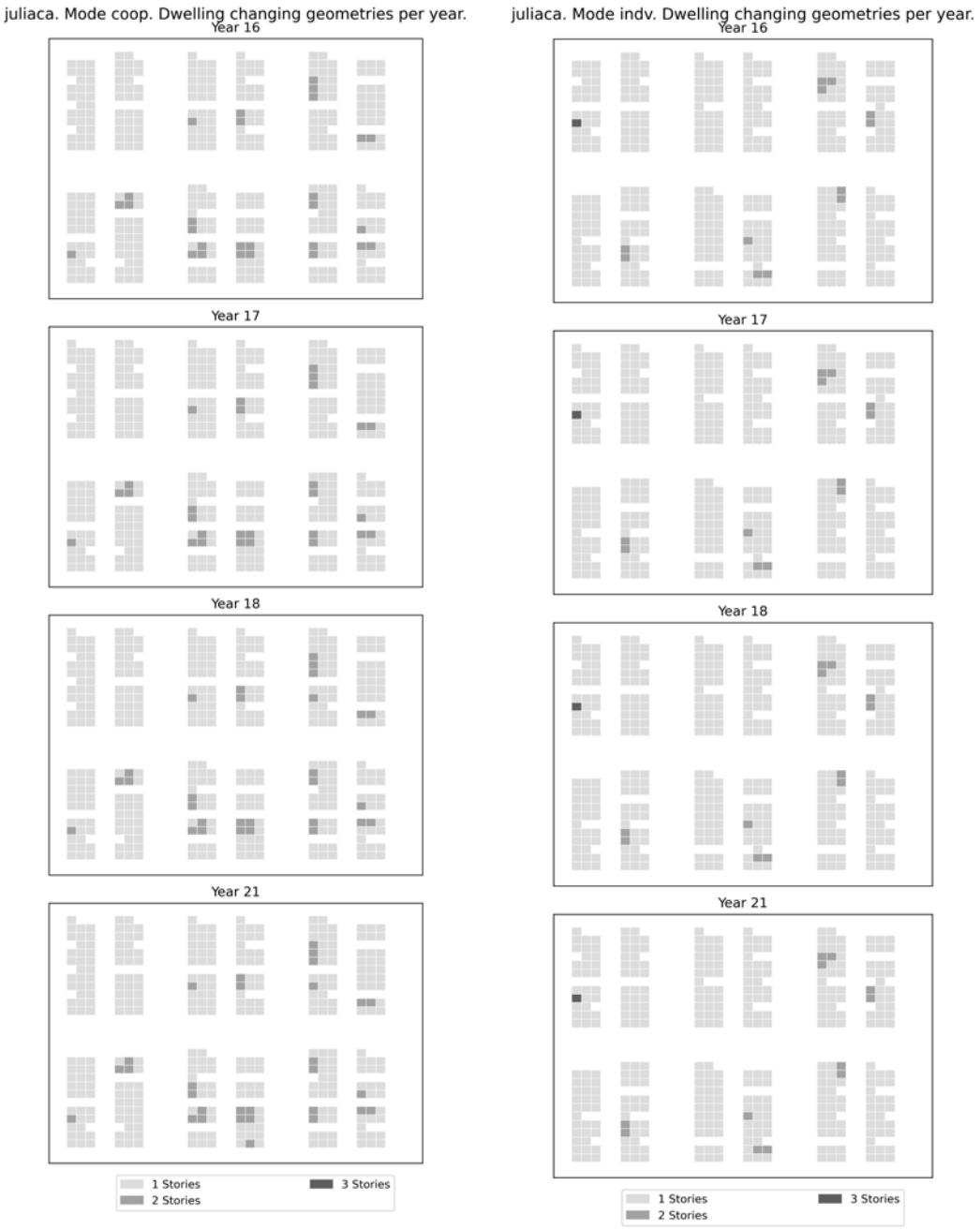
() Competitive



Emerging Spatial outcomes by year and scenarios cooperative and competitve (4)

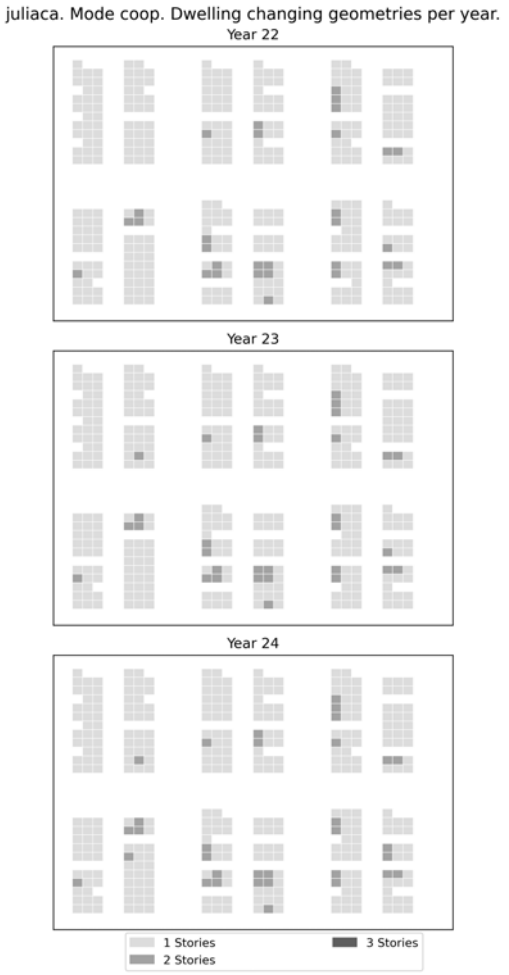
() Cooperative

() Competitive

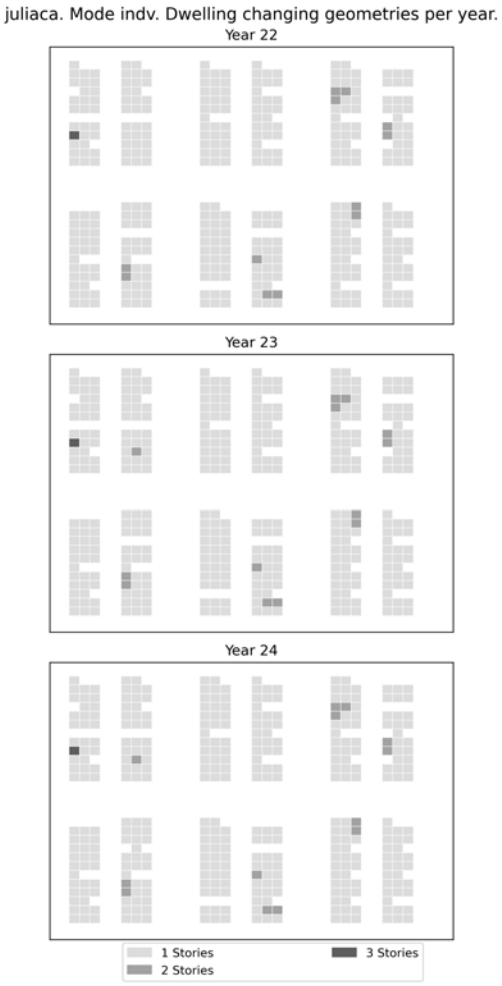


Emerging Spatial outcomes by year and scenarios cooperative and competitve (5)

() Cooperative



() Competitive



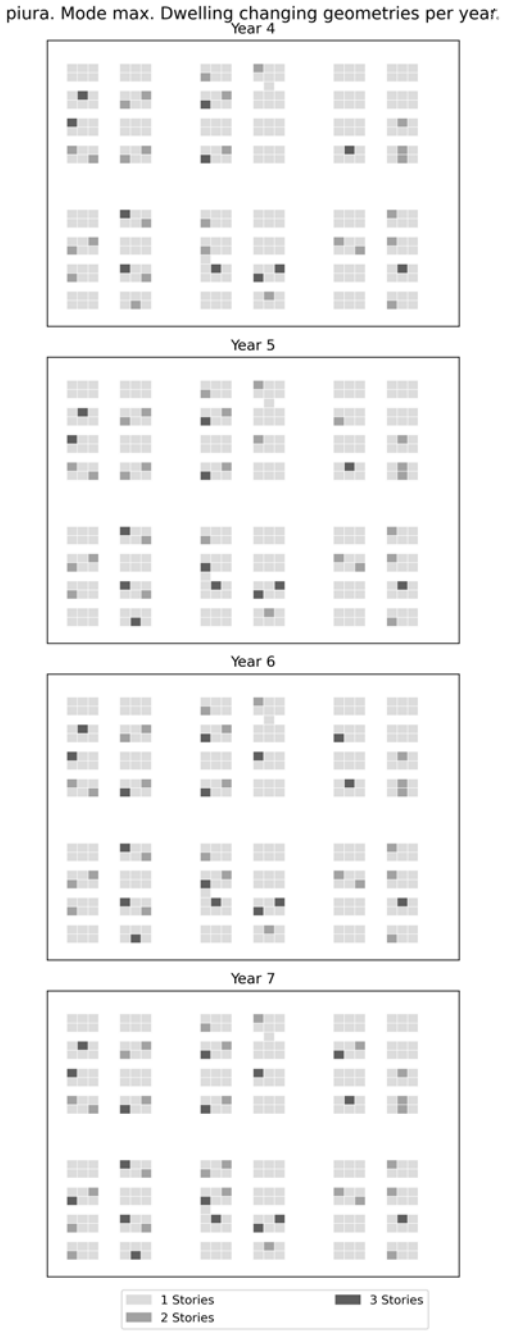
Emerging Spatial outcomes by year and scenarios cooperative and competitve (6)

Appendix 23

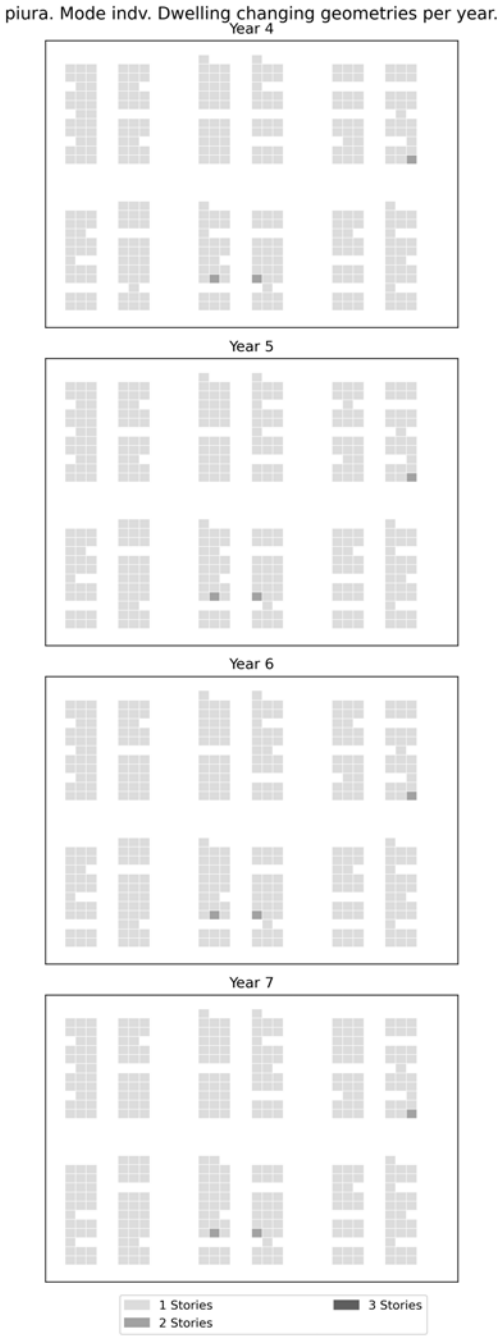
Spatial outcomes comparison by scenario, maximisation and minimisation

Emerging Spatial outcomes by year and scenarios maximisation and minimisation (1)

() Maximisation

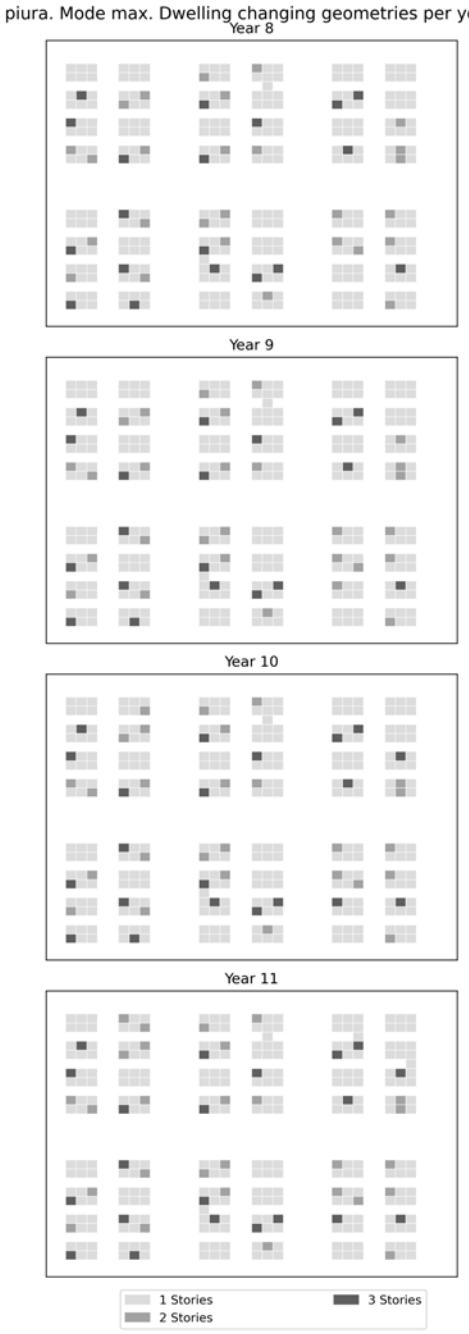


() Minimisation

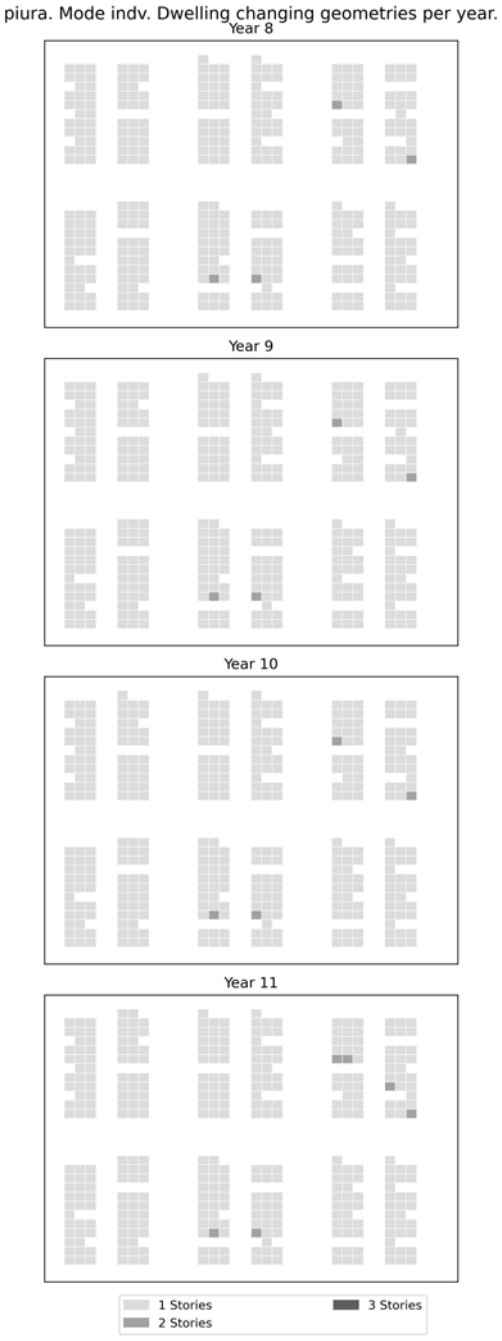


Emerging Spatial outcomes by year and scenarios maximisation and minimisation (2)

() Maximisation

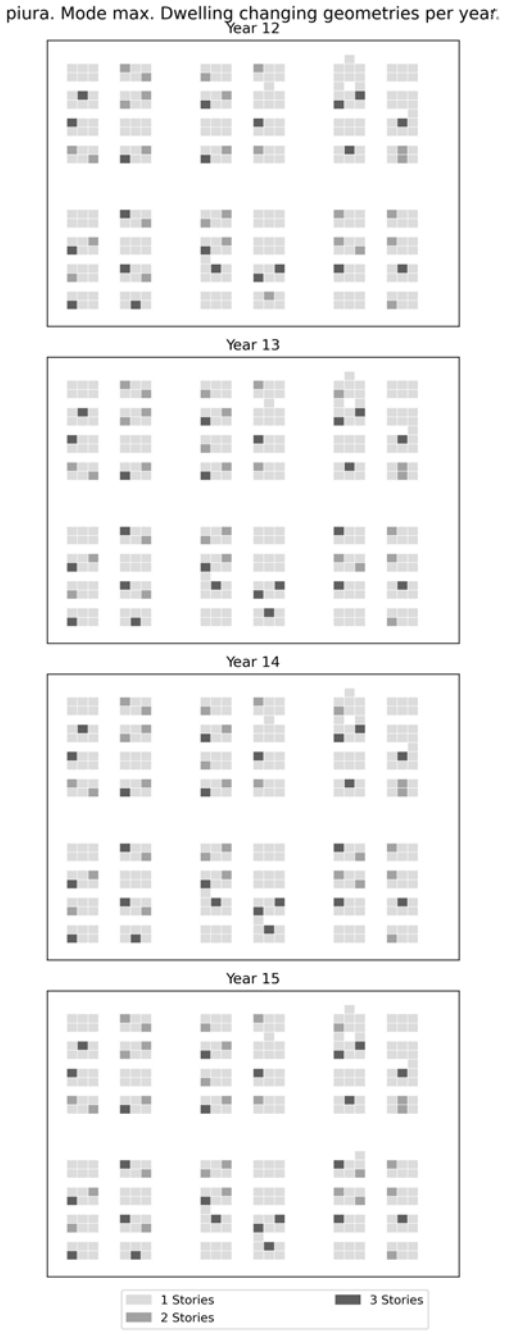


() Minimisation

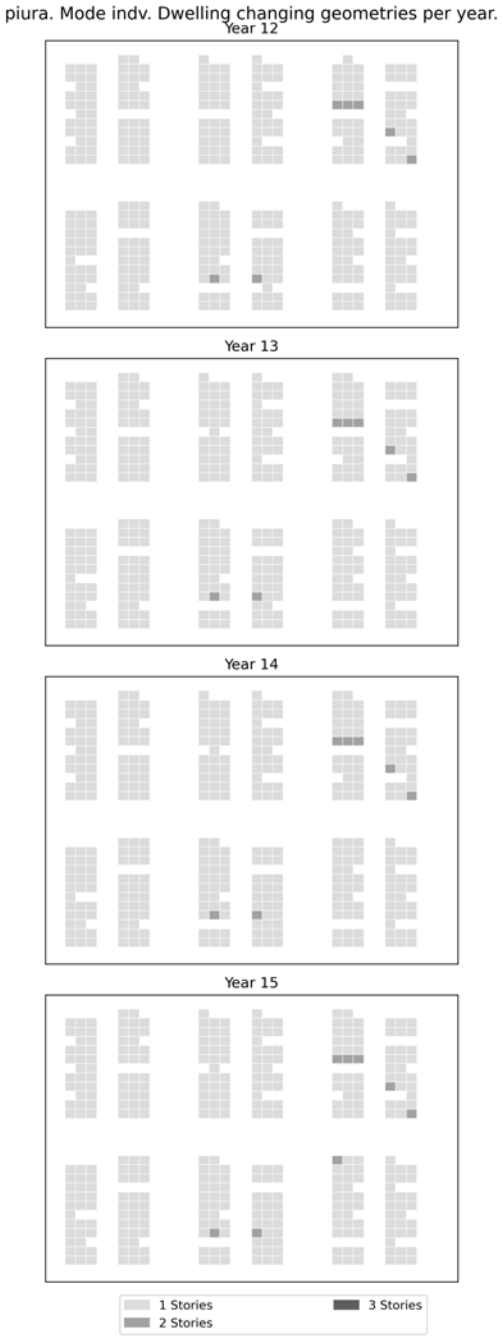


Emerging Spatial outcomes by year and scenarios maximisation and minimisation (3)

() Maximisation



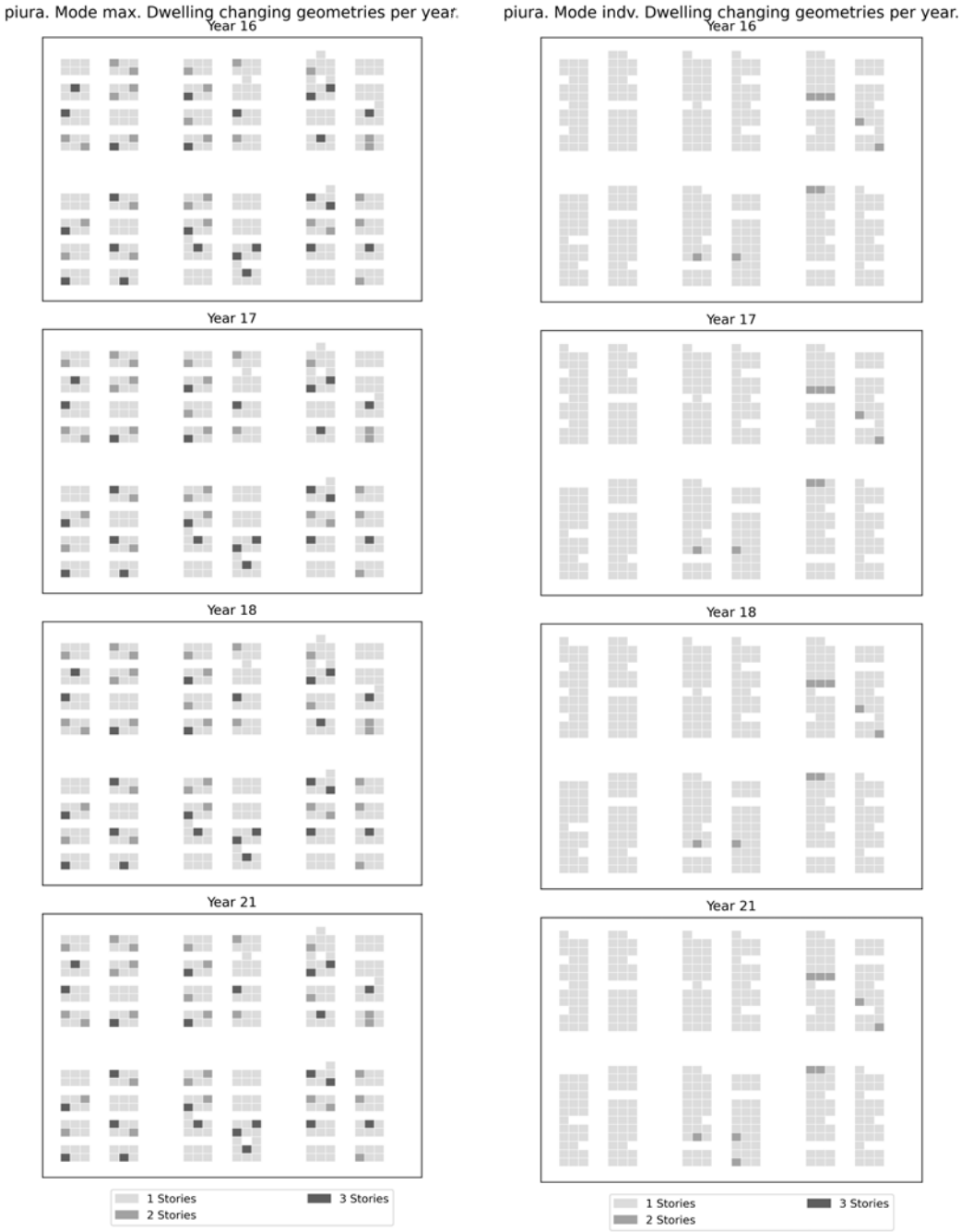
() Minimisation



Emerging Spatial outcomes by year and scenarios maximisation and minimisation (4)

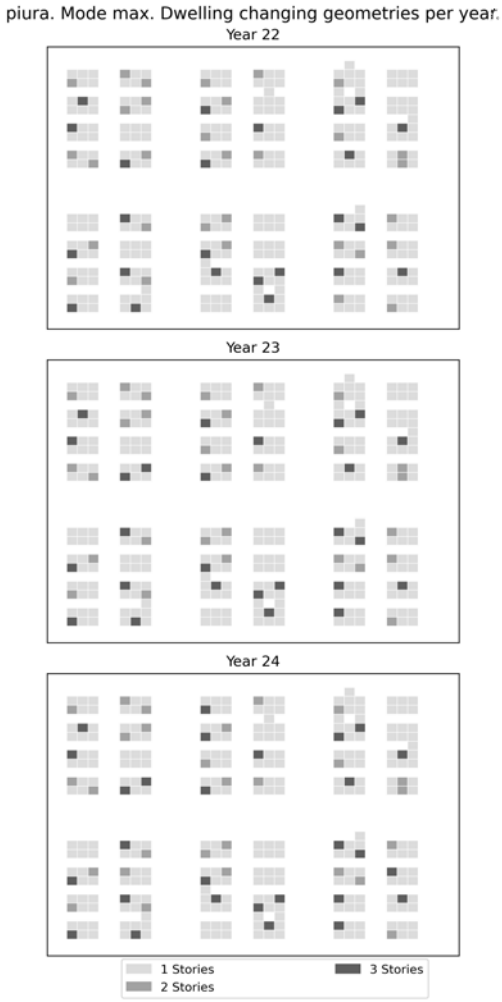
() Maximisation

() Minimisation

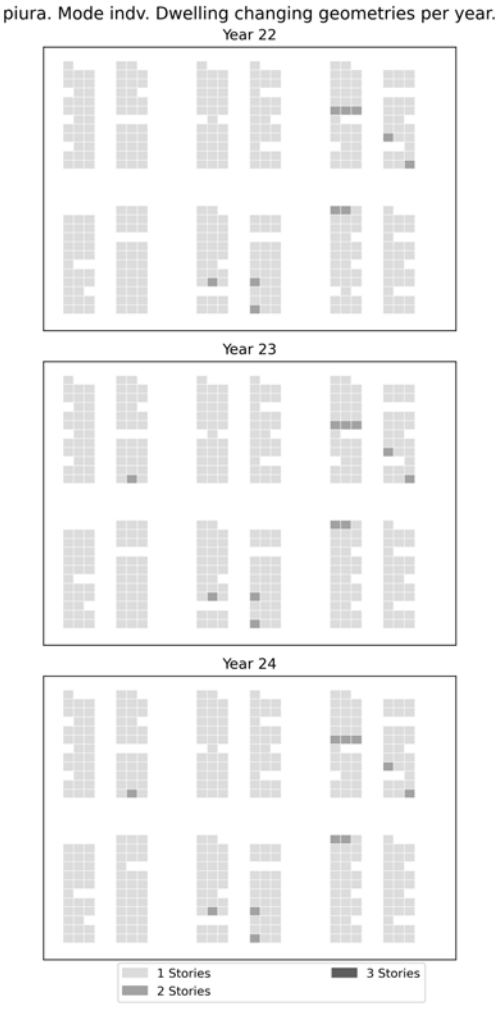


Emerging Spatial outcomes by year and scenarios maximisation and minimisation (5)

() Maximisation



() Minimisation



Emerging Spatial outcomes by year and scenarios maximisation and minimisation (6)

References

Cardenas Vargas, V. (2013, 11). Planeamiento integral de la construcción de 142 viviendas unifamiliares en la ciudad de puno aplicando lineamientos de la guía del pmbok. Retrieved from http://hdl.handle.net/20.500.12404/4910

De La Cruz, R., Salazar, C., & Coello, F. (2021). *Informe de resultados. consumo y usos de los hidrocarburos líquidos y glp. encuesta residencial de consumo y usos de energía – 2019-2020.* Retrieved from https://cdn.www.gob.pe/uploads/document/file/2560069/ERCUE%20Hidrocarburos%20%202019-2020.pdf?v=1638461831

De La Cruz, R., Salazar, C., & Santos, W. (2021). *Informe de resultados: Consumo y usos de la electricidad 2019-2020.* Retrieved from https://cdn.www.gob.pe/uploads/document/file/2691020/ERCUE%20Electricidad%202019-2020.pdf

INEI. (2016). *Encuesta nacional de hogares sobre condiciones de vida y pobreza - 2015 (enaho).* Instituto Nacional de Estadistica e Informatica. Retrieved from https://proyectos.inei.gob.pe/iinei/srienaho/Consulta_por_Encuesta.asp ([Data Set])

INEI. (2017). *Encuesta nacional de hogares sobre condiciones de vida y pobreza - 2016 (enaho).* Instituto Nacional de Estadistica e Informatica. Retrieved from https://proyectos.inei.gob.pe/iinei/srienaho/Consulta_por_Encuesta.asp ([Data Set])

INEI. (2018). *Encuesta nacional de hogares sobre condiciones de vida y pobreza - 2017 (enaho).* Instituto Nacional de Estadistica e Informatica. Retrieved from https://proyectos.inei.gob.pe/iinei/srienaho/Consulta_por_Encuesta.asp ([Data Set])

INEI. (2019a). *Enaho panel 2015-2019.* Instituto Nacional de Estadistica e Informatica. Retrieved from https://proyectos.inei.gob.pe/iinei/srienaho/Consulta_por_Encuesta.asp ([Data Set])

INEI. (2019b). *Encuesta nacional de hogares sobre condiciones de vida y pobreza - 2018 (enaho).* Instituto Nacional de Estadistica e Informatica. Retrieved from https://proyectos.inei.gob.pe/iinei/srienaho/Consulta_por_Encuesta.asp ([Data Set])

INEI. (2020). *Encuesta nacional de hogares sobre condiciones de vida y pobreza - 2019 (enaho).* Instituto Nacional de Estadistica e Informatica. Retrieved from https://proyectos.inei.gob.pe/iinei/srienaho/Consulta_por_Encuesta.asp ([Data Set])

Ministerio de Vivienda. (2006, 6). *Norma is.010.* Diario Oficial El Peruano.

Ministerio de Vivienda. (2015, 10). *Resolucion ministerial 286-2015-vivienda.* Diario Oficial El Peruano. Retrieved from https://busquedas.elperuano.pe/dispositivo/NL/1305437-1

Ministerio de Vivienda. (2016, 10). *Resolucion ministerial 373-2016-vivienda.* Diario Oficial El Peruano. Retrieved from https://busquedas.elperuano.pe/dispositivo/NL/1447405-5

Ministerio de Vivienda. (2017, 10). *Resolucion ministerial 415-2017-vivienda.* Diario Oficial El Peruano. Retrieved from https://busquedas.elperuano.pe/dispositivo/NL/1581335-5

Ministerio de Vivienda. (2018, 10). *Resolucion ministerial 370-2018-vivienda.* Diario Oficial El Peruano. Retrieved from https://busquedas.elperuano.pe/dispositivo/NL/1707409-1

Ministerio de Vivienda. (2019, 10). *Resolucion ministerial 351-2019-vivienda.* Diario Oficial El Peruano. Retrieved from https://busquedas.elperuano.pe/dispositivo/NL/1821938-5

Oraiopoulos, A., Fennell, P., Amrith, S., Wieser, M., Korolija, I., & Ruyssevelt, P. (2022). Towards a universal access to urban building energy modelling-the case of low-income, self-constructed houses in informal settlements in lima, peru. In *Bso-vi 2022 sixth building simulation and optimisation virtual conference.*

Superintendencia de Banca Seguros y AFP. (2024, 2). *Tasas de interés promedio.* Retrieved from https://www.sbs.gob.pe/estadisticas/tasa-de-interes/tasas-de-interes-promedio