Non-symmetric Preferences in the IPA Market with Reinforcement Learning

Eduardo Rodrigues Gomes and Ryszard Kowalczyk
Swinburne University of Technology
Faculty of Information and Communication Technology
Hawthorn, 3122 Victoria, Australia
{egomes, rkowalczyk}@ict.swin.edu.au

Abstract

Machine Learning has been proposed to support and optimize market-based resource allocation. In particular, Reinforcement Learning (RL) has been used to improve the allocation in terms of the utility received by resource requesting agents in the Iterative Price Adjustment (IPA) mechanism. In such an approach, utility functions describe the agents' preferences for resource attributes and are the basis for RL to learn demand functions that are optimized for the market. It has been shown that the reward functions based on the individual utility of the agents and the social welfare of the allocation can deliver similar social results when the market consists only of learning agents with symmetric preferences. In this paper we investigate the IPA market-based resource allocation with RL for the case of agents with non-symmetric preferences. We show through experimental investigation that the results observed above are also held in this case. In particular, we show that the individual-based reward function is able to approximate the solution to the fairest Pareto-Optimal allocation in situations where the social-based reward function fails.

1 Introduction

An efficient allocation of resources is one of the main challenges faced by large-scale distributed systems, such as the Grid and service-oriented architectures. Several features complicate the problem. Computational and geographical distribution, dynamic architecture, lack of coherent global knowledge and lack of centralized control are just some of them. Current research in this area points to market-based resource allocation mechanisms as a promising approach [2, 12, 17].

This paper considers the Iterative Price Adjustment (IPA) mechanism [4]. The IPA is a *tâtonnement*-like pricing mechanism that can be used in commodity-market resource allocation systems. Pricing mechanisms are responsible for defining the price level at which resources will be traded. In the IPA, the price is adjusted iteratively. The price is increased if demand exceeds the supply and decreased otherwise.

Gomes & Kowalczyk [6] have investigated the IPA in the scenario where the agents use utility functions to describe their preferences over resource attributes and learn the demand functions optimized for the market by Reinforcement Learning (RL) [14]. The reward functions used during the learning process are based either on the individual utility of the agents or the Social Welfare (SW) resulting from the final allocation. An interesting outcome revealed by that investigation is that both reward functions can deliver similar social results when the market consists only of learning agents. Such a result is potentially important in domains where the social utility should be maximized but the agents are unwilling to reveal private preferences. The authors, however, have evaluated only the case of agents with *symmetric preferences*[1], i.e. when the agents use the same utility functions, generating a symmetric payoff table for the game.

In this paper we investigate the IPA market-based resource allocation with RL for the case of agents with non-symmetric preferences. We show through experimental investigation that the results observed above hold for this case. In particular, we show that the individual-based reward function is able to approximate the solution to the fairest Pareto-Optimal allocation in situations where the social-based reward function fails.

The next section presents the background information on the IPA Market with RL. Section 3 shows the setup and results of the experimental investigation. Section 4 discusses some related works. Finally, Section 5 concludes the paper and presents some future directions.

[1] Different meanings for this term can be found on the Economics literature.

2 Iterative Price Adjustment with Reinforcement Learning

We address the scenario in which a limited amount of resources has to be allocated to a set of self-interested agents in a commodity-market resource allocation system using the IPA mechanism. The IPA decomposes the resource allocation optimization problem into smaller and easier sub-problems. Its behaviour mimics the law of demand and supply. The price is increased if the demand exceeds the supply and decreased otherwise. The process is a cycle that begins with a facilitator (the market) announcing the initial prices for the resources. Based on this information, the agents decide on the demand requests that maximize their private utilities (the sub-problems) and send these values to the facilitator. The facilitator adjusts the prices according to the total demand received and announces the new prices. The new prices are calculated using the following formula: $p_i(t+1) = \max\{0,\, p_i(t) + \alpha(\sum_{j=1}^{n} d_{i,j}(t) - C_i)\}$, where $p_i(t)$ is the price of the resource i at time t, $d_{i,j}(t)$ is the demand request of the agent j for the resource i, C_i is the total supply of the resource i, and α is a constant. The process continues until an equilibrium price is reached, when the resources are sold. At the equilibrium, the total demand equals the supply or the price of the excessive supply is zero. Under some circumstances, the equilibrium price may not exist [7], but that problem is out of the scope of this paper.

It should be noted that the agent's utility maximization task in the IPA is actually the maximization of its instantaneous profit. As described in [16], the agent has a revenue function and a cost function over the resources and, at each time step, its task is to find the demand request that maximizes the difference between these two functions given the current price. The result is the existence of a demand function, in which all the points lying in the characteristic curve are equally preferred by the agent. This approach is easy to be understood and quite reasonable. However, it limits the power of the agents because the agents cannot express preferences over different attributes of the allocation, in particular over the price and the amount of resources. It follows that they cannot develop a strategic behaviour to influence the mechanism, which is particularly interesting in small markets where the possible gains from strategic attempts are larger.

The approach investigated in this paper solves the problem identified above proposing the scenario in which agents use utility functions to describe their preferences in the allocation and learn demand functions optimized for the market by RL. The idea is that, given a particular market configuration, there is at least one demand function that maximizes the utility obtained from the aggregation of the agent's utility functions. Learning such a demand function leads to the development of the agent's strategic behaviour.

RL agents learn how to map states of the environment to actions so as to maximize a numerical reward signal. *Q*-learning [15] is probably the most common algorithm for RL. It is simple and easy to implement. In this algorithm, the agent maintains a table of *Q*(s, a)-values that are updated as it gathers more experience in the environment. In our case the environment states *s* represent the resource prices *p* and the actions *a* are the demand requests *d* of the agents, thus *Q*(p, d). *Q*-values are estimations of *Q**(p, d)-values, which are the sum of the immediate reward r obtained by requesting demand *d* at price *p* and the total discounted expected future rewards obtained by following the optimal policy (demand function) thereafter. By updating *Q*(p, d), the agent eventually makes it converge to *Q**(p, d). The optimal demand function π^* is then followed by selecting the actions where the *Q**-values are maximum. Q-values are updated using $Q(p,d) = Q(p,d) + \alpha(r(p,d) + \gamma \max_{d'} Q(p',d') - Q(p,d))$, where $\alpha \in]0,1[$ is the learning rate and $\gamma \in]0,1[$ is the discount rate for infinite horizon problems.

An important component of Q-learning is the action selection mechanism. It is used to harmonize the trade-off between exploration and exploitation. We use the ϵ-greedy method [14], which selects a random action with probability ϵ and the greedy, the one that is currently the best, with probability 1-ϵ.

The application of RL in the IPA changes the objective of the agents in the resource allocation. Instead of maximizing their private utility in an immediate fashion, they now have to maximize the accumulated reward: $\sum_{t=0}^{\infty} \gamma^t r_t$, where r is the reward given by the reward function in use and $\gamma \in]0,1[$ is the discount rate. The two reward functions evaluated in this paper are presented in Section 3.1.

3 Learning the IPA Market

This section presents the investigation on the addressed scenario. We first describe the experiments' setup and then discuss the results found.

3.1 Experimental Setup

We consider a single IPA market with one type of resource, e.g. memory, and two agents. The agents have preferences over the price and the amount of resources. They are described using a utility function for each attribute, $UP(p)$ for the price and $UM(m)$ for the amount of resource. The total utility of an agent is given by the product of its utility functions, $U(p,m) = UP(p) * UM(m)$. The product is used to express the existence of dependency between the attributes.

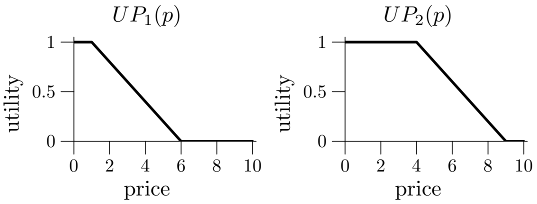

Figure 1: Utility functions for price.

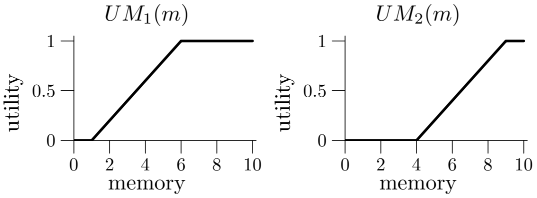

Figure 2: Utility functions for amount of resource.

Table 1: Experiments' configuration.

Type	Agent 1		Agent 2	
A	Wants same	(UM_1)	Wants same	(UM_1)
	Pays same	(UP_1)	Pays same	(UP_1)
B	Wants same	(UM_1)	Wants same	(UM_1)
	Pays less	(UP_1)	Pays more	(UP_2)
C	Wants less	(UM_1)	Wants more	(UM_2)
	Pays same	(UP_1)	Pays same	(UP_1)
D	Wants less	(UM_1)	Wants more	(UM_2)
	Pays less	(UP_1)	Pays more	(UP_2)
E	Wants less	(UM_1)	Wants more	(UM_2)
	Pays more	(UP_2)	Pays less	(UP_1)

In contrast to [6], we focus on the case of agents with non-symmetric preferences. For this, we define 2 different utility functions for each attribute, the price and the amount of resource, and perform experiments using the combinations between them. Figures 1 and 2 show the utility functions. Table 1 presents the configurations of the experiments.

Two different reward schemas are applied: a local reward function based on the individual utilities of the agents and a global reward function based on the SW of the allocation. In the individual reward function, the agents receive a positive reward given by $U(p, m)$ when the market reaches an equilibrium state, and zero when it reaches other states. In the social reward function, the agents receive a reward equal to the SW of the allocation for the equilibrium states, and zero for the others. Note that the multiagent learning problem generated is competitive for the individual reward function and collaborative for the social reward function. The SW is calculated using the Nash Product (NP) function, which is given by the product of the individual utility of the agents: $NP = \prod_{i=1}^{n} U_i$, where U_i is the utility of agent i. The NP is suitable for the resource allocation domain because it encourages both the balance and the improvement of the utility of the agents.

The market was set with 5 units of resources per agent. From the analysis of the utility functions, we can note that such an amount does not allow for all the agents to have a complete satisfaction in the allocation, but it permits the analysis of the market under a condition of limited supply, which is the most interesting case.

During the experiments, the price (states) and the demand requests (actions) were bounded in $[0, 1]$. *Q*-learning formally relies on discrete sets, so both the price and the demand requests were rounded to 1 decimal place. Therefore, the market had 101 possible states and each agent had 101 actions to choose from. In the IPA market, the only available information the agents have is the current price of the resources. It means that they do not know the actions taken and the rewards received by the other agents.

We ran 20 learning experiments for each configuration and reward function. Each experiment consisted of 500 000 learning episodes. The learning parameters were set to $\alpha = 0.1$, $\gamma = 0.95$ and $\epsilon = 0.35$, and the market constant set to $\alpha = 0.05$.

3.2 Experimental Results

We evaluated the quality of the learnt demand functions from two perspectives: the individual utility received by the agents and the SW of the market. The evaluation was made with the trends of the actual demand functions learnt by the agents. One of the reasons for using the trends was that we transformed prices (states) and demands (actions) into discrete sets to implement the learning algorithm. However, such a discretization may lead to small losses of economical efficiency in the market. The other reason was that by using the trends we can avoid local instabilities present in the learnt demand functions [6]. The trends were obtained by a process of curve-fitting using the Sigmoidal-Boltzmann model $y = a + \frac{b-a}{1+e^{-\frac{x-c}{d}}}$. We selected this model based on the visual analysis of the learnt demand functions, which presented a sigmoidal trend. Demand functions were extracted and evaluated in intervals of 5000 learning episodes.

3.2.1 Social Level

We first present and discuss the results from the social perspective. Figure 3 shows the comparison of the median SW of the market over the learning episodes for both reward

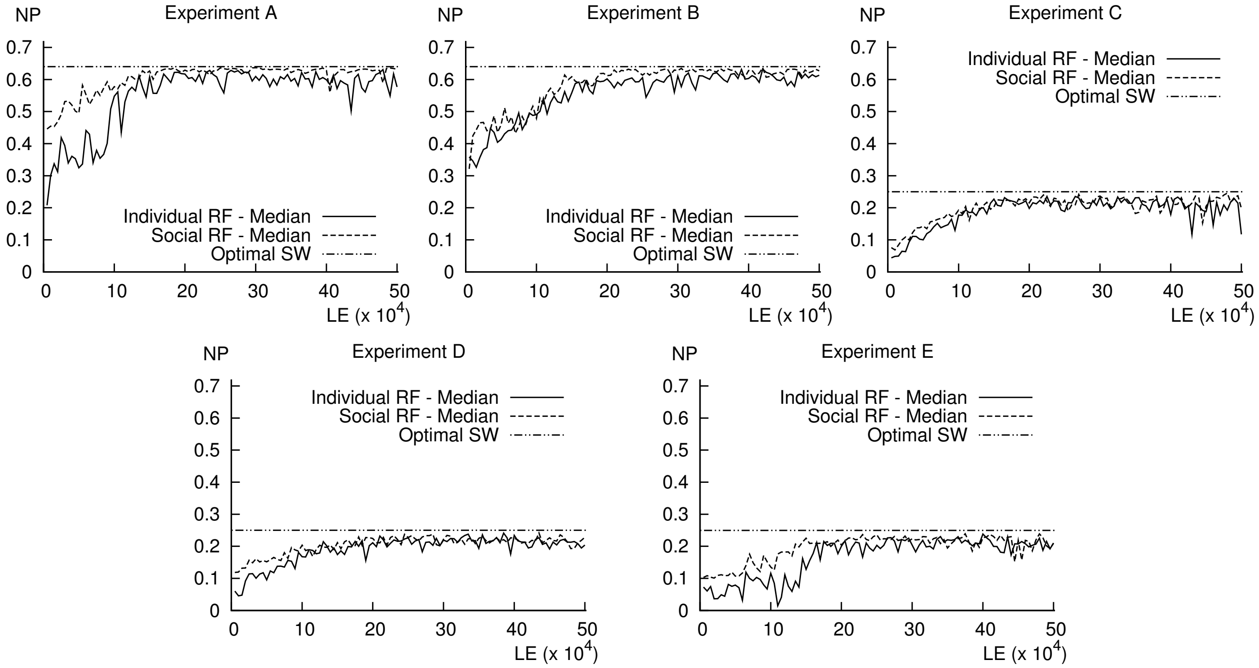

Figure 3: Median Nash Product (NP) of the market for the Individual and Social Reward Functions (RF) over the Learning Episodes (LE).

functions. The median approaches the optimal SW in all configurations. It achieves a level of relative stability before 200 000 learning episodes and fluctuates slightly afterwards.

The fluctuation is mostly influenced by characteristics of Q-learning inherent in multi-agent scenarios. Q-learning is only proven to converge in stationary environments and if the parameters for learning and exploration were decayed properly. However, with multiple learners, the environment is dynamic because of the effect of co-adaptation. Co-adaptation occurs when one agent adapts its strategy to the others', and vice-versa, in a continuous cycle. In addition, we have not applied decay rules for the parameters. We expect that the instabilities can be reduced with the use of decay rules and other multiagent RL algorithms [13]. Nevertheless, Q-learning has been applied with success in multiagent environments (for example [5] and [8]) and the results obtained in this research indicate its suitability for the IPA Market.

It is also noticeable in Figure 3 that the social reward function generated better SW during the first learning episodes. This result was expected as those agents explicitly learn how to maximize the SW.

3.2.2 Individual Level

We now turn to the individual utility of the agents. Figure 4 shows the comparison of the median utility received by the agents over the learning episodes. It is important to recall that the utility of an agent in our scenario is actually the product of its utility for price and amount of resource, $U(p, m) = UP(p) * UM(m)$.

Experiment A reflects the situation of agents with symmetric preferences. The optimal SW for this configuration is 0.64. The fairest Pareto-Optimal (PO) allocation that maximizes the SW produces a utility of 0.8 to each agent. To achieve it, the price has to be low enough ($p \leq 1$), so the agents' utility for the price is optimal (i.e. $UP(p) = 1$), and they need to obtain 5 units of resources each. It can be seen in the graphs that both reward functions were able to direct the agents' utility to the optimal level. It confirms the results found in [6] for the same case.

In **experiment B** the agents have the same preferences for amount of resource but the second agent is willing to pay more. The optimal social welfare in this case is also 0.64. The graphs show that both reward functions direct the agents to 0.8, which maximizes the SW. The graphs also show a difference between the utility of the agents 1 and 2 in the first learning episodes. It happens because the agent 2 has an easier utility function for price and learns how to maximize it very early. Nevertheless, the agent 1 also approaches the optimal utility after some episodes, indicating that the agents have learnt to coordinate the demand functions in order to decrease the price. This learning process was quicker for the social reward function because it carries information about the social utility in the reward signal.

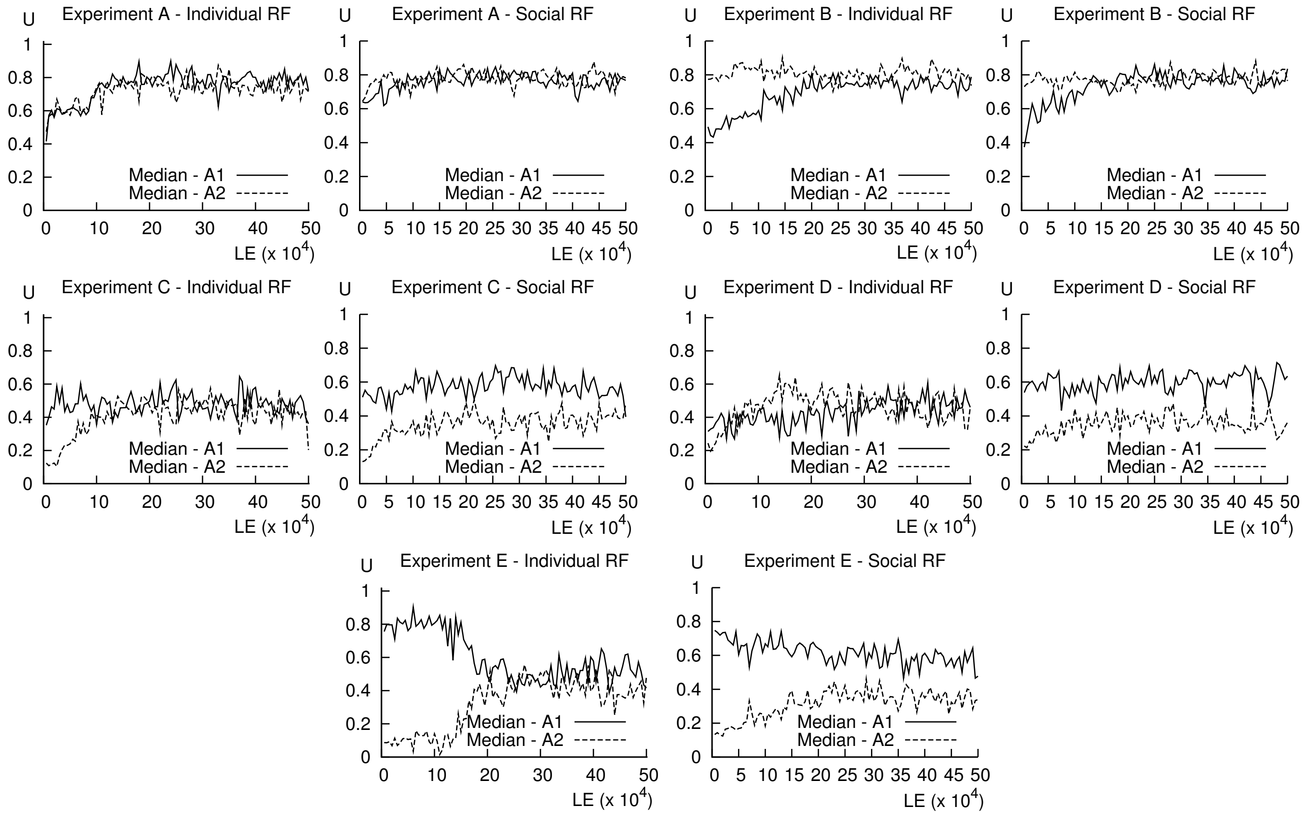

Figure 4: Median Utility (U) of the agents for the Individual and Social Reward Functions (RF) over the Learning Episodes (LE).

Such information is not carried by the individual reward function, making the coordination more difficult to emerge.

Experiment C investigates the situation in which both the agents are willing to pay the same price but the second agent needs more resources. The optimal social welfare in this case is around 0.25. In the fairest PO allocation, each agent receives a utility of 0.5, which corresponds to demand levels 3.5 for the first agent and 6.5 for the second (see Figure 2). The graph for the individual reward function shows the median utilities of both agents fluctuating around 0.5, indicating that they have learnt demand functions able to find the equilibrium around the optimal levels. The graph for the social reward function, however, shows a gap between the agents' utility. This gap was generated because the agents developed demand functions that directed the equilibrium to demand levels around 4 and 6. We comment on this behaviour below.

In **Experiment D**, the first agent needs less resource and prefers to pay less for it. The results found for this case are a mixture between the ones found for experiments B and C. As in B, the agents coordinated to lower the price so both agents maximized UP. As in C, there was a gap between the utility of the agents when the social reward function was used. Similar behaviour was observed in **Experiment E**, which studies the case where the first agent wants less resource but is willing to pay more for it.

To illustrate the cases of the experiments C, D and E, Figures 5 and 6 present the evolution of the individual components for experiment E. Figure 5 shows how the demand requests evolved over the learning episodes. Note that they stabilized around 6.5 and 3.5 for the individual reward function and around 6 and 4 for the social reward function. This behaviour generates a gap in the agents' utility for amounts of resource, being the responsible for the gap between the final utility of the agents using the social reward function. An interesting aspect shown in Figure 6 is the evolution of the utility for the price. Note that the agent 1's utility for this component is optimal from the very first learning episodes while the agent 2's utility starts lower but improves very quickly and stabilize at the optimal level. Similar behaviour was found in experiments B and D. Also note the price decreasing over the learning episodes, showing that the agents have learnt to coordinate the demand functions in order to decrease the price.

Figure 7 presents the relationship between the individual utility and the social welfare for the experiment E. The

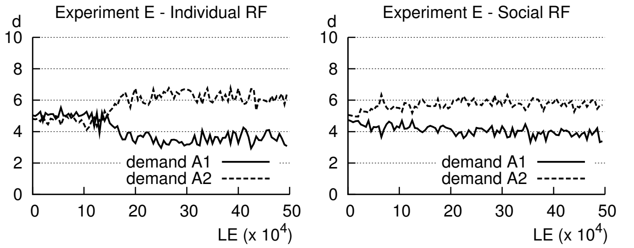

Figure 5: Median demand (d) requests of the agents over the Learning Episodes (LE) for experiments of type E.

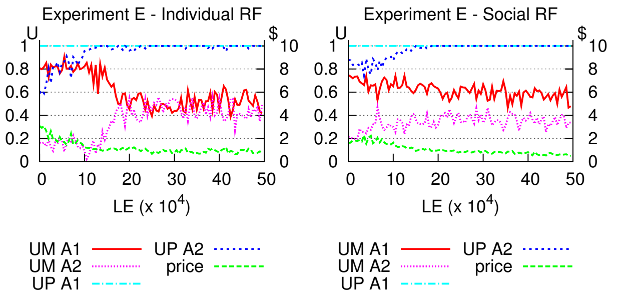

Figure 6: Median Utility (U) for amount of resource (UM) and price (UP) of the agents in experiments of type E.

graph shows that the social reward function is noisy in the sense that small changes in the demand requests produce large differences in the individual utility of the agents but reflect very little changes in the social reward signal. For example if the equilibrium is found at the demand levels 3.5 and 6.5 for agent 1 and agent 2, respectively, the optimal social reward of 0.25 is achieved and the agents receive a utility of 0.5 each. On the other hand, if the equilibrium is found at 4 and 6, the social reward produced is very close to the optimal one, i.e. 0.24, but the utility of the agents is significantly different, 0.4 and 0.6. This illustrates that the individual results obtained by the social reward function in the experiments C, D and E are reasonable. However, there are two open issues. First, the relationship shown in Figure 7 is similar for the experiments A and B, but the results for these experiments do not present the gap in the median utility of the agents. And second, the median demand requests at the equilibrium have evolved to 4 and 6, rather than to 3 and 7, which would produce the same social reward. These issues are being investigated and will be reported in a separate paper.

In summary, regarding the agents' preferences for prices, the experiments have shown that the agents can learn to coordinate the demand requests in order to lower the price and, consequently, maximize $UP(p)$. This behaviour held true for both reward functions even when different preferences are present. For the amount of resources, in experiments where the agents have the same preferences (A and B), both reward functions were able to approximate the solution to the optimal and fairest allocation. For experiments in which different preferences are applied (C, D and E), only the individual reward function was able to do it.

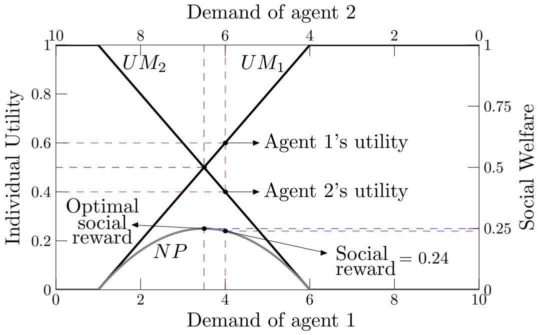

Figure 7: Relationship between individual utility and social welfare

4 Related Works

There is a significant amount of work on learning in market-based systems. While most of the approaches have not been developed exclusively for resource allocation, they can be naturally extended for it. One branch of research corresponds to the improvement of auction mechanisms by learning bidding and asking strategies [11] or learning the auction parameters in order to maximize a given objective function (e.g. auctioneer revenue) [10]. Another area involves the learning of negotiation strategies, which can be used to improve bargaining-based resource allocation systems, as in [12]. There is also some work on learning for pricing [8].

The fundamental difference between the above works and our approach is in the expression of the preferences of the agents. Previous works usually consider the utility of one agent as a function of the profit it makes in the market. Our agents, in contrast, describe their preferences by explicitly modelling utility functions for attributes of the allocation. It has the advantage of providing more power to the agents in developing strategic behaviours, which is particularly interesting in small markets where the possible gains from strategic attempts are larger, and achieving fair-optimal allocations.

Learning has also been explored to improve non-market-based resource allocation, for example [3, 5]. The preferences of the agents in these systems are typically based on scheduling parameters, such as reducing the time interval between a job submission and its completion.

Finally, the development of a utility-based pricing mechanism has been proposed by Chunlin & Layuan [2]. The authors, however, do not use utility functions for individual resource attributes, as we do. Instead, they use a single function taking into account some attributes. In addition, learning is not applied so the behaviour of the agents is rather static.

To the best of our knowledge, the approach presented in this paper is the first attempt to address the problem of learning demand functions.

5 Conclusions

In this paper we have studied the IPA market-based resource allocation with RL for the case of agents with non-symmetric preferences. We have shown through experimental investigation that the application of a reward function based on the individual utility of the agents can generate social results similar to the ones obtained with the application of a reward function based on the social welfare of the allocation. Remarkably, the individual-based reward function was also able to approximate the solution to the fairest Pareto-Optimal allocation in situations where the social-based one failed. This outcome is potentially important in the domains where social utility should be maximized but agents are unwilling to reveal private preferences, required for explicit optimization of social welfare.

There are several aspects in which this work could be extended. In particular, it is important to model the problem theoretically in order to identify what are the reasons for the behaviours found and the conditions in which they hold. Stochastic Games Theory has been applied to model multi-agent learning problems [1] and is useful to highlight some properties of the solutions found by the learners. However, they fail to capture the dynamic essence of the learning process. Evolutionary Game Theory (as applied in [9]) may be an alley for additional theoretical exploration.

Another area for extension involves the reduction of the required amount of learning, which is in general related to well-known scalability issues of *Q*-learning. It is also necessary to evaluate the approach in extended scenarios, including agents with preferences described over multiple attributes, multiple markets, and the existence of resource provider agents. The later deals with a limitation of the IPA mechanism itself. In particular, the IPA does not model resource provider agents. Simply adding those agents to the model may change its theoretical implications as the resource supply becomes dynamic and, therefore, has to be carefully considered. Different pricing mechanisms will be subject of our future works.

References

[1] M. H. Bowling and M. M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136(2):215–250, 2002.

[2] L. Chunlin and L. Layuan. Pricing and resource allocation in computational grid with utility functions. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*, volume II, pages 175–180, Washington, DC, USA, Apr 2005. IEEE Computer Society.

[3] B. C. Csáji and L. Monostori. Adaptive algorithms in distributed resource allocation. In *Proceedings of the 6th international workshop on emergent synthesis (IWES 06)*, 2006.

[4] H. Everett. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11(3):399–417, 1963.

[5] A. Galstyan, K. Czajkowski, and K. Lerman. Resource allocation in the grid using reinforcement learning. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, volume 3, pages 1314–1315, Washington, DC, USA, 2004. IEEE Computer Society.

[6] E. R. Gomes and R. Kowalczyk. Learning the ipa market with individual and social rewards. In *Proceedings of the International Conference on Intelligent Agent Technology (IAT 2007)*, pages 328–334. IEEE Computer Society, 2007.

[7] P. Jennergren. A price schedules decomposition algorithm for linear programming problems. *Econometrica*, 41(5):965–980, Sep 1973.

[8] J. O. Kephart and G. Tesauro. Pseudo-convergent q-learning by competitive pricebots. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML'00)*, pages 463–470, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[9] L. Panait, K. Tuyls, and S. Luke. Theoretical advantages of lenient learners: An evolutionary game theoretic perspective. *Journal of Machine Learning Research*, 9(Mar):423–457, 2008.

[10] D. Pardoe, P. Stone, M. Saar-Tsechansky, and K. Tomak. Adaptive mechanism design: a metalearning approach. In *Proceedings of the 8th International Conference on Electronic Commerce*, pages 92–102, New York, NY, USA, 2006. ACM Press.

[11] C. Preist, A. Byde, and C. Bartolini. Economic dynamics of agents in multiple auctions. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 545–551, New York, NY, USA, 2001. ACM Press.

[12] B. Schnizler, D. Neumann, D. Veit, M. Reinicke, W. Streitberger, T. Eymann, F. Freitag, I. Chao, and P. Chacin. Catnets - wp 1: Theoretical and computational basis, 2005.

[13] Y. Shoham, R. Powers, and T. Grenager. Multi-agent reinforcement learning: a critical survey., 2003.

[14] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[15] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK, 1989.

[16] T. Wu, N. Ye, and D. Zhang. Comparison of distributed methods for resource allocation. *International Journal of Production Research*, 43(3):515–536, 2005.

[17] C. S. Yeo and R. Buyya. A taxonomy of market-based resource management systems for utility-driven cluster computing. *Softw. Pract. Exper.*, 36(13):1381–1419, 2006.