Reinforcement Learning with Utility-aware Agents for Market-based Resource Allocation

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

Ryszard Kowalczyk
Swinburne University of Technology
Faculty of Information and Communication Technology
Hawthorn, 3122 Victoria, Australia
rkowalczyk@ict.swin.edu.au

ABSTRACT
In this paper we propose and investigate the use of Reinforcement Learning in a market-based resource allocation mechanism called Iterative Price Adjustment. Under standard assumptions, this mechanism uses demand functions that do not allow the agents to have preferences over the attributes of the allocation, e.g. the price of the resources. To address this limitation, we study the case where the agent's preferences in the resource allocation are described by utility functions and they learn the demand functions given their utility functions. The approach has been evaluated with extensive experiments.

Categories and Subject Descriptors
I.2.6 [**Artificial Intelligence**]: Learning I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms
Experimentation, Algorithms

Keywords
Reinforcement Learning, Market-based Resource Allocation

1. INTRODUCTION
Market-based resource allocation mechanisms offer a promising approach for systems that need distributed resource allocation without centralized control [8], for example the GRID. One of those mechanisms is the Iterative Price Adjustment (IPA) [4]. This mechanism uses the concept of a "price" which is iteratively adjusted to find equilibrium between a set of demands and a limited supply of resources.

In the IPA, the interests of the agents in the resource allocation are described by means of demand functions. Under standard assumptions, those demand functions specify a relationship between price and demand, and as such do not allow the agents to have preferences over the attributes of the allocation, for example the price. It makes difficult to influence and optimize the allocation quality in terms of the utility received by the agents. One of the alternatives to address this problem is to let the agents to describe their interests using utility functions instead of demand functions. In such a scenario however, the mapping between the utility functions and a demand function is not unique. If "made by hand", the definition of this mapping is very subjective and may turn into a very complex and time-consuming task, especially if one considers agents with multiple utility functions. In addition, the resulting agents might not perform well in the real system. Thus, in this paper we propose and investigate the use of Reinforcement Learning (RL) [6] to let the agents learn the best demand functions given their utility functions. The resulting demand functions are applied in an ordinary IPA market for their evaluation.

AAMAS'07, May 14-18, 2007, Honolulu, Hawai'i, USA.

2. LEARNING THE DEMAND FUNCTIONS
In the experiments we considered a single IPA market with two client agents. The agents have preferences over the price and over the amount of resources. Such preferences are represented by two utility functions: $U_1(p)$, for price; and $U_2(m)$, for the amount of resource. For the sake of simplicity, but without loosing the generality, we use only one type of resource, e.g. *memory*.

We used the ordinary *Q*-learning algorithm [7] and the ε-greedy action selection mechanism for the learning. The current prices of the resources at each step of the IPA negotiation process are mapped into the environment states, as this is the only information the clients have available in the IPA market. The agents' actions are mapped to the amounts of resource they can require. And, the rewards are mapped to a function of the utilities the agent receives at the end of the allocation procedure.

To encourage the agents to improve the social welfare, they are trained jointly and using the same reward functions. The final state of each learning episode is reached when the market is cleared. The agents receive a positive reward only when they reach the final state, i.e. they act towards the market clearance. This reward is given by the function $U(p, m)$ which is a combination obtained from the product of $U_1(p)$ and $U_2(m)$. It is used to stress the fact that both criteria are important in resource allocation. In all other states, the agent receives a reward equal to zero.

The utility functions $U_1(p)$ and $U_2(m)$ used by the agents are:

$$U_1(p) = \begin{cases} 1, & \text{if } p < 2.85714 \\ -((0.35p)-1)^2 + 1, & \text{if } 2.85714 \le p \le 5.71429 \\ 0, & \text{if } p > 5.71429 \end{cases}$$

$$U_2(m) = \begin{cases} 0, & \text{if } m < 2 \\ 0.5Log(m-1), & \text{if } 2 \le m \le \sim 8.38906 \\ 1, & \text{if } m > \sim 8.38906 \end{cases}$$

The market had a supply of 10 units of memory in all experiments. This amount does not permit for all the agents to have a complete satisfaction but allows us to analyze the behavior of the market and the learning mechanism under a condition of limited supply. The initial prices of the resources were obtained from a random number between 0 and 10 units of price.

We performed a series of preliminary experiments in order to identify a feasible configuration for the values of the parameters used in the learning algorithm. Based on this experiments we set α= 0.1, γ= 0.4 and ε= 0.4. The price of the resources is adjusted by the IPA market using a constant parameter set to 0.1. The continuity of the states and actions is treated by the application of a rounding procedure. Both, states and actions, are rounded to 1 decimal place.

We ran 4 different experiments, all with the same configuration given above. Each experiment was run for a total of $1x10^6$ episodes. From these experiments we obtained 8 agents. The evolution of the demand functions of these agents over the episodes showed a similar behavior. In the first 50 000 episodes, they are still not so consistent, what can be explained by the fact that the agent has probably not visited all the actions in all the states yet. From this point on, they start to evolve towards a well defined trend as you can see in Figure 1, which shows the demand functions of four of our agents. An interesting point is that the demand functions did not present a high demand for lower prices as it might be expected from observing the utility functions. This behavior is most likely generated from the learning format we used, which incentives the social welfare improvement rather than the individual, even though the agents are self-interested.

Analyzing the evolution of the demand functions, we also note that it is not completely stable. Although the shape is always there and is quite stable between the same values for demand and price, the individual values presented some variability from one episode to another. This behavior can have its origin in several factors, from the application of *Q*-learning for two simultaneous learners, passing by the not appropriate decay rule for the learning rate, to the design choices for the learning implementation.

Even though the agents have not achieved exactly the same demand function, the functions obtained are consistent and presented a very similar overall trend.

3. USING THE LEARNT DEMAND FUNCTIONS

In order to avoid any possible deadlock if the learnt demand functions are used directly in an ordinary market running the IPA method, we use the trends of the demand functions to evaluate their performance in the resource allocation. We randomly selected 4 of our 8 agents to apply a curve-fitting method. We used the demand functions obtained at step 450 x 10 [3]. The rationale for this is that at this step some of the agents have already developed a demand function with a consistent shape, as you can note in the Figure 1.

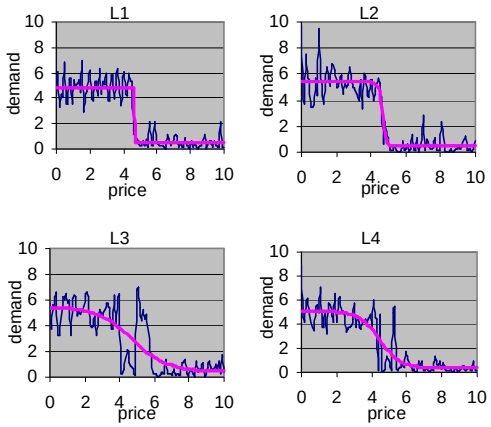

Figure 1. Agents' demand functions with trend line.

We made experiments using the 4 demand functions resulting from the curve-fitting method in the ordinary IPA market. We evaluated the agents using these demand functions against 3 other agents with predefined "static" demand functions. The demand functions of these agents were defined by hand, based on subjective criteria, and they also use the function *U(p, m)* to evaluate the quality of the allocation. The new agents are S1, S2 and S3. We ran a total of 49 experiments using the same market configuration as in the learning phase and the same number of agents.

The results of the experiments were assessed from an individual and a social perspective. The individual perspective is important because it shows how the agents performed in terms of their own utilities. However, the social perspective is the most suitable evaluation criteria for this work as the learner agents were trained jointly and encouraged to improve the social welfare.

Figure 2 shows the individual utilities for experiments of type learner vs. learner and static vs. static. In general, the utilities of the learner agents are better than the static ones. There is also some similarity among the utilities received by the learner agents. The situation is little different for the static agents, where one of them, S3, achieved a very poor individual performance.

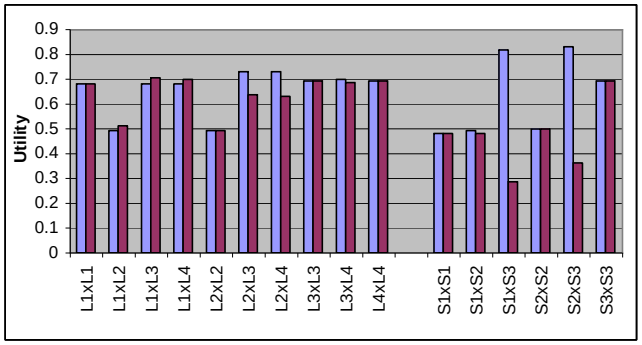

Figure 2. Individual utilities in learner vs. learner and static vs. static experiments

Experiments of type learner vs. static showed that the four learner agents presented similar performances against the same static agents. This was expected as they learn using the same set of utility functions. These experiments also presented that the performance of the learner agents was not so good if compared

with the performance of the static ones. It is important to mention that this does not mean that the learned demand function is meaningless. One has to consider the fact that the learners had not considered the static agents during their learning phase and that they are trained together to achieve equilibrium and obtain a better social welfare. This objective is successfully achieved as presented next.

Regarding the average individual utilities obtained by the agents over the experiments, the agents using the learned function obtained almost the same utility as the static agents. It is quite interesting if you note that in the individual comparisons, the learner agents were beaten in most of the experiments. This equalization is achieved because when running against themselves, the learner agents achieve a much better solution.

We use the concept of Social Welfare (SW) [2] to evaluate how the market performed under the social perspective. We applied three different functions to calculate the SW of our market: the Utilitarian Social Welfare (USW), defined as the sum of individual utilities; the Egalitarian Social Welfare (ESW), given by the utility of the agent which is worst off; and the Nash Product (NP), defined as the product of the individual utilities. These three SW functions showed that the market's performance is improved using the agents with learnt demand functions as it is presented in Figure 3 a), b) and c).

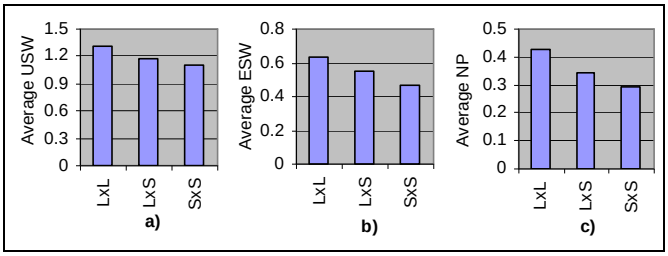

Figure 3. a) Average USW; b) Average ESW; c) Average NP

4. RELATED WORKS

As far as we are aware no work has addressed the problem of learning a demand function based on a set of utility functions. However, the use of reinforcement learning in resource allocation is not new. Galstyan *et al*. [5] used reinforcement learning in a scenario where a large number of users submit their jobs to resources that are scheduled by a local scheduler in a grid environment. Abdallah and Lesser [1] proposed a multiagent reinforcement learning algorithm and applied this algorithm in distributed task allocation.

Regarding the application of utility-aware agents, Chunlin & Layuan [3] developed an algorithm based on utility functions for resource allocation for the Grid. However, such algorithm does not make any reference about agents having preferences over the price of the resources, as we make in our work.

5. CONCLUSION

In this paper we proposed and investigated the use of Reinforcement Learning in a market-based resource allocation mechanism called Iterative Price Adjustment. Under standard assumptions, this mechanism uses demand functions that do not allow the agents to have preferences over the attributes of the allocation, e.g. the price of the resources. We studied the case where the agent's preferences in the resource allocation are described by utility functions and they learn the demand functions given their utility functions. The experiments were divided in two phases. In the first phase we applied the *Q*-learning algorithm to let the agents learn their demand functions. This phase considered a single market composed of two client agents. They are trained jointly and encouraged to improve the market's social welfare while being self-interested. The second phase was the application of the learnt demand functions in the standard IPA market. The results of the experiments have shown that through the application of ordinary Q-learning the agents were able to learn meaningful demand functions. Under an individual perspective, agents using the learnt demand functions performed well in comparison to ones with demand functions defined "by hand". More remarkably, under a social perspective, the agents using the learnt demand functions achieved a much better solution, being able to improve the system's social welfare.

The experiments have shown the feasibility of the approach and pointed out that further investigations in this direction are worthwhile. An immediate future work is to further investigate the problem of co-adaptation as we use agents being trained jointly. Another future work is the extension of the scenario to better reflect a real distributed system, such as the Grid. This extension is likely to involve the use of agents described by multiple utility functions and participating in more than one market at same time.

6. REFERENCES

[1] Abdallah, S. and Lesser, V. Learning the task allocation game. In *Proceedings of the Fifth international Joint Conference on Autonomous Agents and Multiagent Systems* (Hakodate, Japan, May 08 - 12, 2006). AAMAS '06. ACM Press, New York, NY, 2006, 850-857.

[2] Chevaleyre, Y., Dunne, P. E., Endriss, U., Lang, J., Lemaître, M., Maudet, N., Padget, J., Phelps, S., Rodríguez-Aguilar, J. A., and Sousa, P. Issues in Multiagent Resource Allocation. *Informatica*, 30, 2006, 3-31.

[3] Chunlin, L. and Layuan, L. 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 02 (April 04 - 06, 2005). ITCC. IEEE Computer Society, Washington, DC, 2005, 175-180.

[4] Everett. H. Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resource. *Operations Research*, 11, 1963, 399-417.

[5] Galstyan, A., Czajkowski, K., and Lerman, K. Resource Allocation in the Grid Using Reinforcement Learning. In *Proceedings of the Third international Joint Conference on Autonomous Agents and Multiagent Systems* - Volume 3 (New York, New York, July 19 - 23, 2004. IEEE Computer Society, Washington, DC, 2004, 1314-1315.

[6] Sutton, R. S., and Barto, A. G. *Reinforcement learning: An introduction.* MIT Press, Cambridge, MA, 1998.

[7] Watkins, C.J.C.H. *Learning from Delayed Rewards*. PhD thesis, Cambridge University, Cambridge, England. 1989.

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