Embedded Games and the Homo-Egualis Principle

Martin Smit

University of Warwick

Abstract. In job scheduling games, each agent wants to minimise the time it takes to finish their own task. When public processors are faster than personal ones we see a tragedy of the commons situation arise. In this paper we attempt to alleviate this by introducing the homo-egualis principle that directly ties agents' utilities together, making them favour more equal outcomes. We show that this principle –comment about approximating socially optimal outcomes– and that –comment about reinforcement learning convergence–.

Keywords: Multi-agent systems, Mechanism design, Queuing theory, Multi-agent reinforcement learning

1 Introduction

Something about servers sending information over a network wanting to choose equilibria that maximise their own efficiency. Common example is of printers in an office, public resource depleted, inefficient if everyone is greedy.

2 Model

Consider a multi-agent system in which each agent receives jobs and must choose to send them either to a slower private processor, or a faster public processor. Assume the arrival of jobs of size α to agent i (which can be fixed or drawn from a probability distribution) is a Poisson process of rate λ_i, personal processors complete jobs at some rate $\mu(\alpha)$, shared processors complete jobs at some rate $\nu(\alpha)$ where μ and ν are (not necessarily linear) functions of α, and agents do *not* observe the state (i.e. number of jobs beyond what it has already sent and time until free) of the shared processor.

The agents observe the entire system at fixed intervals. After observing they choose the strategy that they will play for the duration of the next interval. The strategy space of an agent is the unit interval, and a strategy is the probability of sending a job to the public processor.

We initially assume that every utilisation ratio (ρ, representing the ratio of arrivals to services) is less than 1 for every value of p and q i.e. every server will have an equilibrium distribution and its queue length will not explode to infinity. Mathematically our assumptions become

$$\lambda_1 \, p \leq \mu(\alpha) \tag{1}$$
$$\lambda_2 \, q \leq \mu(\alpha) \tag{2}$$
$$\lambda_1 \, p + \lambda_2 \, q \leq \nu(\alpha) \tag{3}$$

for all values of p and q.

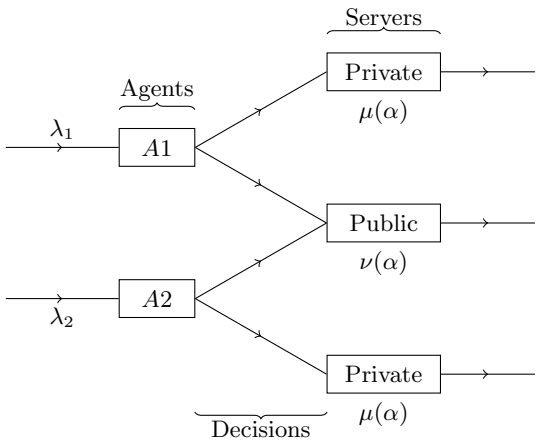


Fig. 1. The network realised by 2 players

3 Equilibria

We initially consider a 2 player game. We present the players' cost function as the following:

$$f(p) = p \, \mathbb{E}(W_{pu}) + (1 - p) \, \mathbb{E}(W_{pr}) \tag{4}$$
$$= \frac{p}{\nu(\alpha) - \lambda_1 \, p - \lambda_2 \, q} + \frac{1 - p}{\mu(\alpha) - \lambda_1 \, (1 - p)} \tag{5}$$

where W_{pr} and W_{pu} are the expected waiting time in the private and public queues respectively, and p and q are the respective players' probability of sending an incoming job to the public processor. An alternative was to take the maximum of either waiting time, but this neglects to take into account that players would care more about the speed of the processor that they are sending the majority of their jobs to.

Proposition 1. *Given the previous setup, the probability for player 1 that minimises (5) is equal to*

$$p^* = \frac{(\lambda_1 - 2\,\mu)\,(\nu - \lambda_2\,q) + (\lambda_1 + \lambda_2\,q - \mu - \nu)\,\sqrt{\mu\,(\nu - \lambda_2\,q)}}{\lambda_1\,(\mu - \nu - \lambda_2\,q)}$$

under the assumption that an interior solution exists, where q is the optimal probability for player 2, and μ and ν written without the α is an abuse of notation and implicitly are functions of α.

Proof. Taking the derivative of the cost function we find

$$\frac{df}{dp} = \frac{\nu - \lambda_2\,q}{(\lambda_1\,p - \nu + \lambda_2\,q)^2} - \frac{\mu}{(\mu - \lambda_1 + \lambda_1\,p)^2}$$

which has stationary points at

$$p^* = \frac{(\lambda_1 - 2\,\mu)\,(\nu - \lambda_2\,q) \pm (\lambda_1 + \lambda_2\,q - \mu - \nu)\,\sqrt{\mu\,(\nu - \lambda_2\,q)}}{\lambda_1\,(\mu - \nu - \lambda_2\,q)}$$

Using the second derivative of f it can be shown that the "plus" solution is always a local minimum and the "minus" solution is always a local maximum. □

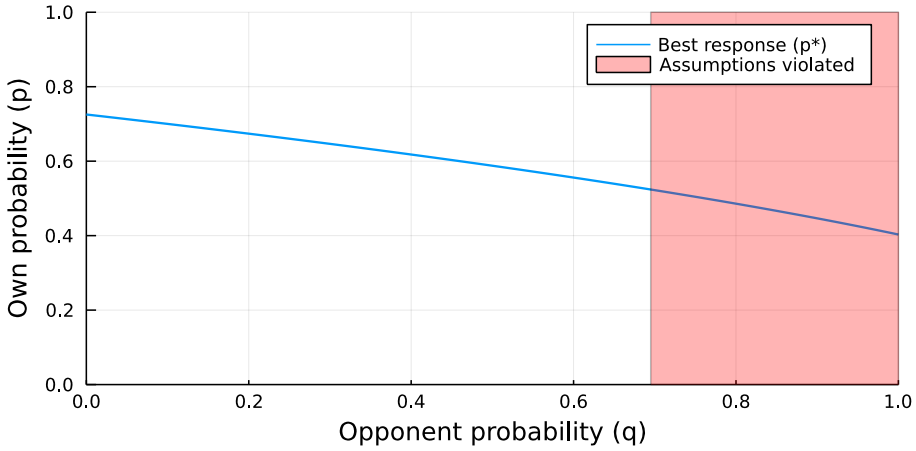


Fig. 2. A demonstration of how the best response function ($p*$) looks when player 1's arrival rate is higher than that of player 2. Note that after a certain level of q, there is no value of $p*$ that allows player 1 to be under both its private and public limits on the rate of arrival.

In a setting of hyper-rationality and complete information we would expect that players would play symmetrically ex-ante the revelation of their type i.e.

they would go into the game with the same strategy in mind before their arrival rate λ_i is revealed to them. Hence, by the same logic as above we can also derive that

$$q^* = \frac{(\lambda_2 - 2\,\mu)\,(\nu - \lambda_1\,p) + (\lambda_2 + \lambda_1\,p - \mu - \nu)\,\sqrt{\mu\,(\nu - \lambda_1\,p)}}{\lambda_2\,(\mu - \nu - \lambda_1\,p)}.$$

We now have an expression for each player's best response to the other player's action. This means that we can construct a recursive sequence of best responses by continuously substituting the equations into one another, essentially allowing players to best respond ad infinitum. This very quickly leads to both players converging to the Nash Equilibrium.

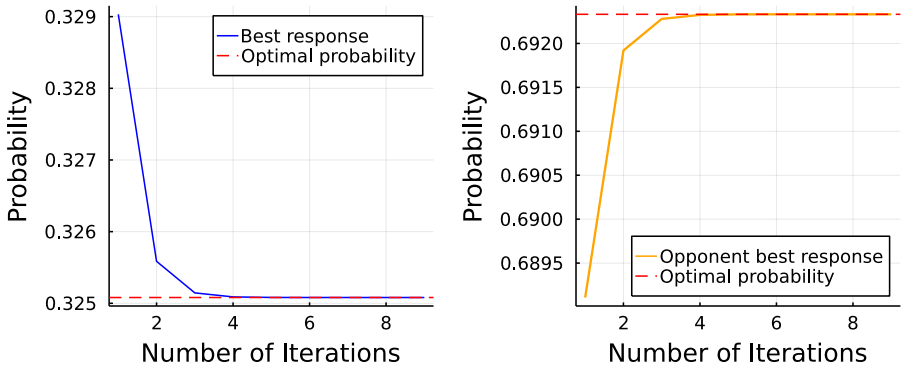


Fig. 3. The convergence of best responses to the equilibrium happens after only a few iterations.

3.1 Notes on p^*

It should be noted that due to the square root in the formula for p^*, the value provided by the formula can have complex values. Mathematically, this corresponds to the fact that for these inputs there is no interior optimal solution, as there is no value which makes the gradient zero. This means that the optimal value for p will be on the boundary, either 0 or 1.

Additionally, we can see that the denominator will always be negative. This is due to the fact that μ is assumed to be less than ν. Due to this, we would expect the numerator's terms to also be negative. The numerator contains the terms $\lambda_1 - 2\,\mu$ and $\nu - \lambda_2\,q^*$. The first of these terms is negative and the second is positive by assumption. Similarly, the second multiplication term is

$$(\lambda_1 + \lambda_2\,q^* - \mu - \nu)\,\sqrt{\mu\,(\nu - \lambda_2\,q^*)}$$

which again has the second term positive (and not complex as discussed before), and the first will be negative by assumption. This means that both of the multiplications in the numerator produce negative numbers which are added together to produce something negative. As both the numerator and denominator are always negative as long as the assumptions on $\lambda_1, \lambda_2, \mu$, and ν are met.

3.2 Effect of altering α for different functions μ and ν:

We assume all functions go through the origin as we would expect a job of 0 size to take 0 time to complete. In this section we assume $\lambda_i = 1$ and fix player 2's strategy arbitrarily at $q = 0.6$. If we let both functions be linear and alter α we get the following for the optimal value of p:

4 TODO

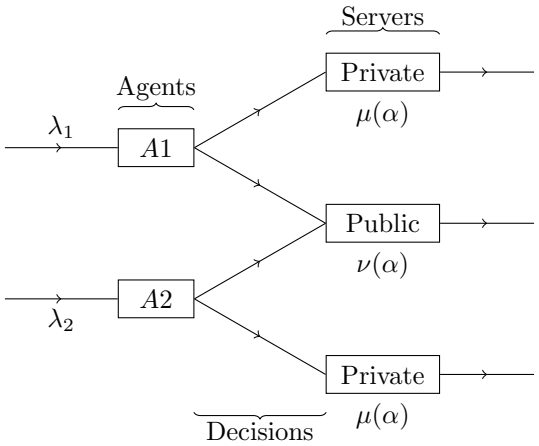
- Calculate socially optimal solution,
- Price of anarchy

5 Mathematical Setup

To me, there are two different but connected processes taking place: the continuous time queue and the discrete time game. As in Ann's paper on the homo-egualis principle, the game is constructed by taking the queue and condense it into a static repeated game then apply established techniques and theories from game theory. There is also the possibility of directly studying the original queue itself and applying techniques from queueing theory. It would be interesting to find a neat way to tie results from one side to the other. I explain later a bit about how this connection would work.

Simultaneous: I'm not entirely sure what this constitutes? What would be the use case and why is it not modelled instead by the continuous time setup?

Continuous time: In a continuous time structure, the model is that of a network of queues. Assume the arrival of jobs of fixed size α to agent i is a poisson process of rate λ_i, personal processors complete jobs at some rate $\mu(\alpha)$, shared processors complete jobs at some rate $\nu(\alpha)$, and agents do *not* the state (i.e. number of jobs beyond what it has already sent and time until free) of the shared processor. The strategy space of an agent is a decision process to determine where to send an arriving job given the current (unknown) state of the system.



Questions to explore:

- What is the efficiency of a random/mixed strategy? (Mixed or pure strategies transform this into a system of M/M/1 queues which have neat formulae for expected queue length, waiting time, and time spend in the queue)
- What is the optimal strategy for individual agents to minimise the time that their jobs spend in the system?
- How can the homo-egualis principle be introduced into this system and how does this affect optimal strategies?

- What is the socially optimal strategy for the agents?
- Compare the three previous strategies in terms of the utility difference for the best and worst off agents in each case, and overall utility. (What is the price of anarchy? What is the price/benefit of information of the system?)
- How does all of the previous answers depend on the size of the jobs? If we change the size of the jobs does this significantly alter the strategies? Is this alteration predictable or chaotic?
- Can learning agents be incentivised to converge to the strategies previously derived?

6 Proposed Model:

Communication could be done at certain intervals or perhaps if one agent becomes overloaded or reaches a threshold it could communicate this to others. We could the discretise the events by setting a point of communication as the "jump" event in the embedded discrete time stochastic network. Using these discrete times, players can update their utility functions using the homo-egualis principle and choose the strategy they will perform in the next block of time.

In many cases there are nice closed form solutions for properties of queues such as expected waiting times in equilibrium and reversibility criteria to give the rate of the output. These criteria are met if everyone is playing a mixed strategy, hence we can immediately evaluate these particular strategies and perhaps compare anything else against them.

7 Brief Lit Review

Queuing theory combined with agent based modelling has been studied since 1971 with [4], but the decisions made were generally those of customers joining queues.

- More recently, in [2] the agents making decisions are the jobs themselves and not the servers that want to optimise the jobs. This is also the case in [3], where agents are customers and are guided by external queue managers to influence their decisions.
- In [1] the focus of the RL is about optimising the pathway that a customer takes through a network and what order to serve arrivals, which is not really the case for this network as the network structure is predetermined by the number of agents, only one routing decision must be made, and this is the only decision an agent makes. Additionally, the route of a customer in this paper cannot be updated on the fly, which is not the case for our model.
- I couldn't find any paper where the servers themselves were the ones making the decisions. I'm not sure if I just don't have the keyword necessary to search for it.

References

1. Daniel Barry Fuller, Edilson Fernandes de Arruda & Virgílio José Martins Ferreira Filho (2020) Learning-agent-based simulation for queue network systems, Journal of the Operational Research Society, 71:11, 1723-1739, DOI: 10.1080/01605682.2019.1633232
2. Sankaranarayanan, Karthik & Delgado, Carlos & Larsen, Erik & Ackere, Ann. (2012). Behavioral Queueing: An Agent Based Modeling Approach. International Journal of Modeling and Optimization. 408-412. 10.7763/IJMO.2012.V2.153.
3. Zhang Q. (2011) Multi-agent Based Supermarket Queuing Model and Optimization. In: Jin D., Lin S. (eds) Advances in Multimedia, Software Engineering and Computing Vol.2. Advances in Intelligent and Soft Computing, vol 129. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-25986-9_99
4. Uri Yechiali On Optimal Balking Rules and Toll Charges in the GI/M/1 Queuing Process. Operations Research 19 (2) 349-370 https://doi.org/10.1287/opre.19.2.349