A survey of Market Oriented Programming

by Daniel Flower

The University of Auckland

March 2005

Abstract

Multiagent systems (MAS) have great potential in many areas. However, the larger the size of the population in the MAS, the more difficult central control becomes. When systems are open, allowing other, unknown agents to participate, making sure the system operates well becomes even more difficult. It is for this reason that economic ideas have become an attractive area of study for MAS, because human society has flourished where individuals, trying to maximise their own well being subject to economic forces, maximise the well being of the entire society. This survey will look at the benefits and problems behind using economic ideas, the achievements and areas that still need to be improved.

1. Introduction

Multiagent systems have a great promise in many areas, from controlling air conditioning in buildings to creating civilisations on other planets. However, when the number of agents in the system numbers hundreds or even thousands, there is great difficultly in controlling them to effectively work together. When the system is an open system where the trustworthiness of other agents is unknown, the problem becomes even more difficult. It was soon realised that there already existed many such systems that were very efficient: capitalist economies from all over the world. The economic ideas that explained how a group of individuals who were concerned mostly with bettering their own situation could lead to the best outcome for all of the society was the inspiration for the area of using Market Oriented Programming (MOP) to control multiagent systems.

MOP can be described as a paradigm for creating MAS where the individuals are free from any central control, and they each act in their own self-interest, ignoring the good of the group as a whole. The agents act rationally, subject to economic constraints such as scarcity of resources, and there must be actions that they can pay for in order to gain a benefit from them. Global goals are reached by giving incentives to agents to perform certain actions, and it is claimed that having greedy, individual agents can yield better results than having an all-knowing, central controller.

This paper will overview the history of MOP, followed by the key components of a MOP system in section 3. Possible pitfalls are outlined in section 4, followed by comparisons to centrally based systems in section 5, ending with a conclusion which looks at future areas that need work.

2. History

The use of economic theory to implement MAS was explored as early as 1968. In the paper "A Futures Market in Computer Time" [10], Sutherland used ideas of supply and demand to allocate computer time to users at Harvard University. The expense of computers made time allocation an important problem, and rather than using a (human) computer operator to allocate time or using some sort of priority queue, users were given credits to buy computer time, and the price of computer time depended on demand at that time (higher demand meant a higher price). The system was therefore self-regulating: it effectively determined when peak times were and tried to discourage use during those times. On the other hand, idle time was minimised because when the computer had very low demand, the price to use it was very cheap, encouraging its use. The result was that computer utilisation was "very high" compared to other solutions.

Other studies throughout the seventies and eighties followed (eg. [2] which examined resource allocation, and [4] which looked at simulating negotiations to solve distributed problems).

In 1993, Wellman [14] introduced the term "Market Oriented Programming". It used the idea of agents being set up as producers and consumers of goods (for example, a consumer may "consume" information from a "producer", e.g. a database agent produces data for an agent who requests it). If "goods" are given prices, and consumers given "money" to purchase goods, supply and demand curves can be calculated. Wellman focused on the importance of reaching equilibrium (i.e. where the supply and demand curves meet), and having agents act competitively. To be competitive, an agent takes the prices set at equilibrium while ignoring the effect its actions have on society. To illustrate, he used a multicommodity flow problem where agents were given the task of shipping cargo to a destination, where different routes had different costs, and congestion from overuse of routes emerged. Agents would be "paid" for delivery of goods, and would themselves "pay" for the use of roads etc. The optimal solution (i.e. the solution with the lowest cost) was known beforehand. In his first simulation, the outcome was not optimal, although close to optimal. However, he then went on to introduce different types of agents ("carriers" transported goods down certain routes only, and arbitrageurs[1] were used to fix inefficiencies in the market) and an optimal solution was found.

The fact that the problem was solved by simply programming agents to act in their own best interests showed a lot of promise for Market Oriented Programming, however there were caveats. In economic theory, it is assumed that when there is enough competition, the acts of a single consumer or producer can be regarded as negligible. However, if there are not many agents in a simulation, or one agent has a disproportionate amount influence over the economy, the agent's actions can affect the economy, and unless the agent takes these effects into account, non-optimal actions will occur. The obvious solution is to promote competitiveness in the virtual market by introducing more agents.

Two years later, Wellman reflected on some more lessons learnt in [15]. He pointed out that the most important aspect of MOP was in defining the goods of the economy. He also pointed out that the development of multi-good market was important, and also potentially a difficulty. As in a real economy, different goods may compliment each other, be substitutes, be used in the manufacture of other goods etc. Different goods are therefore interrelated and may have complex relationships.

Finally, he pointed out the importance of measuring the results of MOP systems, finding out whether these systems were computationally efficient, and whether the design was

[1] An arbitrageur purchases from one market with the purpose of immediately selling those goods on another market where the price is higher.

easier (in terms of implementation, testing, prediction of outcomes) than existing software construction paradigms. His work – and questions – lead the way for more detailed studies in the proceeding decade.

3. The Basic MOP Model

This section will overview the most important aspects required to create a successful virtual marketplace that will find an optimal solution to a distributed problem.

There must be producers and consumers, and goods and services must be purchased at a price. As pointed out above, the definition of goods (and services) is crucial to having a good economic system. Consumers must demand the goods and services (that is, they need to know that particular commodities will help them achieve their goals), and producers need to know if and how they can create products. The specification of the goods and services is what can create the goals of the system. For example, in the cargo example, the goal of the system was to transport cargo, and this could be realised by letting the agents know that they would be paid for delivery of cargo.

The supply and demand of individual agents must somehow be combined to create aggregate supply and demand curves, which should be used to discover what the market price of goods ought to be, and how many ought to be produced.

There must be scarcity in the market for all agents to flourish. In [1], robots could trade goods for energy. If some robots became too rich, they would have no need to trade, and therefore other robots would not receive any energy and die out. By making resources scarcer, the robots were encouraged to trade more, and fewer robots died out. Indeed, economics is based on the fact that resources are scarce, and trade is a result of this.

All agents must act rationally [8]. For consumers, this means agents will not pay more than the market price, and will not purchase goods unless utility gained is expected to exceed cost, and where there are options, the option with the greatest net utility will be chosen. Producers must try to maximise their profits.

Agents therefore require some sort of utility function which they must be able to calculate without help from an outside party (such as a central controller). Both consumers and producers are left with a constraint optimisation problem, where the cost for both types of agents is a decrease in their own available resources.

Negotiation between buyers and sellers therefore becomes very important. In [8], Mullen & Wellman made the following observation regarding negotiation in the market:

> *…even this simplest case is complicated by the fact that multiple agents are submitting bids for multiple goods simultaneously and asynchronously. Because the goods are highly interconnected through joint preferences and technology relations, the price of one good affects demand for another*

In [5], Ferguson et al. were trying to allocate computer system resources using MOP. They encountered the problem where an agent would discover the market price, make a decision based on the price, and then purchase the good. However, a common occurrence was that by the time the agent purchased the good, the market price had already changed, and the agent would end up making poor decisions. Mullen & Wellman suggested using an iterative approach to bidding for goods, where any price change will be propagated to agents with bids. This issue is only important in markets where the price is fluctuating, so having more stable markets would solve this problem.

4. Possible Pitfalls

Economic theory includes many scenarios which are considered as potential problems in the marketplace. This section will give details of a couple of these problems, and the solutions given.

The Tragedy of the Commons

The tragedy of the commons is a well-known problem in economics, which is where individuals are compelled to overuse commonly owned resources. The famous example (expressed in [6]) is of commonly owned pasture, where individuals are free to graze cattle. By having more cattle, more profit can be made, and therefore an individual will decide to graze more cattle. The tragedy is that every individual will come to this conclusion, and therefore the pasture will soon suffer from ruinous overuse. This problem in MOP systems can occur when agents have access to a free resource (for example, they may be free to use as much memory as they want, in which case the system could run out of memory) and in [15] Wellman stated that it can be fixed with the MOP version of privatisation: ownership of such resources would be given to certain agents, who would then be able to sell the resource in the market. The marginal cost of using the resource would no longer be zero, and so the tragedy of the commons is averted.

Lack of Diversity

A major component of an efficient economy is the division of labour. This is where individuals specialise in precise, distinct areas, and hence the work done in each part of the economy is done by a specialist in that area. For MOP systems to reach their potential, their must be a similar division of labour and specialisation.

This problem has already been discussed by Chislenko & Ramakrishnan in [3]. They point to the example of exploration, and concluded that an individual agent's best strategy is to wait for someone else to discover the possible benefits of untested tools, unexplored terrain etc, rather than taking the risk that no benefit will be gained but cost would be incurred. While one fix could be making the agents "curious", it is suggested that a better solution would be to introduce new agents that would pay others for exploration, and reap any benefits gained.

An interesting study into the area of having agents specialise was done by Smith et al. in [9]. Here, they used genetic algorithms to evolve just 3 agents that lived in an economy. They found that if they started with "Jacks of all trades", they soon evolved into specialists, with each agent specialising into a distinct area from the others, out of the 3 areas that were available to them. The heterogeneity required in an efficient economy is an important design issue for MOP, and [9] illustrates that perhaps a good way to achieve this is with machine learning algorithms.

5. Comparison to Centrally Controlled Systems

Part of the attractiveness in creating MOP systems is their lack of central control. This begs the question: how do they compare to centrally controlled systems? This section will give an overview of 2 papers that concentrated on problems that could be solved using both central and market oriented approaches.

Case study: "Nongovernance Rather Than Governance in a Multiagent Economic Society"

In this 2001 paper ([11]), the specific problem was that of allocating the locations of buildings in a city. The goal of companies who were creating the buildings was to be as close as possible to other buildings because in general, locations that have many companies have better infrastructure. The goal of the local government was to use as little space as possible. An interesting aspect of this study is that both goals of the companies and government lead to the same result: a concentrated city centre. The evaluation for both would therefore be based on the average density of buildings and the total area used.

Rather than dividing the simulation into two runs (centralised, or government controlled, and decentralised, or company controlled), they realised that there were four possibilities, which arise from a 2-dimensional matrix of possibilities[2]:

	Local evaluation	Global evaluation
Government intervention	Level 0	Level 1
No government intervention	Level 2	Level 3

[2] The column names used in this matrix are my own. Takadama et al. referred to the dimensions as direct and indirect control, with the options being "no intervention" or "behaviour intervention".

“Government intervention” refers to centralised control, and “no government intervention” refers to a market oriented approach. The evaluation function (similar to the utility function discussed earlier) could either be local, which is where the individual company would evaluate their own situation, or global, where the actions of individuals were evaluated based on the whole city’s situation.

In the simulation, companies were placed randomly on the map ignoring the fact that buildings may overlap each other. As a consequence, many buildings did overlap each other, and so the problem is solved by companies who have overlapping buildings moving their building into a free spot, either using their own strategy or being forced to follow government guidelines. They found that in both cases where global evaluation was used, the solution was worse than when local evaluation was used. This goes against intuition, because it is tempting to presume that if all companies are trying to minimise the total area of all companies, they should do it more successfully than if they are only concerned with their own situation. Their conclusion was that this was a result of companies trying to please the government, and hence acting in similar ways. With local evaluation, each company can do its own thing, and more novel solutions are found.

They found that there was not a big difference between government intervention and non-intervention to the overall solution; however the cost in reaching the solution was higher for government intervention. They stated that government intervention equated to a gradient search, which could easily get stuck in a local minima. They did find on repeated trials however, that the volatility of the cost for non-intervention was significantly greater than government intervention, and this was the only advantage they identified in using government control.

Case study: “Using Virtual Markets to Program Global Behavior”

In this 2004 paper ([7]), the problem of implementing a sensor network is examined. A sensor network is simply a collection of sensors that together could, for example, track the movement of a car. This would occur by sensors near the car sensing the car, and sending it to a central place where the movement of the car as a whole could be seen. The goal is to reliably track objects while using the fewest resources possible.

They acknowledge the difficulty of the problem, and point out that most solutions involve macro-programming, which is an attempt to program the system as a whole. However, the coding of individual nodes in the network still cannot be avoided with the previous methods, and so MOP seems to be an ideal solution.

In their simulation, nodes could sleep to conserve energy, sense to check for objects which costs energy, and send and receive messages to other nodes. Nodes were rewarded for sensing objects, and also for transmitting useful messages that were heard by other nodes. Each node would then need to decide on what to do based on the expected cost of the action, and the predicted utility to be gained. The nodes could learn with experience, and so become better at predicting utility, and hence making better decisions. Reinforcement learning was used: a successful decision was rewarded to encourage that reaction in the future.

The result was a network where the activity of a node depended on the location of the object being tracked. When a node knew that the object was near, it stayed alert, and when the object was further away it would sleep to preserve energy, while intermittently listening for messages, ready to spring to action again if necessary.

While it wasn't shown that this system gave a better solution than other centrally-controlled systems, it illustrated a powerful feature of MOP: the behaviour of the entire system could be orchestrated by simply changing prices. Any sensor network is trying to find a trade-off between energy use and tracking accuracy, and this level could easily be changed by changing prices. For example, if it was very important that the object was tracked accurately, prices for sensing could be dropped, while if energy conservation was important prices could be increased to make the sensors spend more time sleeping.

Furthermore, the solution was relatively easily realised compared to trying to program the coordination of the system as a whole, and as is claimed to be the case in many MOP systems, the communication overhead was kept to a minimum, because the agents don't want to waste their resources in sending unnecessary messages throughout the system.

6. Conclusion

This paper has given an overview of the history of Market Oriented Programming, and the current state of the art. To summarise, MOP has become a popular area of study due to the fact that multi-agent systems share many similarities with human economies, and human economies are widely believed to be the most efficient way to distribute resources and solve problems in a distributed environment. The implementation is also easier than centrally controlled architectures, as programming for single agents that act economically rationally is much easier than trying to get hundreds or even thousands of agents to successfully cooperate together.

Important steps have been taken in the last few decades. These started with the observation that MOP would be a good paradigm, and the determination of important factors needed in a MOP system (well defined goods and services, creating good utility functions, etc).

An advantage of MOP is that there has been centuries worth of studies in economics. As a result, much new work concentrates on more specific economic ideas, such as auctions and game theory, and also classic problems such as the Tragedy of the Commons. These problems are often solved using ideas from economics, and indeed many problems outlined in this paper were solved by simply adding more types of agents and hence increasing competition; an obvious solution to any economist! Looking at more specific economic theories to increase the efficiency of these virtual markets will be an ongoing area of research for market oriented programming.

Possibly the most important ongoing problem is making agents that can learn from experience using ideas from machine learning, such as genetic algorithms and reinforcement learning. Because it is imperative that individuals in an economy make good, rational decisions, the ability to make decisions based on previous experience is essential.

Future Work

In most of the papers contained here, the agents base their decisions depending on the current market price; however this is not always how humans make their decisions. For example, the behaviour of the past and prediction of future prices is taken into consideration. In [12], Tassier et al. studied this problem, and looked at recording the past prices of a good to base decision making on. For example, if a price stays steady for a while, agents may decide that they have found the "normal" price of a good. If the price increases, they may be willing to wait, expecting the price to return to the normal price later on. Giving agents the ability to reason based on more than just the current price of a good is an important area in MOP that should lead to more efficient markets, due to the greater rationality being used.

Another area requiring work is to do with positive and negative externalities. These are, respectively, where positive and negative effects, due to an exchange, affect others that did not participate in the exchange. For example, smoking has negative externalities because it increases (overall) the amount of healthcare needed, which is paid for by all taxpayers. On the other hand, studying can have positive externalities because research and higher skills benefit society. Without any outside intervention, an economy will (largely) ignore externalities, and the market will not reach a global optimum. Disincentives and incentives (e.g. taxes on cigarettes, subsidies for tertiary study) therefore need to somehow be established. So, the problem facing MOP designers is to identify any externalities (for example, the use of the CPU in resource allocation may slow down other processes), and then to somehow deal with the externalities. Whether there is a systematic approach to this problem, or whether designers will need to rely on ad-hoc approaches, remains to be seen.

More research into the difference between human economies and agent-based economies also needs to be undertaken. For example, while economic theory is based on rational behaviour of individuals, humans do not always act rationally; they are swayed by emotion, aesthetics, and sometimes inability to calculate what a fair price for a good is. In [13], Varian points out that game theory has often been criticised for it's "hyper-rationality", and that in this sort of case the economic theory is in fact more applicable to agents than humans. The exact problems and benefits that arise in an agent-based market rather than a human-based market, and how these would affect the market, are yet to be fully explored, and would be interesting from both a computer science and economic standpoint.

References

[1] Ali Cherif, A., (1994), *A constitution and an economic model for the organisation and emergence of collective behaviour in a colony of robots*. Lab. d'Intelligence Artificielle, Univ. de Paris-Nord, St. Denis, France, This paper appears in: From Perception to Action Conference, 1994., Proceedings, pp: 334 - 337

[2] Arrow, K. J., & Hurwicz, L. (Eds.). (1997). *Studies in Resource Allocation Processes*. Cambridge University Press, Cambridge.

[3] Chislenko, A., Ramakrishnan, M., (1998), *Hyper-economy: combining price and utility communication in multi-agent systems*, Soc. of Mind Group, MIT, Cambridge, MA, USA; This paper appears in: Intelligent Control (ISIC).

[4] Davis, R. & Smith, R. G. (1983). *Negotiation as a metaphor for distributed problem solving*. Artificial Intelligence, 20, 63-109.

[5] Ferguson, D., Nikolaou, C., Sairamesh, J., & Yemini, Y., (1996), *Economic models for allocating resources in computer systems*. In Scott Clearwater, editor, Market-Based Control: A Paradigm for Distributed Resource Allocation, Scott Clearwater. World Scientific, Hong Kong, 1996. http://citeseer.ist.psu.edu/ferguson96economic.html

[6] Hardin, G. (1968), *The Tragedy of the Commons*, Science, 162(1968):1243-1248.

[7] Mainland, G., Kang, L., Lahaie, S., Parkes, D. C., & Welsh, M., (2004), *Using Virtual Markets to Program Global Behavior*, http://citeseer.ist.psu.edu/708474.html

[8] Mullen, T., & Wellman, M.P., *Some issues in the design of market-oriented agents*. In M. Wooldridge, J. Mueller, and M. Tambe, editors, Intelligent Agents: Theories, Architectures, and Languages, volume II. Springer-Verlag, 1996. http://citeseer.ist.psu.edu/mullen96some.html

[9] Smith, R.E., Bonacine, C., Kearney, P., Eymann, T., (2000), *Integrating economics and genetics models in information ecosystems*, Evolutionary Computation, 2000. Proceedings of the 2000 Congress on , Volume: 2 , 16-19 July 2000, pp:959 - 966 vol.2

[10] Sutherland, I. E. (1968). *A futures market in computer time*. Communications of the ACM, 11 (6), 449{451.

[11] Takadama, K., Terano, T., Shimohara, K., (2001), *Nongovernance rather than governance in a multiagent economic society*, Adv. Telecommun. Res. Inst. Int., Kyoto, Japan; This paper appears in: Evolutionary Computation, IEEE Transactions on, pp: 535 – 545

[12] Tassier, T., Everson, M.P., Ostrowski, D., (2002), *Agent-based models as a complement to economic theory: a durable goods example*, Econ. Dept., Iowa Univ., Iowa City, IA, USA; This paper appears in: Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on, pp: 729 – 734

[13] Varian, H. R., (1995), *Economic mechanism design for computerized agents*. In USENIX Workshop on Electronic Commerce, New York, NY, July 1995. http://citeseer.ist.psu.edu/article/varian95economic.html

[14] Wellman, M. P. (1993) *A market-oriented programming environment and its application to distributed multicommodity flow problems*. Journal of Artificial Intelligence Research, 1:1-22, 1993. http://citeseer.ist.psu.edu/wellman93marketoriented.html

[15] Wellman, M. P. (1995), *Market-oriented programming: some early lessons*. In S. H. Clearwater, editor, Market-based control: a paradigm for distributed resource allocation, chapter 4. World Scientific, 1995. http://citeseer.ist.psu.edu/wellman95marketoriented.html