An Autonomous Mizusumachi System

May 2018

Eduardo Guilherme Ferreira Morais de Araújo *Instituto Superior Técnico, Universidade de Lisboa*
Lisboa, Portugal
eduardo.araujo@tecnico.ulisboa.pt

1. Introduction

Lean Manufacturing System is a systematic method for waste minimization within a manufacturing system without sacrificing productivity [3]. Following this methodology, operators must solely focus on performing their own assignment, under the prescribed Standard Time of the task; each time an operator has to perform activities others than his task – for instance feeding the machine, performing set-ups or move to get necessary tools – he is wasting time. All necessary resources (e.g. tools, parts to assembly, and so on) must be provided to the operator in order for him to perform his task [1, 6].

The Japanese word *mizusumashi* – in English *water spider* – is a common term used in Lean Manufacturing bibliography to refer to an internal supply operator, whose task is to supply the required materials for each workstation (WS). Thus, *Mizusumashi* is the core element of a production management system, whose role is to increase the operators' productivity [1].

In a nutshell, using a *Mizusumashi* increases productivity and minimizes waste by sparing the WS operator to unnecessary tasks, while assuring constant, regular and safe WS feeding [6].

2. Problem Description

In the last section it was presented the concept of *Mizusumachi* including the benefits of implementing such a system. The next paragraphs will focus on describing the problem to be solved.

A given factory has 3 production lines outputting Products A, B and C. Depending on the way these parts are assembled, the final product will be different, enabling the company to have multiple customers. In order to satisfy each customer's demands the company decided to implement 3 different assembly lines, each with its own stock storage – Storages 1, 2 and 3.

The same company wants to develop an autonomous *Mizusumachi* system whose task is to move Products A, B and C to the Storage Area according to their demands.

The transportation task is subject to the following constraints:

- each product has a production time that follows a particular probability distribution
- each storage has a demand that follows a particular probability distribution

Fig. 1 presents a schematic of the described task.

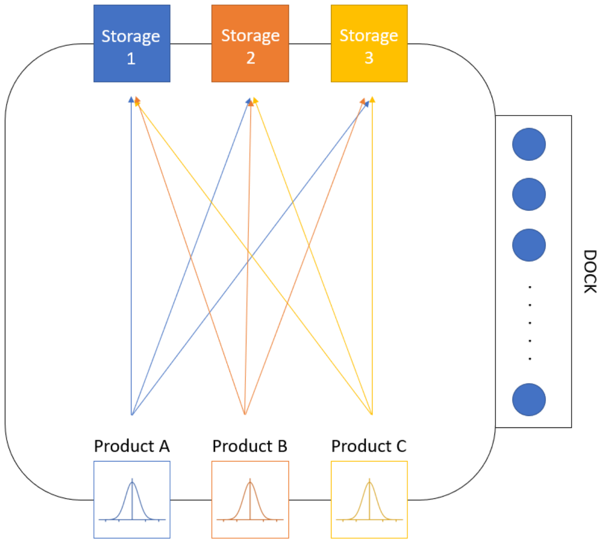

Figure 1. Schematic of the problem

3. Goals

The proposed project has three major goals: 1) to evaluate the impact of the number of Robots on the overall system performance; 2) to compare and assess which model of agency (*Cooperative* or *Free-Agent*) would produce better results; 3) to determine which *Ratio of Personalities* (see appendix A) should one rely on in order to improve the overall performance.

4. Methodology

4.1. Assumptions

- The environment is completely reachable and observable
- All agents have the same sensors and actuators, i.e. agents differ only in decision making and execution architectures.
- If at any given time there are 3 products waiting to be dispatched at a given product line, that line will stop production until there is space available again.

4.2. Modelling

This section focuses on the system's modelling. It starts with a brief overview of its function mechanisms as well as the characterization of both the *agents* and *environment*.

1) System: A multi-agent approach of the problem at hand requires the definition of *agents* and *environment*. Based on the description of the problem in section 2, three types of *agents* were created: *Assembly Line* (AL), *Production Line* (PL) and *Robot*.

Environment Dynamics: The fundamental dynamics of the proposed system is simple and can be described as follows:

- The three AL require different types of product to be produced based on different demand distributions.
- Each PL is restricted to producing a particular type of product and its production starts on receiving a request specifying the product type and the amount required. This agents are yet further constrained: if at any given time there are 3 products waiting to be dispatched production will be halted until there is space available.
- As soon as a product is available, Robots are assigned to a transportation task.

Thus, the system can be viewed as a closed-loop based on three key moments: request, production, delivery. Akin to any supply chain information and goods flow in opposite directions.

Environment: Following the Russell and Norvig [9] classification of environment properties, one can easily characterize the environment of the problem herein presented as: *accessible*, *nondeterministic*, *nonepisodic*, *dynamic* and *discrete*.

Agents: Similarly, the aforementioned agents can be divided into two classes: reactive (AL and PL) and hybrid (Robots) [9, 10].

2) Production Time and Demand: In order to have a dynamic environment it was required to have some uncertainty associated with both *Production Time* and *Demand* features.

Each PL agent is characterized by a particular production time distribution. To model this feature the following expression was used:

$$P_T = p_t + \Delta t$$

where p_t is a fixed number unique to each PL and Δt is a random number.

Conversely, a key property of AL agents is their demand distributions which were modelled resorting to a periodic function given by the following expression:

$$D = \frac{1}{2}\left[\sin\left(\frac{2\pi t}{T}\right) + 1\right]$$

where T is the period and t the time.

Uncertainty is modelled making T depend on a parameter n of random value, as follows: $T = 48n$.

A request is published on the appropriate server when the previous function equals 1.

3) Communication: As mentioned before, agents rely on explicit communication. The next paragraphs will focus on the designed protocol and its mechanisms.

Servers: In order to share information, each AL and PL agents have servers where they publish messages. AL agents have public servers used to request products. PL agents listen to these server and produce the requested products. Robots that are available to perform a delivery task listen to and publish on specific PL public servers.

Request Standards: An AL request respect the following standard: *{request_id: (time, product_type, amount)}*, where *request_id* is a string of the form *AL_id - Number of request*, e.g. *1-22* refers to the 22$^{\text{nd}}$ request placed by AL 1. Since this identifier is unique PL agents can use it as product identification.

Task Assignment: When a PL agent finishes producing a new product, an auction is created. Robots listen to the PL auction servers and if they are available to perform a delivery task, a bid is published on the appropriate auctioneer's bid server. The winner Robot is awarded a reward that can be used in future bids. The type of auction is *first-price sealed-bid*, which means auctioneers must reward the highest bid.

Product Delivery: When AL agents receive a product, its *unique_id* is deleted from the request server, closing the *request, production, delivery* loop.

4) Robots: Robots are *hybrid agents* which means they have both *reactive* and *deliberative* modules. The former is used to avoid collisions, while the latter focuses on goal definition and planning. This type of agent can have one of four goals at any given time: *bid, get product, deliver product* or *go back to base.*

Planning and Re-Planning: A key feature of intelligent agents is their ability to plan and re-plan. In a multi-agent system agents interact with each other. This might be specially problematic if agents are mobile, where agents are required to avoid collisions while seeking to achieve a goal.

Route Planning: Consider an agent wanting to move from a position x_{c_k} at the time instant k to a target position x_t. In each time instant the agent can move to a different location, which means that for a given x_{c_k}, there is a set of possible next positions $x_{c_{k+1}}$ that the agent can move to. Thus, the route is simply the sequence of movements a given agent can perform, i.e. the collection of locations $x_{c_{k+1}}$. A simple, yet efficient way of choosing which possible position to move to is by using an heuristic-based decision process. A typical heuristic is the *euclidean distance*: in each time step k the agent computes the distance between all possible

$x_{c_{k+1}}$ and x_t, then the agent chooses the position with the minimum value associated [9]. In order to avoid collisions, it suffices to render the position of other agents as invalid.

Personalities: Attending to the proposed goals of the project (section 3), it is worth understating the impact deliberative agents (Robots) with different driving mechanisms (called *personalities* hereafter, by abuse of language) might have on the overall system performance. For this purpose, two types of personality were designed: *risk prone* and *risk averse*. These personalities will drive Robots to choose an auction in detriment of another, i.e. these agents will decide which auction to bid based on *utility*.

Risk Prone vs Risk Averse: Risk prone Robots compute the *utility* of a given auction by $U(x) = x^2$. Whereas, *risk averse* agents have a *utility function* given by a modified Grayson utility of money [5], given by: $U(x) = 22.09 \log(x + 150,000)$

Tolerance to Risk: In order to have a more heterogeneous population of agents, a second layer of personality was considered: the Robots' tolerance to risk. A good intuition to understand this second layer is the idea that *an agent is willing to risk giving up the most rewarding auction, if by choosing another one the probabilities of him winning that new auction increases*. This tolerance to risk is then modelled defining a condition such as *choose auction a_2 instead of a_1 if $U(a_2) \geq pU(a_1)$*, where $p \in [0, 1]$.

Cooperative Agents: It should also be noted, in the *Cooperative* model of agency Robots are not concerned with selfish gains, which means they try to spread themselves across all auctions, in a way of dispatching the greatest number of produced products waiting to be transported.

4.3. Performance indices

Collective performance:

- *Gini Coefficient*
- *Idle Time*
- *Produced Products*
- *Waiting Time*

Individual Performance:

- *Wealth*

For a thorough definition of the aforementioned metrics, please refer to Appendix A.

4.4. Data Collection

To find the overall distributions generated by the model, and to analyse how they drive the model's outputs and behaviours it was used a data collection approach based on batch runs.

This methodology was employed for each model of agency (*Cooperative* and *Free-Agent*), varying the number of mobile agents (*Robots*), and the ratio between their personalities (*Risk Prone* vs *Risk Averse*), i.e. for each variable parameter (*N* and *Personality Ratio*), results were obtained using batches of 50 runs of 300 steps each, as shown in Fig. 2.

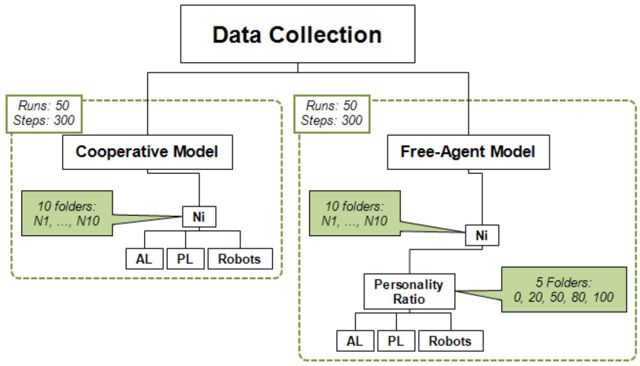

Figure 2. Data Collection Methodology

5. Results

In order to fully comprehend the overall performance of the multi-agent system, two distinct and complementary analyses were devised. The first focused mainly on understanding the impact increasing the number of mobile agents might have on the overall performance. The second is an explicit comparison of the overall system performance between the *Cooperative* and *Free-Agent* models.

The next two sections present the obtained results for both analyses in greater detail.

5.1. 1st Analysis

The present analysis focuses on understanding the impact increasing the number of mobile agents (Robots) might have on the overall system performance. Thus, this section is divided into two parts. The first is dedicated to the *Cooperative Model*, while the second centres on the *Free-Agent Model*.

Since the goal is to evaluate the whole system's performance, each section is subdivided into subsections dedicated to each agent.

1) Cooperative Model:

Assembly Lines: Regarding assembly lines (AL), there are two attributes of major importance for evaluating the system's performance. Namely, the *Received Products* and the *Waiting Time*. Figures 3 and 4 present the variation of those attributes with increasing number of Robots.

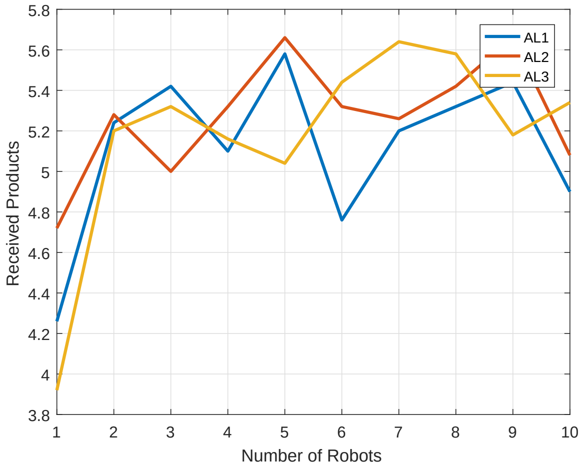

Figure 3. Assembly lines received products with increasing number of Robots.

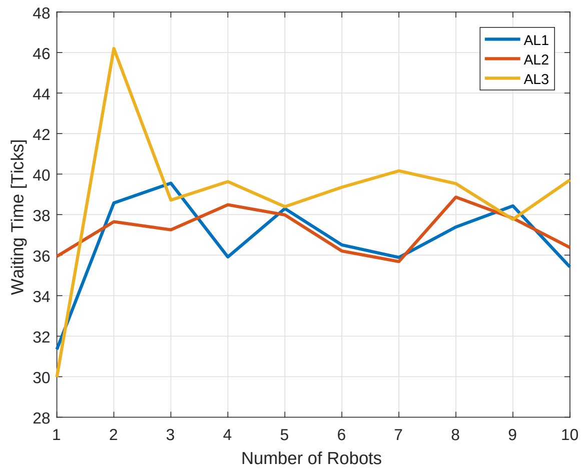

Figure 4. Assembly lines average waiting time with increasing number of Robots.

It is clear the increasing number of Robots results in an increase of received products. On the other hand, the waiting time seems to tend to an average value of 38 to 40 ticks (see Fig.4), which makes sense if one takes into account the geographical disposition of the Robots' dock, the PL and AL. These last two are 20 cells apart, and the dock is located on the mid-right side of the grid. Therefore, the distance an agent needs to cover to complete a task can be estimated using the following expression $d = 20 + e$, where the e refers to statistical variation that results from the fact that sometimes agents are assigned to a task as soon as they deliver a product. An e with a value close to 20 suggests that agents are usually far away from the PL when they are assigned to a task.

Production Lines: Analogously, for production lines (PL) a way of evaluating the system performance is by focusing on the *Idle Time* and the *Queue Time*.

Figures 5 and 6 present the variation of those attributes with increasing number of *Robots*.

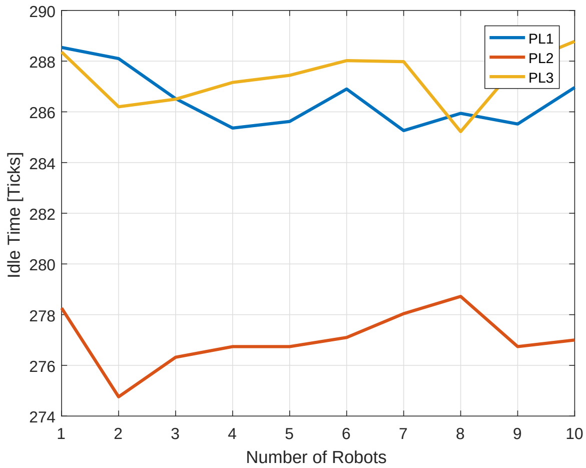

Figure 5. Production lines average idle time with increasing number of Robots.

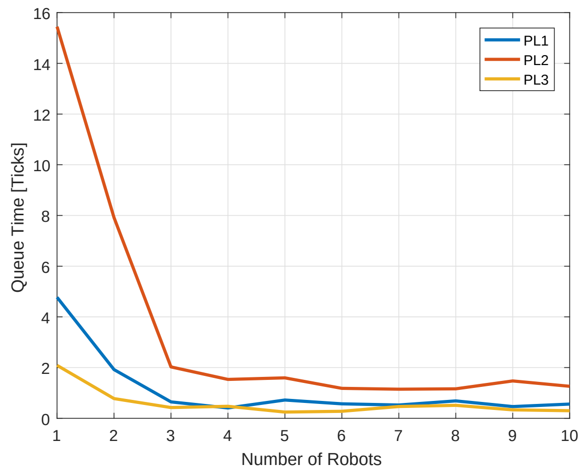

Figure 6. Production lines average queue time with increasing number of Robots.

These results clearly show the queue time decreases with the increasing number of Robots. This makes sense if one takes into account that PL stop their production when there is no space available to deposit its products. The limitation of space is a clear bottleneck with great consequences to the overall system performance. It seems that for the demand distribution used throughout the simulations the optimal number of Robots is 4, which is consistent with the previous results.

Robots: Making use of the *Gini coefficient* (refer to Appendix A for further information), it is possible to measure the wealth inequality of the Robots' population.

As can be seen from Fig. 7 the inequality increases with increasing number of Robots. This result suggests that even if agents do not actively pursue an egotistic interest, wealth inequality is bound to occur in a system like the one under analysis. This subject will be revisited and further developed on subsection 5.12.

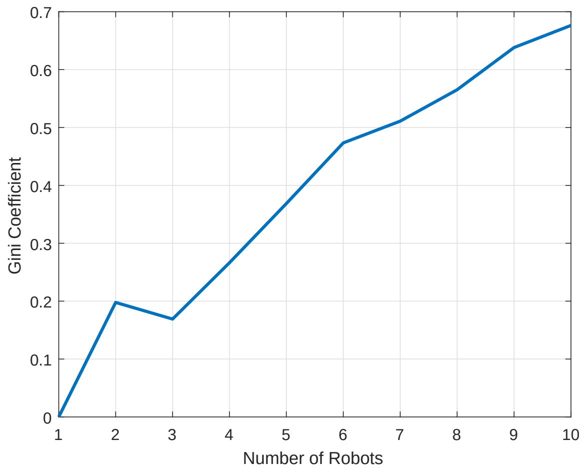

Figure 7. Robots' Gini coefficient.

2) Free-Agent Model: Regarding the *Free-Agent Model*, some modifications to the previous fashion of results presentation had to be taken into account, for the sake of brevity and simplicity.

Since this model of agency makes use of another variable (*Ratio of Personalities*), the average values will be presented instead of presenting the discriminate results for each one of the available ratio combinations. This solution has yet the advantage of being possible to evaluate differences between different values of *Ratio of Personalities*, while seeing the impact of increasing the number of Robots on the overall system's performance.

Assembly Lines: Figures 8 and 9 present the results related to the *Received Products* and the *Waiting Time* attributes for all assembly lines.

As expected, as the number of Robots increase so does the number of received products, regardless of the agents' personality. Moreover, the waiting time tends to a value between 38 and 40 ticks.

Production Lines: Figures 10 and 11 present the variation of the *Idle Time* and the *Queue Time* attributes with increasing number of Robots, for all production lines.

It is clear from the results both the idle and queue time decrease with the increase of Robots. This results are complementary, since by definition a PL stops working if there is no space available to deposit the produced items. Thus, these two metrics are thus intrinsically linked and they should be expected to have a similar trend of behaviour. Once again, the optimal number of Robots for the particular demand distribution used throughout the simulations seems to be 4.

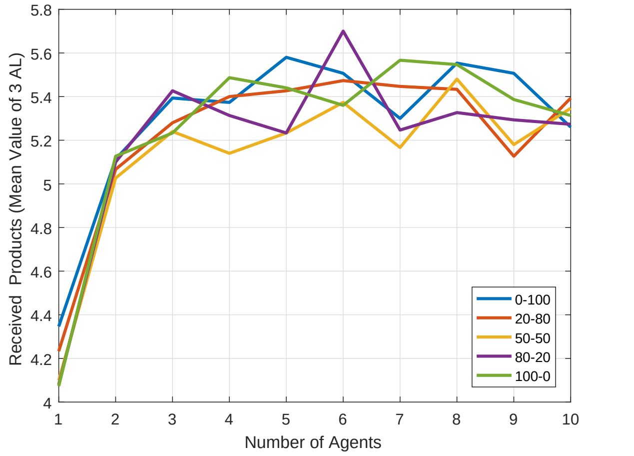

Figure 8. Assembly lines' average received products with increasing number of Robots, for all combinations of *Ratio of Personalities*.

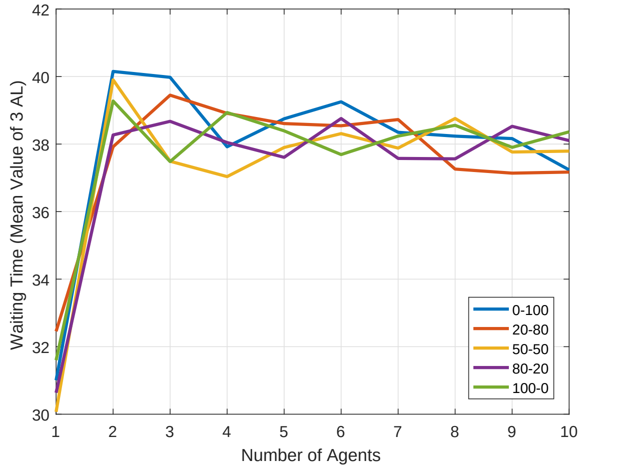

Figure 9. Assembly lines' average waiting time with increasing number of Robots, for all combinations of *Ratio of Personalities*.

Robots: Figure 12 present the *Gini coefficient* variation.

Once again the results show wealth inequality increases with the increasing number of Robots. Moreover, this inequality seems to be independent of both the model of agency and the willingness agents have to take risks, which means it is the system itself that is promoting this inequality.

From an Economics perspective, a market system that rewards individuals based on their contributions in producing society's output is prone to an income inequality, since its fundamental drive mechanism is discrimination [8]. Particularizing for the problem at hand, agents are being selected (discriminating) by ability and this ability is increased/reinforced by the same entity that discriminates.

In addition, there is yet another systemic feature that might render the system permissive to high wealth

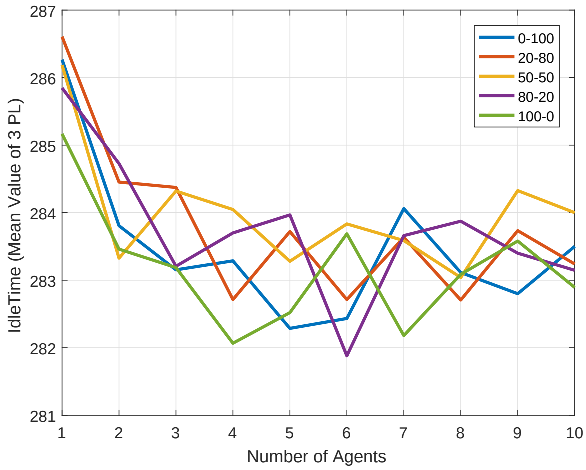

Figure 10. Production lines average idle time with increasing number of Robots, for all combinations of *Ratio of Personalities*.

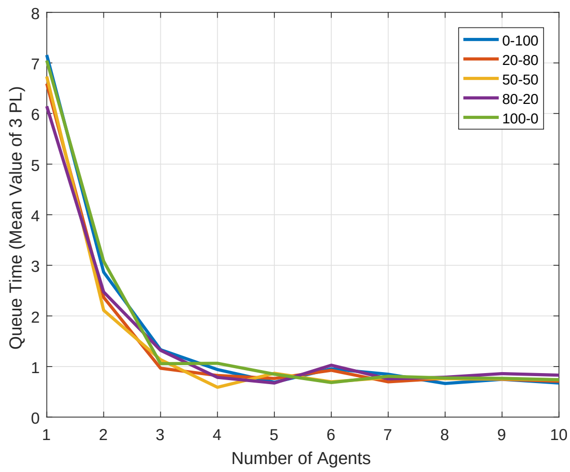

Figure 11. Production lines average queue time with increasing number of Robots, for all combinations of *Ratio of Personalities*.

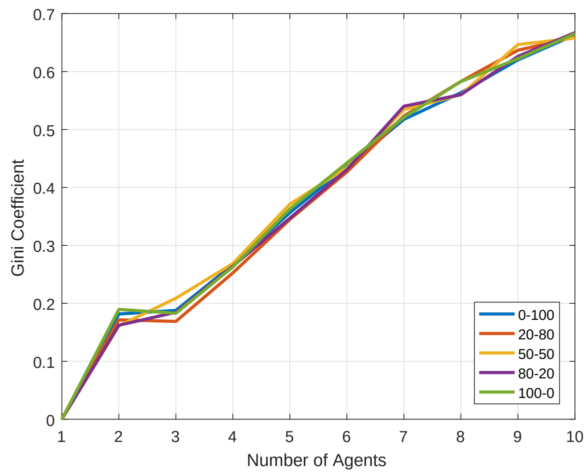

Figure 12. Robots' Gini coefficient, for all combinations of *Ratio of Personalities*.

inequality. Namely, the fact that this is not a closed economy, in the sense that the amount monetary units is not conserved. This scenario is analogous to a financial system that allows credit without accounting for the risk liability, creating a sense of ever growing value market [4, 8].

It should however be noted that, although wealth inequality is a real world problem that must of course be tackled, the author does not suggest these results can or should be regarded as an economic argument. These results only show the task assignment mechanism devised for the problem herein subject to analysis does not promote a fair and balanced use of the available resources. This conclusion, although unexpected, is of major importance, specially when considering a real world application of such a system, since it will cause some Robots to wear faster than others, which in turn will require a much more careful maintenance strategy.

5.2. 2ND Analysis

While the previous analysis focused on each of the models of agency independently, this one lays an explicit comparison of the overall system performance between the *Cooperative* and *Free-Agent* models.

To do so, two different, critical situations were taken into consideration: one in which there was excess and another in which there was scarcity of resources, namely Robots to transport the products.

From the previous analysis is clear that there is excess of Robots when there are more than 4, and scarcity when there are 1 or 2 mobile agents available.

Since the goal is to evaluate which model of agency is better without disregarding the whole system's performance, the situations under analysis throughout this section are: 1) when there are 10 Robots, and 2) when there are only 2 Robots.

At the end of this analysis it will presented a summary of the results for better and clear understanding.

1) Excess of Resources (N = 10):

Assembly Lines: For the sake of brevity and simplicity, since the system performance follows the same trend using different *Free-Agent Model* agents, the following plots will focus on presenting results of the *Cooperative Model* juxtaposed with the *Free-Agent Model* using a population composed of 50% *Risk Prone* and 50% *Risk Averse* agents.

Figures 13 and 14 show the amount of *Received Products* and *Waiting Time* for each AL, respectively.

There is, in this case a slight advantage to resort to the *Free-Agent Model*, since the amount of *Received Products* is higher and there is no substantial difference in *Waiting Time*.

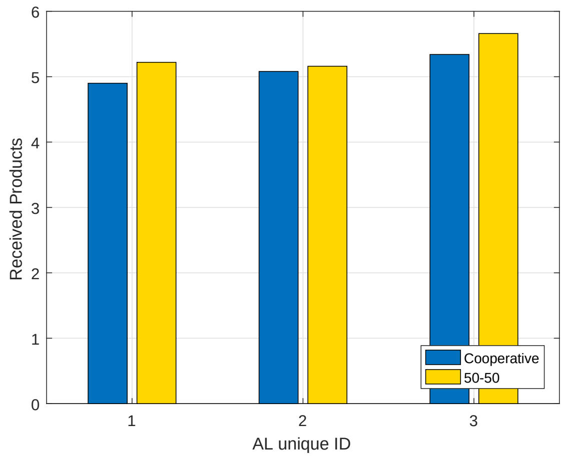

FIGURE 13. Assembly lines received products with excess of resources – N=10.

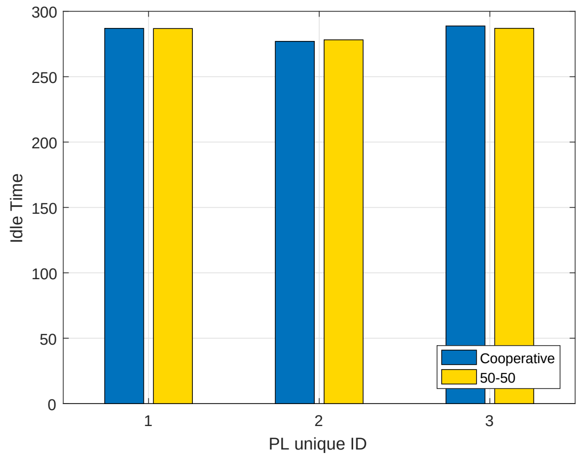

FIGURE 15. Production lines' average idle time with excess of resources – N=10.

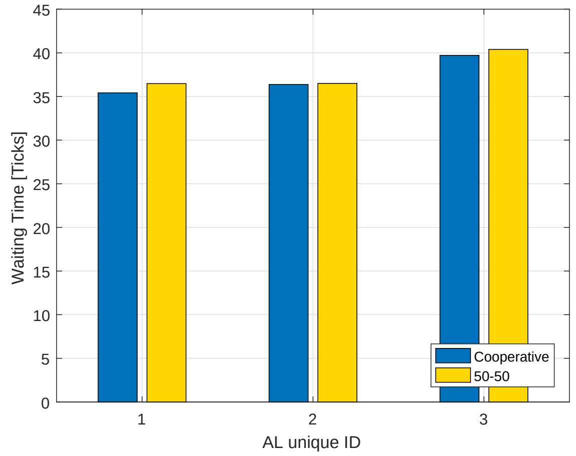

FIGURE 14. Assembly lines average waiting time with excess of resources – N=10.

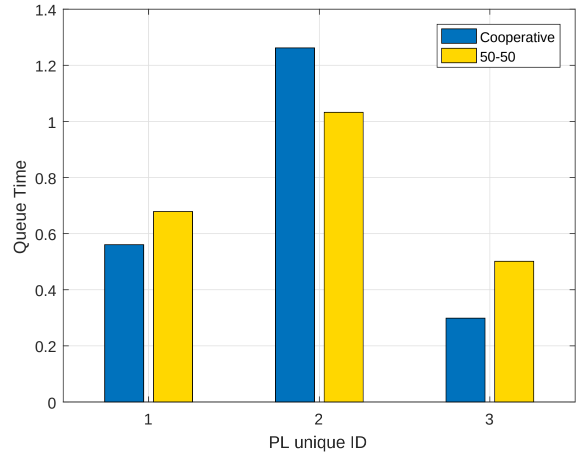

FIGURE 16. Production lines' average queue time with excess of resources – N=10.

Production Lines: Adopting the same fashion of showing results, from figures 15 and 16 one can see the total *Idle Time* and *Queue Time* for each PL, respectively.

Attending to the fact that the *Queue Time* is in general shorter when using cooperative agents, it suggests this model of agency is more suitable to a situation where one wants to minimize this index in a situation of excess of Robots, since there is no clear difference regarding the *Idle Time*.

Robots: In order to have a clear picture of the wealth distribution of these agents refer to figure 17, where it is presented the wealth of all Robots when using different models of agency. In addition, table 1 presents the Gini coefficient for each model.

As can be seen, the problem of the task assignment unbalance persists, as expected. And since it is a systemic problem, no further comments will be made about this subject.

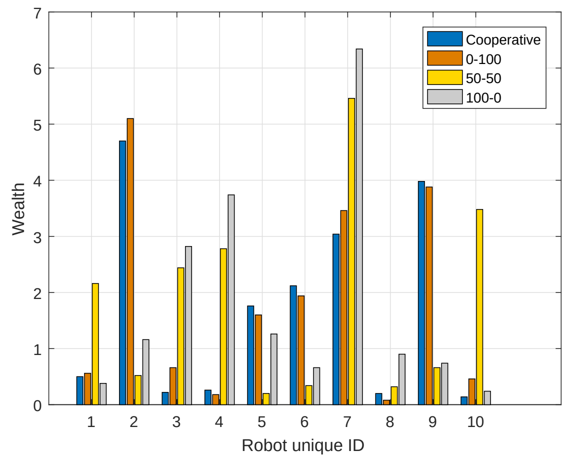

FIGURE 17. Robots' average wealth in a situation with excess of resources – N=10.

2) Scarcity of Resources (N = 2):

Table 1
Gini coefficient for a population of 10 agents with different types of personality, in a situation with scarcity of resources – N=10.

Cooperative	0-100	50-50	100-0
0.6763	0.6632	0.6578	0.6653

Assembly Lines: Analogously to what was done for the previous situation, figures 18 and 19 present the total *Received Products* and *Waiting Time* for each AL, respectively.

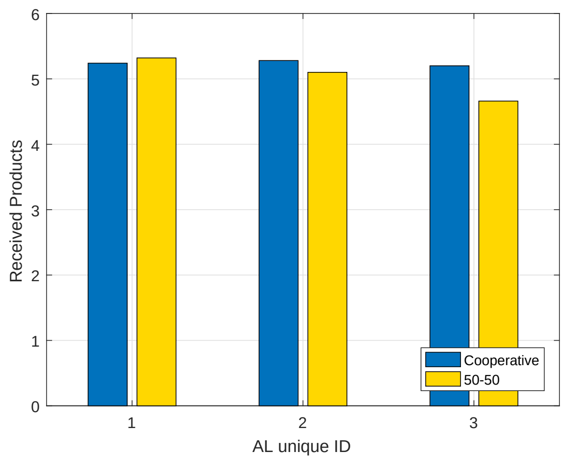

Figure 18. Assembly lines received products with scarcity of resources – N=02.

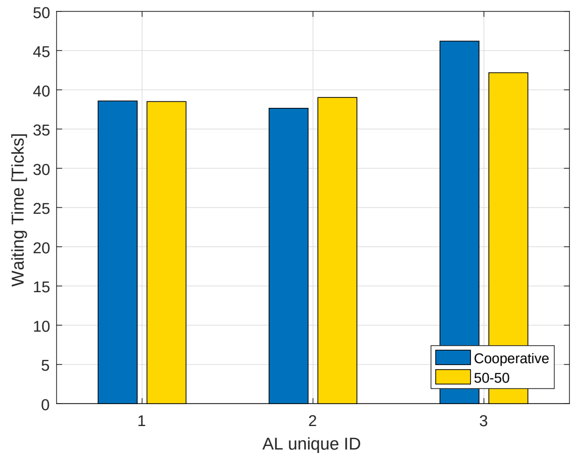

Figure 19. Assembly lines average waiting time with scarcity of resources – N=02.

As can be seen, there is no substantial difference in either of the performance indices.

Production Lines: Regarding PL, figures 20 and 21 seem to suggest a clear advantage of employing the *Free-Agent Model* in a situation of scarcity of Robots, since the resulting *Queue Time* is substantially shorter.

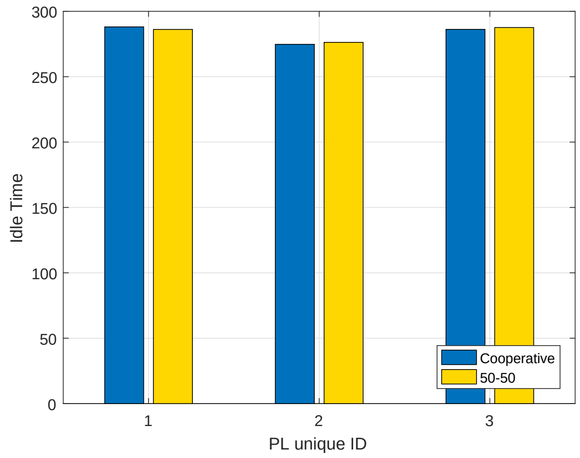

Figure 20. Production lines' average idle time with scarcity of resources – N=02.

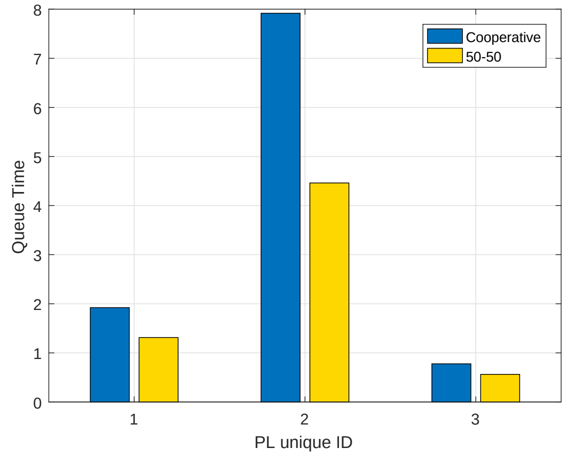

Figure 21. Production lines' average queue time with scarcity of resources – N=02.

Regarding the results, there is a clear advantage of resorting to the *Free-Agent Model*, since the *Queue Time* is much shorter and there is no difference regarding *Idle Time*.

Robots: In a situation with so few Robots the amount of wealth are of the order of 10^n, where n ranges from 4 to 12. The fact that there is such a wide span of values, the bar plot is not a suitable choice for results presentation. Instead, it the information regarding the Robots' wealth will be presented in the form of a table, namely table 2. As before, it is also presented the Gini coefficient for the present situation in table 3.

It should be noted, the values presented in tables **??** are computed independently. Table 2 presents the average wealth of a given Robot over the number of simulation *Runs*. While table 3 shows the average of the mean Gini coefficient per simulation *Run*.

The results clearly corroborate the conclusions of the previous section, namely in a situation where only

Table 2
Robots' average wealth for a population of 2 agents with different types of personality, in a situation with scarcity of resources – N=02.

	Cooperative	0 - 100	50 - 50	100 - 0
Robot 1	4.1958e+05	4.3983e+10	1.3526e+04	5.6295e+12
Robot 2	4.3285e+07	4.3938e+05	1.0578e+04	4.3054e+05

Table 3
Gini coefficient for a population of 2 agents with different types of personality, in a situation with scarcity of resources – N=02.

Cooperative	0-100	50-50	100-0
0.1978	0.1816	0.1612	0.1896

a small number of Robots are available they perform a similar number of tasks, i.e. the task assignment is balanced in such a situation.

3) Summary: As mentioned previously the system performance depends heavily on four fundamental indices: *Received Products*, *Waiting Time*, *Idle Time* and *Queue Time*. If one seeks to optimize the overall performance, it is thus required to maximize both the *Received Products* and *Idle Time*, while at the same time minimize the *Waiting* and *Queue Time*.

Tables 4 and 5 present which model of agency should be employed in order to optimize each performance index in a situation of excess or scarcity of Robots, respectively.

Table 4
Results' summary when N = 10

	Received Products	Waiting Time	Idle Time	Queue Time
AL	Free-Agent	Indifferent	—	—
PL	—	—	Indifferent	Cooperative

Table 5
Results' summary when N = 02

	Received Products	Waiting Time	Idle Time	Queue Time
AL	Indifferent	Indifferent	—	—
PL	—	—	Indifferent	Free-Agent

6. Conclusions

The proposed project had three major goals: 1) to evaluate the impact of the number of Robots on the overall system performance; 2) to compare and assess which model of agency (*Cooperative* or *Free-Agent*) would produce better results; 3) to determine which *Ratio of Personalities* should one rely on in order to improve the overall performance.

Regarding the first goal, it was clear from both figures 6 and 11 there is an improvement in performance if the number of Robots is increased. This number, however, is not constant and depends on the demand. For the particular case of the simulations the optimal number of Robots is 4.

Taking into account the results from the 2nd Analysis, one should resort to a *Free-Agent Model* when there is scarcity of Robots, since it tends to minimize the *Queue Time*. If, on the other hand one is in a situation of excess of resources, there is no clear advantage of using either model. (See tables 4 and 5.)

Regarding the *Ratio of Personalities*, it was clear there was no substantial difference in performance by using different personalities in a *Free-Agent Model* framework.

An unexpected result was the verification that Robots' wealth inequality increased with the increasing number of Robots and that it was the system itself that was promoting it. Namely, the task assignment mechanism itself in addition to its rewarding system created an open economy based on discrimination and that did not satisfy the principle of conservation [4, 8]. This results proved the devised task assignment mechanism does not promote a fair and balanced use of the available resources, which might have implications on the maintenance strategy of Robots if a real world application of such a system is considered.

Finally it should be noted the conclusions presented hold only for a particular demand distribution, and that a number of results suggest more simulations should have been performed for faster demands. Namely, the fact that the *Idle Time* is roughly 90% of the simulation time clearly corroborates that hypothesis.

7. Future Work

Different adaptations, tests, and experiments have been left for future due to lack of time. One pressing issue, already stressed in previous sections, was the fact that the system should have been tested for different demand rates, particularly in conditions of faster demand.

To properly evaluate the solution herein proposed, it would be necessary to solve the same problem resorting to standard logistics methods, namely Linear Programming.

It would be of great interest to extend the problem and to force agents to negotiate with one another. A good way of achieving this would be for instance to make one of the products to require two agents to carry it. In such a situation, the agent that won the auction would have to negotiate with other agents in order for the task to be completed.

Different negotiating mechanisms could and should be tested. However the most interesting aspect of this new situation would be the possibility of modelling and testing social behaviours. In particular, it would be very interesting to see if the *indirect reciprocity mechanism* [7] alone would be sufficient to have a population of cooperative agents.

References

[1] Engenharia de produção industrial. http://engenhariadeproducaoindustrial.blogspot.pt/2009/07/metodologia-mizusumashi.html.

[2] Gini coefficient. https://en.wikipedia.org/wiki/Gini_coefficient.

[3] Lean manufacturing. https://en.wikipedia.org/wiki/Lean_manufacturing.

[4] Adrian Dragulescu and Victor M Yakovenko. Statistical mechanics of money. *The European Physical Journal B-Condensed Matter and Complex Systems*, 17(4):723–729, 2000.

[5] Charles Jackson Grayson. *Decisions under uncertainty: Drilling decisions by oil and gas operators*. Ayer, 1960.

[6] Paulo Peças. Lecture notes on Production Management – New Approaches to Production Management – Mizusumashi, 2017-2018.

[7] David G Rand and Martin A Nowak. Human cooperation. *Trends in cognitive sciences*, 17(8):413–425, 2013.

[8] Manuel C Rios, Campbell R McConnell, and Stanley L Brue. *Economics: Principles, problems, and policies*. McGraw-Hill, 2013.

[9] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[10] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.

Appendix

Throughout the document several concepts are used without deeper or explicit definition. The present section tries to mitigate that situation by methodically and extensively define the most important concepts. To do so it will be used the following standard:

Concept, Entity the concept refers to – Concept definition.

The most relevant concepts and respective definitions are listed bellow.

Gini Coefficient, Robot Population – A numerical measure of statistical dispersion, given by $\frac{1}{n}\left(n+1-2\left(\frac{\sum_{i=1}^{n}(n+1-i)y_i}{\sum_{i=1}^{n}y_i}\right)\right)$ [2, 8], herein used as a measure of wealth inequality.

Idle Time, PL – Time during which a PL is not producing products.

Produced Products, PL – Total amount of products produced by one or more PL.

Queue Time, PL – Elapsed time between a request for a particular product and the moment a PL starts producing it.

Ratio of Personalities, Robot – The ratio of *Risk Prone* to *Risk Averse* agents in the population of robots. Throughout this document, this ratio will often be addressed as a tuple *percentage of risk prone agents - percentage of risk averse agents*, e.g. the tuple *0-100* refers to a population composed only by risk prone agents, whereas the tuple *100-0* refers to a totally risk prone population. It should however be noted that this type of agents are characterized by two layers of personality: its willingness to take risks, and its tolerance to risk. For a deeper understanding of these two dimensions, please refer to section 4.24.

Received Products, AL – Total amount of products received by by one or more AL.

Run, all – A batch of simulation steps. For the present project, each Run consisted of 300 steps. Data collection was performed using batches of 50 runs, as mentioned in section 4.4.

Tick, all – Unit of time during simulation time.

Waiting Time, AL – Elapsed time between a request by an AL for a particular product and the satisfaction of that request. This is equivalent to the PL's *Queue Time* plus the transportation time from that PL to the AL that made the request.

Wealth, Robot – Amount of monetary units a PL rewards an agent for having won the auction.