A market-based optimization approach to sensor and resource management

Dan Schrage, Christopher Farnham, and Paul G. Gonsalves
Charles River Analytics; 625 Mount Auburn Street; Cambridge, MA 02138

ABSTRACT

Dynamic resource allocation for sensor management is a problem that demands solutions beyond traditional approaches to optimization. Market-based optimization applies solutions from economic theory, particularly game theory, to the resource allocation problem by creating an artificial market for sensor information and computational resources. Intelligent agents are the buyers and sellers in this market, and they represent all the elements of the sensor network, from sensors to sensor platforms to computational resources. These agents interact based on a negotiation mechanism that determines their bidding strategies. This negotiation mechanism and the agents’ bidding strategies are based on game theory, and they are designed so that the aggregate result of the multi-agent negotiation process is a market in competitive equilibrium, which guarantees an optimal allocation of resources throughout the sensor network. This paper makes two contributions to the field of market-based optimization: First, we develop a market protocol to handle heterogeneous goods in a dynamic setting. Second, we develop arbitrage agents to improve the efficiency in the market in light of its dynamic nature.

Keywords: sensor management, resource management, market-based optimization, dynamic planning

1. INTRODUCTION

Many current approaches to dynamic resource allocation simply repackage traditional planning and scheduling algorithms and try to apply them to a real-time environment, when what is needed is an approach specifically aimed at distributed resource networks.\(^1\) Multi-agent negotiation is a distributed artificial intelligence technique that has been successfully applied to resource allocation in a variety of domains, including allocation of distributed computing resources, just-in-time manufacturing controls, and real-time collection management (see Section 2), which depend on distributed resources and real-time adaptation to the mission environment. Success in these areas, all of which share important similarities with the real-time sensor network optimization problem in the target tracking domain, demonstrate a solid track record that indicates a high likelihood that this multi-agent negotiation approach will also succeed for target tracking.

Here, we present our system for Sensor Network Optimization using Multi-Agent Negotiation (SNOMAN) to meet the challenge of this real-time sensor management problem. Market-based optimization applies solutions from economic theory, particularly game theory, to the resource allocation problem by creating an artificial market for sensor information and computational resources. Intelligent agents are the buyers and sellers in this market, and they represent all the elements of the sensor network, from sensors to sensor platforms to computational resources. These agents interact based on a negotiation mechanism that determines their bidding strategies. This negotiation mechanism and the agents’ bidding strategies are based on game theory, and they are designed so that the aggregate result of the multi-agent negotiation process is a market in competitive equilibrium, which guarantees an optimal allocation of resources throughout the sensor network. Negotiation works continuously, providing dynamic adaptation to changes in the mission environment. Negotiation protocols are designed to minimize communication resource requirements, ensuring that the system scales well to more complex sensor networks.

Take, for example, the situation where an enemy aircraft has been detected by space-based surveillance heading towards naval assets, and ship-based anti-aircraft missiles are targeted at the hostile aircraft. The anti-aircraft missile system, in conjunction with the ship’s tracking system, becomes a consumer of sensor information at this stage. In order to hit its target, it must discriminate between the aircraft itself and other objects or debris around it. At the same time, it must continue to track the path of the aircraft in order to determine an engagement strategy. In the context of SNOMAN, the
anti-aircraft missile system is represented by a system agent, which drives information requirements. Sensors onboard the ship, along with air- and space-based sensors, are represented by sensor agents, which collect information. Computational resources such as processing nodes and network communication resources are represented by resource agents, which provide computational resources to other agents that need them to complete their tasks. All of these agents participate in the market for information through the mechanism of negotiation. For example, the anti-aircraft missile guidance system agent might calculate that the utility of LADAR data about its target is very high since it can help pinpoint the target’s position, so that agent will place bids with the available LADAR sensor agents throughout the sensor network. In order to gather LADAR data, these sensor agents need to obtain signal-processing resources from the resource agents, so they bid on the available processing nodes. Resource agents thus “sell” processing time to LADAR sensor agents, who then “sell” the information they gather to the anti-aircraft missile guidance system agent, who uses that data to compute the target’s trajectory.

This is only one example of a chain of transactions within the multi-agent sensor network, but game theory guarantees that, with properly designed negotiation mechanisms, the result for the system as a whole will be an optimal allocation of resources. Moreover, the negotiation process happens continuously, even as conditions change, which makes the approach perfectly suited for solving the real-time allocation problem unique to the target tracking domain. New targets might be detected, new sensor information might be needed, or sensor resources and computational resources might fail. The sensor network using this scheme seamlessly adapts to all of these kinds of circumstances. Moreover, because the system takes a multi-agent approach, it is highly flexible and extensible, so it can maintain optimal resource allocation under any configuration of sensors and network interfaces, including the addition of new sensor types.

Much past work in market-based optimization has focused on markets with homogeneous goods. In this case, however, the sensor market consists of heterogeneous buyers, sellers, and goods (information). This causes a number of complications and means that traditional general equilibrium models are not immediately applicable to this domain. To accommodate heterogeneous goods, we design a new kind of market infrastructure called the just-in-time (JIT) market. Traditional general equilibrium markets try to create a situation in which every part of the market is simultaneously in equilibrium and all transactions clear at the same time. For a sensor market with dynamically changing requirements and heterogeneous information tasks, this is unrealistic. Our JIT market addresses this by designing a market protocol that clears each market separately and at the latest possible moment, which allows us to maintain certain desirable efficiency properties in the market.

The JIT market framework also brings with it some new challenges. Foremost among these is what we refer to as the transitive selfish seller agent problem. This is a problem that results from the dynamic nature of the market, where the potential exists for a new task to appear that makes an existing allocation of buyers and sellers inefficient. To solve this problem, we introduce agents capable of arbitrage that search for and correct potential market inefficiencies.

This paper thus makes two contributions to the field of market-based optimization: First, we develop a market protocol to handle heterogeneous goods in a dynamic setting. Second, we develop arbitrage agents to improve the efficiency in the market in light of its dynamic nature.

Section 2 provides an overview of market-based optimization, including a review of related applications. In Section 3, we introduce our JIT market and compare it to past market implementations based on general equilibrium models. Section 4 describes the transitive selfish seller agent problem and our approach to solving it. Section 5 concludes the paper and outlines our plan for future improvements.

2. MARKET-BASED OPTIMIZATION

The application of economic theory, and game theory in particular, to computer science problems such as resource allocation has grown into a mature field of research over the past two decades (Bogan, 1994). The problem of allocating scarce resources among a set of distributed agents is the very problem faced by a market economy. A very large body of economic research exists on the functioning of markets, so they are well understood. A central result of economic theory is that, given proper conditions, a market will produce an optimal distribution of resources with a minimum of transaction costs, which are analogous to communication resources in the computational problem.
In adapting this work to the problem of optimizing resources in a sensor network, we can exploit the results of economic theory in order to design multi-agent systems that produce optimal resource allocations. Markets have buyers and sellers of goods and services, so we must formulate the sensor resource optimization problem in these terms in order to take advantage of market-based solutions. This is easily done using the multi-agent framework described in the previous section. Agents represent both the buyers and the sellers in our artificial economy, and the goods and services for sale are the assets controlled by those agents, which include computational resources and information collected by sensors. Our multi-agent system simulates a marketplace where these agents exchange their services. By finding the competitive equilibrium of this artificial economy, we can solve the resource allocation problem and ensure optimal usage of our sensor resource system.

Market-based optimization, using a particular negotiation/bidding mechanism, allows the agents to reach this equilibrium. The negotiation mechanism, or protocol, defines the rules used by agents in conducting transactions. It determines how agents make bids for the services of other agents as well as how agents communicate with one another. In addition to following these rules for negotiation, each agent uses a negotiation strategy that determines how that agent bids. Negotiation strategies are based on the results of game theory. Game theory examines how individuals make decisions when they know that their actions affect other individuals and when they assume that other individuals also take this into account. For example, if two agents are bidding on the use of a shared antenna, they will formulate their bids not only based on how they value that antenna but also on how they think the other agent values that antenna. They do this because, in order to win the bidding war, they do not necessarily have to bid as high as they believe use of the antenna is worth; they must only bid higher than the other agent who is also bidding. The results of game theory allow bidders to maximize their own utility in a competitive marketplace; when everyone follows these strategies, the market as a whole is optimized.2,3

In our implementation, each agent has the ability to calculate the utility of its possible actions. ("Utility" is the economic term for the value of an action or situation.) System agents can calculate the utility of buying information from sensors, sensor agents can calculate the utility of buying computational resources and of selling sensor information, and resource agents can calculate the utility of selling computational resources. Using these utility calculations, agents carry out strategies for formulating and accepting bids. By using the results of game theory, these strategies can be optimized in order to produce the best possible aggregate outcome. Our in-house SAMPLE tool for building intelligent, multi-agent systems provides every agent with capabilities for reasoning based on Bayesian belief networks and on rule-based expert systems. Both of these approaches are appropriate for encoding optimal negotiation strategies, so one task of future work will be to determine which of these options is better suited to reasoning in negotiation.

Market-based mechanisms using negotiation for resource allocation have been used to solve problems in a variety of domains. A number of efforts have used this approach to allocate computational resources for distributed computing.4,5,6,7 Others have used the method for information retrieval from digital networks.8,9,10 More recently, the technique has become widespread in solving just-in-time manufacturing control problems.11 Finally, negotiation has been successfully applied to the military collection management domain for multi-sensor multi-target tracking.12 The success of negotiation in these fields, all of which share significant similarities with the real-time sensor optimization problem under examination here, leads us to believe that we can achieve the same degree of success in this target tracking domain.

As mentioned above, if the negotiation mechanism is designed properly, it will produce optimal resource allocations. One major assumption on which this rests is that agents can accurately calculate the utility of their actions. In the case of sensor network optimization, this means accurately calculating the value of the information that sensors can provide. In a way, this is a question of properly prioritizing sensor tasks, because the greater a task’s priority, the greater should be its utility. The promise of negotiation as a means of optimization is that, as long as this step is correct, it guarantees that the results will be optimal. Utility functions may be as simple or complex as is necessary in order to be accurate. Thus, they could range from simple mathematical functions to complex chains of reasoning. Again, SAMPLE agent technology is helpful here, because it provides both Bayesian belief networks and rule-based expert systems that can be used for more complex modeling of utility functions. Moreover, the communication interface available to each agent allows it to obtain feedback from signal processing algorithms to aid in calculating the value of various sensor tasks. Another important part of our future work will thus be to examine alternative methods of augmenting utility calculations and determining which provides the best way of calculating the value of the available sensor information. This includes exploring the use of
libraries of utility functions that use situation assessment (guided by belief networks and/or expert systems) to
dynamically select appropriate methods of utility evaluation, based on the current situation and particular information
requirements.

The other major assumption of the negotiation mechanism is that, in order to generate an optimal allocation of resources,
the mechanism itself must have sufficient computational resources with which to execute in order to maintain
competitive equilibrium in the face of real-time changes in the mission environment. In order to accomplish this, our
negotiation algorithm must be as efficient as possible. Game theory allows us to minimize the communication overhead
needed for negotiation by designing bidding protocols that resolve the negotiation in as few steps as possible. A variety
of such protocols already exist and have been tested in a number of application areas.13 We have tried to design
algorithms that maximize the efficiency with which they use available computational resources so that the system is fast
enough to maintain optimal allocation in real-time.

There are a number of alternative methods of implementing this multi-agent negotiation system across the sensor
network. Choosing among these options is important for realistically simulating the deployment of our optimization
strategies. There are two obvious alternatives: One is to implement it as a truly distributed system, where the agents
reside with and directly control the physical entities they represent. The other is to maintain the agents on a centralized
system that sends tasking orders out to the sensor network itself. The first approach relies more heavily on on-board
processing, while the second relies more heavily on communication, so there are certainly tradeoffs in these approaches.
The best approach might be a hybrid combination of these two methods. However, regardless of what type of
implementation is ultimately required for deployment, the flexible, modular nature of SNOMAN and its ability to
function over networks of sensors allow it to work without modification in any of these contexts. This is a major source
of value in our distributed approach to optimization.

SNOMAN constructs a set of agents that represent all the elements of a sensor network, using three different agent types:

- **Sensor agents** represent sensors themselves as well as the platforms that carry them. Examples include passive
  sensors such as IR and UV sensors, active sensors such as LADAR, and sensor platforms such as satellites. The
  main purpose of sensor agents is to collect information. In order to do this, they must obtain access to the
  computational resources needed to carry out their tasks and in some cases they must cooperate with the agents
  representing the platforms on which they reside. They respond to requests for information from engagement
  systems (e.g. command and control) and other external systems.

- **Resource agents** represent computational resources including processing nodes and communication resources
  such as network bandwidth. Resource agents provide the computational power that sensors and other
  components in the sensor network need to complete their tasks. They respond to requests from sensor and
  system agents and at times they must also coordinate among themselves in order to ensure optimal resource
  allocation.

- **System agents** represent engagement systems as well as other external systems that might require data from the
  sensor network. System agents work on behalf of these systems to obtain the requested information from sensor
  agents. They must also interact with resource agents when they need to carry out tasks such as signal
  processing.

3. JUST-IN-TIME MARKETS FOR HETEROGENEOUS GOODS

Past market-based optimization algorithms have relied on traditional general equilibrium markets. The most prominent
of these is Wellman’s WALRAS market, named for Leon Walras, the economist who first developed general equilibrium
in great detail.10 WALRAS is a seller-driven market, where agents who have valuable goods put them up for sale and
wait for bids from buyers. Such a market begins each step with the auctioneer publishing a vector of past prices to both
buyers and sellers. Following this, the sellers send out a message containing a vector of all the goods they have available
for sell. Buyers then reply with a vector of bid functions that describe how many goods they desire at each possible price.
The auctioneer, which plays a central role in WALRAS markets, then adjusts the market until the market “clears,”
meaning that there is no excess supply or demand and the buyers and sellers have an equal amount of money to exchange
for goods. The market is thus forced to clear all transactions simultaneously. After this happens, the auctioneer informs
buyers of the contracts or goods they succeeded in winning, and the market then begins anew at the next time step. A UML sequence diagram of this process is shown in Figure 1.

Figure 1: UML Sequence Diagram for WALRAS Market

This kind of WALRAS market has a number of obvious disadvantages in the context of the sensor management problem described in this paper. First and foremost, it assumes that goods are homogeneous, meaning that buyer agents simply want one good from a pool of interchangeable goods. This assumption is acceptable for resource markets such as distributed computing, where processing time is the same no matter who provides it. In a sensor market, on the other hand, a large variety of different kinds of information (and different levels of information quality, based on sensor type) are available. In purely computational problems, this obstacle can be overcome by increasing the size of the market (and possibly running separate markets for each type of good). However, in markets such as ours where the goal is to model physical systems of sensors, we cannot simply make the market arbitrarily large, because it must reflect available resources.

Another problem with applying the WALRAS market to our domain is that it forces the market to clear at a “frozen” state. In other words, it must treat the market as static at a particular moment in time in order to balance supply and demand across all buyers and sellers. In our sensor market, requests for information are dynamic and must be allocated immediately, so we require a market environment that can handle a large number and variety of asynchronous information requests. Moreover, the seller-driven nature of the WALRAS market creates a related problem. A sensor market cannot be seller driven, because at any given time each sensor could potentially provide a wide variety of information, most of it useless to buyers in the system. Instead, we require a market that is driven by the buyers, the agents who need information.
To address these issues, we have designed a new protocol called the just-in-time (JIT) market, which is a buyer-driven market that incorporates heterogeneous goods and dynamic task allocation. Figure 2 shows a UML sequence diagram of the JIT market protocol.

In the JIT market, a buyer begins by creating a task request. Each seller then examines the task and checks it against its own capabilities to see if it can perform the task. If so, it determines whether to accept or reject the task at a given bid level. The market assigns tasks based on bids. However, unlike the WALRAS market, the JIT market does not close the bidding for an individual task until the time the task must be performed. In other words, it waits as long as possible to ensure that no better (more efficient) allocations come along, and then closes the market “just in time” to carry out the task. Until the market closes, assigned tasks can be bought and resold by broker agents, whose purpose is to search the market for inefficiencies.

### 4. THE TRANSITIVE SELFISH SELLER AGENT PROBLEM

Inefficiencies exist whenever a broker agent could buy and sell a series of tasks from other agents (essentially re-assigning them) while turning a profit. The existence of this potential profit is called an arbitrage opportunity, and such opportunities reflect transactions where tasks were assigned and those assignments later became inefficient, either because new, more valuable tasks appeared or because other, better-equipped agents became available to fulfill those tasks. In either case, the arbitrage agents make the trades necessary to eliminate these inefficiencies and bring the market back into an optimal equilibrium.

Figure 3 depicts a situation where an arbitrage opportunity exists. The figure depicts a market with three sellers (A, B, and C) and three initial tasks (Tasks 1-3). Lines in the graph between tasks and sellers indicate that the task is within the
capability of that seller. Lines with arrows indicate that a task has been assigned to a particular seller. Here, Task 1 is assigned to Seller A, Task 2 is assigned to Seller B, and Task 3 is assigned to Seller C. Values along the lines with arrows indicate the utility (price) of the assigned task. Task 4 represents a task that has appeared after the initial allocation of tasks had been carried out. Task 4 has a potential value of 5, and it is clear that the total net utility of the outcome (the value of all completed tasks) could be improved by carrying out Task 4 instead of Task 1, which only has a value of 3.

![Net Utility = 18](image)

**Figure 3: Arbitrage opportunity due to market inefficiency**

The problem is that only Seller C is capable of handling Task 4, and there is no incentive for it to take on the task, since it already has a higher-valued task (Task 3). Sellers only pursue their own immediate interests, so in situations like this where a series of trades would be required in order to improve the global utility, those trades will not occur. To solve this problem, we introduce arbitrage agents who look for inefficient situations like this and buy and sell tasks until the optimal allocation results. The arbitrage agents have a profit incentive, because they can carry out a series of trades and claim the surplus utility created by the trades as profit. Figure 4 presents the optimal allocation of tasks after an arbitrage agent has made a series of exchanges.

Three trades must occur for this outcome to result. First, Seller C must sell Task 3 to Seller B. Second, Seller B must sell Task 2 to Seller A, with A dropping Task 1. Finally, Seller C will then buy Task 4, and then net utility of the overall allocation has increased by 2. Because there is no incentive for these trades to occur directly, they must be facilitated indirectly by an arbitrage agent, who buys rights to the necessary tasks and then offers them to the appropriate Sellers at prices such that the Sellers do no worse than they would have before and the arbitrage agent profits from the overall set of transactions. The case shown here is relatively simple, but the notion of arbitrage agents, who build graphs of allocated transactions and look for inefficiencies that they can exploit, is useful for far more complex problems and can be used to ensure the efficiency of the market even in the face of a high level of rapid change.
Third-party agents are useful in a variety of other contexts as well, such as information requests that require coordinated action from multiple sensors. In this case, a broker agent can be assigned to purchase bundles of goods needed to satisfy the simultaneous requirements. This is one of the main advantages of multi-agent systems: behaviors that can be very simple at the level of individual agents yield results that aggregate into complex and efficient equilibrium outcomes.

5. CONCLUSION

In this paper we have developed two extensions to traditional market-based optimization approaches. First, we provide a new market protocol called the just-in-time (JIT) market that incorporates heterogeneous goods that must be allocated dynamically. This allows market-based algorithms to be extended to new, more complicated domains that do not rely on the assumption of homogeneous goods. Second, to accommodate some of the additional complexities caused by our dynamic JIT market, we have developed a particular kind of arbitrage agent that ensures our market remains at an efficient equilibrium even as new tasks appear and objectives change. We apply this approach to the problem of managing sensors and allocating their supporting resources in the domain of target tracking.

ACKNOWLEDGEMENT

The authors thank Amy Sunshine Smith-Carroll from the Navy Surface Warfare Center – Dahlgren for support on this project.

REFERENCES