In *Proceedings of the AAAI Spring Symposium on Intelligent Embedded and Distributed Systems*
pages 21-26, Palo Alto, California, March 25-27, 2002

A market-based formulation of sensor-actuator network coordination

Brian P. Gerkey   Maja J Matarić
Computer Science Dept.
University of Southern California
Los Angeles, CA 90089-0781, USA
{bgerkey|mataric}@cs.usc.edu

1 Introduction

Over half a century ago, rebelling against the prevalent economic theory of his time, Friedrich Hayek (1945) argued that the success of the price-based market system is due entirely to the inherently distributed manner in which in it operates:

> If we can agree that the economic problem of society is mainly one of rapid adaptation to changes in the particular circumstances of time and place, it would seem to follow that the ultimate decisions must be left to the people who are familiar with these circumstances, who know directly of the relevant changes and of the resources immediately available to meet them (p. 524).

That is, the system operates most efficiently when the individuals involved act autonomously based on local information. Given the scale at which such economies operate, the problem of coordinating the actions of all the participants would be intractable without this compartmentalization of data and control.

However, it is not the case that there is no communication at all, for without any transfer of information there could be no coordinated activity in the system. Rather the participants in the economy communicate constantly, but through a remarkably efficient medium: *price*. All information regarding the availability and value of a resource is compressed into a single linear comparator, allowing rational efficient decisions regarding economic transactions. For example, when buying an avocado at the grocery store, we may not know that a drought recently devastated many crops or that avocados are extremely popular right now, but we do know that the store is charging $5 per avocado and we make our buying decision based on that information. We need not take into account the complex (and largely irrelevant) market forces behind the price.

As we search for design methodologies that allow coordination in distributed sensor-actuator networks of a scale comparable to human economies, we take inspiration from the price-based market model. The problems central to economics and synthetic network control are extremely similar. In fact, the fundamental issue in both areas is *resource allocation*: how should the available resources be distributed among the members of the group?

A real-world economy is composed of a large number of individuals and groups, each with different needs, capabilities, and resources. Information regarding the state of the participants is highly distributed and at the very least uncertain, if not contradictory. People act independently, out of self-interest, and are limited in their capacity to acquire and process information in order to make decisions. Yet the result of their collective selfishness is a globally coherent system in which resources are efficiently allocated according to the supplies of and demands for the resources being traded. Further, in large part because of the scale at which they operate, real-world markets are remarkably tolerant of localized failures and adapt quickly to changes in both the makeup of the society and the state of the environment.

Similarly, a sensor-actuator network is composed of a large number of heterogeneous nodes, each possessing different resources and offering different services. Information regarding the state of the network is distributed throughout and is fraught with uncertainty that arises from sensor and actuator noise and environmental change over time. The nodes are autonomous and are constrained in communication bandwidth and processing power. The goal is to coordinate the actions of these nodes such that resources are allocated in a way that enables the execution of complex tasks. Also, the network must be robust to node faults and in general exhibit fast dynamic response to change. I propose to achieve these goals by implementing SANs as synthetic market systems in which the nodes trade resources with each other in order to increase their own wealth.

In this paper we expound on this economic analogy by constructing a market-based model of sensor-actuator networks and showing how it can be used to achieve scalable coordination. The rest of this paper is organized as follows: Section 2 formally states the coordination problem, Section 3 justifies a market-based solution, Section 4 explains how performance is measured, Section 5 describes our market model, and Section 6 concludes with a discussion of future work.

2 Problem Definition

The question with which we are concerned is the following: how can coordinated, task-oriented behavior be elicited from multi-purpose physical sensor-actuator systems? We now formalize this question, first with some underlying definitions, then with a statement of the problem.

Definition 1 Sensor-actuator node (or simply *node*): *A computer that is physically embodied in its environment through a set of sensors and/or actuators, referred to collectively as* resources.

Definition 2 Sensor-actuator network (SAN): *A set of communicating nodes.*

A node's surrounding environment is the noisy, uncertain, non-stationary physical world, and the resources are physical devices. As a result, the use of a node's resources incurs a monetary cost in the real world (e.g., the retail price of a camera, the energy cost of operating a motor). Each node is autonomous and locally controls its resource set. The nodes are also endowed with communication facilities. Every node may communicate with every other node, but bandwidth is limited and messages may be lost (though the communication channel is in general quite reliable)[1]. Using their resources, nodes can provide various *services*:

Definition 3 Service: *a generic capability of one or more nodes.*

In its simplest form, a service will constitute a single action performed by an individual node; more generally, it will constitute a coordinated, time-extended series of actions performed by a collection of nodes. Two examples of a service are transportation of an object and mapping of an area. The specification of a service is independent of its implementation; thus a service may be implemented in several different ways. Together, the set of possible services S constitutes the common "language" through which the nodes communicate. These services are offered by *providers*:

Definition 4 Service provider (or simply *provider*): *one or more nodes that offer a service.*

Each provider $p_i \in P$, the set of available providers, is a subset of the set of nodes N. That is, a service may be provided by a single node or by a group of nodes, also known as a *coalition*. A node $n_i \in N$ can be a member of more than one provider; n_i is generally a member of at least the solo provider $\{n_i\}$ (assuming that n_i can provide some service(s) on its own). Being generic, a service cannot be directly executed; rather it must be made into a specific *task*:

Definition 5 Task: *A task t_i is a parameterized instance of a service $s_j \in S$, denoted by $t_i \sim s_j$.*

For example, a generic object transportation service can be made into a task of the form "transport the orange ball from Room A to Room B now." For evaluation purposes, the performance of each task t_i can be measured by a metric function m_i:

Definition 6 Metric function (or simply *metric*): *a function that maps from a set of task-specific parameters to the real interval $[0, \infty)$.*

See Section 4 for details on how metrics are determined. Every provider is capable of estimating its own performance for a given task, both before and during task execution. These estimates are not perfect, but should provide predictions of performance that are as accurate as possible.

The Problem Given is N, a set of nodes that comprise a sensor-actuator network, and S, a set of services that the network can provide. Also given is the provider structure; that is, the set of providers P is predefined such that every node $n_i \in N$ knows three things:

1. the set of providers of which n_i is a member
2. the identities of the other members (if any) of these providers
3. the set of services offered by each of these providers.

Given an infinite sequence of tasks T that arrive stochastically over time, the problem is to assign the tasks T to the providers P so as to maximize:

$$\sum_{t_j \in T} m_i : t_j \sim s_i$$

That is, the goal is to find allocations that optimize the performance of the SAN on the tasks.

[1]This specification of a communication system roughly models IEEE 802.11b wireless Ethernet (IEEE 1999).

Due to the non-stationarity of both the external state of the world and the internal state of the nodes, any static allocation can easily fall into temporally local extrema of performance[2]. Thus any effective task allocation policy must be dynamic in nature and consider re-allocations.

3 Justification for a Market-based Solution

Any approach that is proposed for solving this task allocation problem requires justification. It should be clear why existing approaches are insufficient and why the new approach is expected to be effective.

Variants of the embodied coordination problem have faced researchers since the advent of multi-robot systems (e.g., Matarić (1992), Parker (1994)). While a number of ad-hoc coordination strategies have been proposed in the robotics literature, an effective, efficient, general-purpose approach to task allocation is yet to be seen. Among those approaches that have been validated, there are indeed interesting examples of coordination. However, few, if any, of those systems have ever been re-used and there still does not exist any consensus in the community as to how good coordination mechanisms should be designed or implemented for embodied domains. Thus, from the point of view of robotics in particular and embedded systems in general, group-level general-purpose task allocation is an open problem. A great deal of work on related problems in task and resource allocation and teamwork has been done in the software multi-agent community (e.g., Shehory & Kraus (1995), Sycara, Decker, Pannu, Williamson et al. (1996), Walsh & Wellman (1998), Martin, Cheyer & Moran (1999)), sometimes using the software environment as a substitute for the physical world (e.g., Tambe (1997), Modi, Jung, Tambe, Shen et al. (2001)). As non-experts with respect to these approaches, we are not in a position to decry them as ineffectual in embodied domains. In fact, any of these (and other) methods could eventually be quite successful in coordinating physical systems. However these software agent approaches (in general) have not yet been physically validated at all and it is not clear whether or when that validation will occur.

As for why one should expect our proposed market-based mechanism to be effective for the embodied task allocation problem, there are several reasons. There is, of course, the analogy given in Section 1: people constantly and efficiently allocate scarce resources using various forms of the price-based market, so artificial systems should do the same. However, an analogy on its own is an insufficient justification and can, in fact, be misleading. Thus it is important to support the analogy with more convincing evidence. First, markets are extremely *practical* for use in large-scale physical systems; the market protocols themselves are inherently distributed computationally and informationally, and the price information by which individuals communicate is compact. Second, in our prior work with multi-robot systems (Gerkey & Matarić 2001*a*, Gerkey & Matarić 2001*b*, Gerkey & Matarić 2002), we have shown that MURDOCH, a simple greedy market-like mechanism similar to the Contract Net Protocol (Smith 1980), can robustly and effectively allocate tasks in dynamic environments, albeit with the danger of falling into temporally local minima of efficiency. Third, there is a significant body of microeconomic theory dealing with *general equilibrium* that can provide certain guarantees about the quality of allocation solutions derived from properly constructed (non-greedy) market systems (see Section 5 for details). Finally, there is the growing body of work on computational markets, which shows that they can be used to solve many different resource/task allocation problems (e.g., Kurose & Simha (1989), Gagliano, Fraser & Schaefer (1995)); however, aside from our own work with MURDOCH, market mechanisms have yet to be studied and empirically validated in embodied domains, which is our current goal.

4 Measuring Performance

Objectively evaluating solutions to the given task allocation problem is non-trivial, but such evaluation is necessary in order to compare the "goodness" of different allocations. As stated in Section 2, the overall performance of an allocation will be determined as a sum[3] of the performance values of the individual tasks, both past and current.

Having decided how to combine individual task performance values, there remains the problem of determining those values. We suggest that in this domain, performance in the context of an single task has two components: *quality* and *cost*. Quality is a measure of how well the task is executed. For example, in an object transportation task, quality might be the time required to transport the object or the accuracy with which it is transported, given some target location or trajectory. Cost is a measure of the resources used in achieving the task. As mentioned earlier, the sensors and actuators that are used to do work in this domain have associated with them real costs in the physical world. Returning to the object transport task, cost

[2] We make the reasonable assumption that perfect knowledge about future states of the world is unavailable.

[3] Instead of a simple sum, overall performance could in general be any arbitrary function of individual task performances. For example, if there exists a known priority among the tasks, then a weighted sum might be a better measure of overall performance.

might be the amount of energy required to power the motors that drive the transporting robot's wheels.

Unfortunately for the goal of evaluation, we are now faced with a multi-objective problem: maximize quality and minimize cost. Quality and cost can easily be in direct conflict, such as when an object can be transported more quickly but more expensively by a large rocket-powered robot than by a small battery-powered robot. In general, the system evaluator must make a decision as to the relative importance of quality and cost in measuring performance. At one extreme, pure cost-minimization suggests the solution in which no tasks are allocated and thus no resources are used. Pure quality-maximization will result in a better but still undesirable solution in which tasks are well executed, but without regard for the costs imposed on the external world, which may play a very real role in the long-term survival and operation of the system.

Clearly there are many possible schemes for mixing quality and cost between these two extremes. Although we are exploring this space to some degree we do not attempt to define the best possible mix. Rather we look for performance measures that combine quality and cost in such a way that the results are intuitive and meaningful for the particular task and domain. By combining these individual measures, we can construct an objective, external metric of system-level performance for use in experimental comparison of different allocation mechanisms (see Section 6).

5 The Market Model

5.1 Background

MURDOCH (Gerkey & Matarić 2001*b*), being a single-shot task allocator similar to the Contract Net Protocol (Smith 1980), is only loosely based on economic ideas and suffers from well-known problems of efficiency common to such myopic, greedy systems. By contrast, we now propose a market formulation of the task allocation problem described in Section 2 through the application of *general equilibrium theory* (Varian 1992). In an economy comprising multiple interconnected markets, general equilibrium (or Walrasian equilibrium) is the state in which supply and demand are in balance simultaneously in all markets, given utility-maximizing behavior on behalf of all participants. Such an equilibrium is described by a vector of prices assigned to the markets. An important and appealing characteristic of a general price equilibrium is that the resultant resource allocation is Pareto efficient; that is, to better the situation of one individual will necessarily worsen the situation of another individual. When solving multi-agent problems (especially in economics), Pareto efficiency is often the stated goal, because it is "best" in the sense that it is not possible to improve upon a Pareto efficient solution without making relative value judgments about the individuals involved.

Given the goal of general equilibrium, a method is required to find it. The original proponent of general equilibrium, the French economist Walras (1954), suggested that it could be achieved through an iterative price-adjustment process, which he termed *tatônnement*. The idea is as follows (as summarized by Cheng & Wellman (1998)):

> Starting with a set of prices arbitrarily given, the excess demand in each market may be positive, zero, or negative. For an arbitrary ordering of these markets, take the first and adjust the price so that supply and demand are equal, given all other prices. Of course, the change in the first price will normally change excess demand in all other markets. Next consider the second market and likewise adjust its price to clear; then repeat the process for all remaining markets. At the end of each round, only the last market is guaranteed to be in equilibrium, since a change of price in some later markets will normally destroy the equilibria established in previous markets. But Walras argued that the change in a good's own price will have a more significant impact on its excess demand than the change in other goods' prices. The own price adjustment goes directly to zero excess demand, whereas the indirect influences of other price changes may increase or decrease demand for the good, and may even cancel each other out. Hence, Walras argued, it is probable that after each round, the prices are closer to equilibrium than before. Eventually, in this story, all markets will clear.

The existence and uniqueness of general equilibria in markets has been studied by economists at great length since Walras's time. In brief, a general equilibrium will exist if individual preferences are convex and continuous. A general equilibrium will be unique if the goods being traded in the markets are *gross substitutes*; that is, an increase in the price of one good increases (or at least does not decrease) the demand for other goods.

5.2 Model

Using general equilibrium theory, we are constructing a market-based model that describes the embodied allocation problem given in Section 2 and simplifies its solution. Construction of the model will be an ongoing and iterative process; thus the exposition given here is necessarily preliminary. Further analysis and experimentation will suggest modifications to the model, as well as answers to the list of remaining questions given in the next section.

In modeling the problem, we treat the nodes in a sensor-actuator network as competitive[4] self-interested individuals participating in an economy of task contracts. The economy is composed of multiple markets, one for each available task contract. The providers vie for the contracts by auction, submitting bids based on their expected performance. For the purposes of this model, performance values are formulated such that they range from 0, the best, to ∞, the worst. Thus, although they incorporate measurements of quality, performance values can be treated like costs, which are incurred by providers when executing tasks. After task execution, a provider is paid a sum of money in compensation, and the provider's profit is the difference between that payment and its performance. Task contracts are auctioned according to the following algorithm:

1. An initial vector of prices representing the amount of money to be paid for the successful completion of each task in the set of currently available tasks T is distributed to the providers.

2. Each provider $p_i \in P$ considers its possibilities. For each task $t_j \in T_i$, where T_i is the subset of T that contains only those tasks of which p_i is capable, p_i estimates its performance, building a vector of performance values. By subtracting this performance vector from the given price vector, p_i determines its expected payoff for each task. In an attempt to maximize profit, p_i chooses the task with the highest expected payoff and bids for that task.

3. Prices are adjusted in a tatônnement-like way: when more than one provider is willing to execute a task (i.e., demand exceeds supply), the price is lowered (in an effort to find the lowest, or best performing, bidder), and vice versa. The new price vector is redistributed to the providers.

4. Steps 2 & 3 are repeated until equilibrium is reached, at which time the markets are cleared, with each task allocated to a provider (assuming that the network contains sufficient resources).

The resulting allocation will be Pareto-efficient, subject to uncertainty in performance estimation.

5.3 Remaining Questions

We have been purposefully vague regarding several important issues of this model and algorithm, for we are currently investigating how to best resolve them. We present these issues now as questions, with preliminary answers.

When is an auction held? An auction should be held whenever there is a high probability that a re-allocation of tasks will result in an increase of system performance. Although it might be possible, in principle, to build a model of the world in order to determine this probability formally, we will employ a simple and intuitive event-driven scheme. A new auction will be called as a result of the occurrence of any "important" event, the set of which will include at least: the arrival of a new task and the completion of an old task. Other important events could be passage of time and lack of progress on one or more tasks.

Who acts as auctioneer? In principle, any node can be auctioneer at any time; in practice there will likely be a set of designated and well-known auctioneer nodes. Since computation and communication will not scale if a single node simultaneously handles all allocations, the markets themselves will be distributed among multiple auctioneers when the economy contains many tasks.

How are the initial price vectors determined? Although Walras imagined them as random, initial prices could be purposefully biased, for at least two reasons. On the one hand, if the auctioneer can somehow estimate the final price for a task, then setting the initial price accordingly will speed convergence to equilibrium, thus shortening the time required to complete the auction. Alternatively, if there exists a relative priority among the tasks, initial prices could be skewed so that more money is offered for higher-priority tasks.

Who participates in an auction? Not every provider will participate in every auction, for two reasons. First, in order to scale to very large systems, auctions will generally be confined to a subset of physically collocated providers, because, considering the energy cost and positional uncertainty involved in long-distance movements, far-away providers are unlikely to perform well. Second, a provider may choose not to participate in an auction if it is heavily invested in (and making progress at) its own task. Should an already-engaged provider decide to participate in an auction, it must consider its sunk investment in its current task when estimating performance on other tasks[5].

Who monitors task progress? Progress monitoring will be essential for good system performance, and each provider will monitor its own progress through re-estimation of performance. However, since nodes are subject to fault and failure, progress should also be monitored

[4] In an economic context, a *competitive* individual is one who takes price information as given, as opposed to a *strategic* individual, who speculates on the effects of his own actions on prices.

[5] Computational tasks in Kurose & Simha's (1989) system make similar decisions.

externally. As for who performs this external monitoring, several schemes are possible, such as a predefined set of dedicated monitors or an egalitarian peer-to-peer system.

How are multi-node providers handled? How does a node avoid over-commitment? We have designed a simple 4-step negotiation protocol that occurs among the nodes in the provider (a sort of "huddle") that solves these problems. For brevity, we leave out the details here.

What about money? Can it be accrued? In the current model, nodes cannot purchase anything with money, and determining a node's performance by examining its accrued wealth involves somewhat circular logic. Thus there might in fact be no advantage in allowing nodes to accrue money.

6 Conclusion & Future Work

In this paper we have formally defined an interesting and open research problem: task allocation in sensor-actuator networks. We have motivated an economic approach to this problem and given our own preliminary market formulation based on general equilibrium theory. We are currently implementing this model, and our next step will be to validate it experimentally.

In order to validate our market model, we plan to perform experiments with various sensor-actuator networks, both simulated and real. Initially, we will pursue a multi-robot foraging domain. Foraging experiments will be carried out exclusively in simulation, allowing us to focus on scaling. Furthermore, it is in simulation where we will be best able to implement and test non-market allocation mechanisms for comparison purposes. Subsequently, we will move to the physical world and apply our market approach to task allocation in a real instrumented environment, consisting of many mobile robots and fixed sensor nodes. Using physical hardware, we will focus on the effects of uncertainty, noise, and non-stationarity in the environment. In both domains, we will perform experiments and explore model parameterizations that address central issues, especially with regard to when and why market-based coordination is applicable to embodied task allocation.

References

Cheng, J. Q. & Wellman, M. P. (1998), 'The WALRAS algorithm: A convergent distributed implementation of general equilibrium outcomes', *Computational Economics* **12**(1), 1–24.

Gagliano, R. A., Fraser, M. D. & Schaefer, M. E. (1995), 'Auction allocation of computing resources', *Communications of the ACM* **38**(6).

Gerkey, B. P. & Matarić, M. J. (2001*a*), Principled communication for dynamic multi-robot task allocation, *in* D. Rus & S. Singh, eds, 'Experimental Robotics VII, LNCIS 271', Springer-Verlag, Berlin, pp. 353–362.

Gerkey, B. P. & Matarić, M. J. (2001*b*), Sold!: Market methods for multi-robot control, Technical Report IRIS-01-399, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California.

Gerkey, B. P. & Matarić, M. J. (2002), Pusher-watcher: An approach to fault-tolerant tightly-coupled robot coordination, *in* 'Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)', Washington D.C., pp. 464–469.

Hayek, F. A. (1945), 'The Use of Knowledge in Society', *The American Economic Review* **35**(4), 519–530.

IEEE (1999), *Wireless LAN Medium Access Control and Physical Layer Specifications*. IEEE Std 802.11b-1999.

Kurose, J. F. & Simha, R. (1989), 'A microeconomic approach to optimal resource allocation in distributed computer systems', *IEEE Transactions on Computers* **38**(5), 705–717.

Martin, D. L., Cheyer, A. J. & Moran, D. B. (1999), 'The open agent architecture: A framework for building distributed software systems', *Applied Artificial Intelligence* **13**(1), 91–128.

Matarić, M. J. (1992), Designing Emergent Behaviors: From Local Interactions to Collective Intelligence, *in* J.-A. Meyer, H. Roitblat & S. Wilson, eds, 'From Animals to Animats 2, Second International Conference on Simulation of Adaptive Behavior (SAB-92)', MIT Press, pp. 432–441.

Modi, J., Jung, H., Tambe, M., Shen, W. et al. (2001), A Dynamic Distributed Constraint Satisfaction Approach to Resource Allocation, *in* 'Proc. of the Intl. Conf. on Principles and Practices of Constraint Programming', Paphos, Cyprus.

Parker, L. E. (1994), ALLIANCE: An Architecture for Fault Tolerant, Cooperative Control of Heterogeneous Mobile Robots, *in* 'Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)', Munich, Germany, pp. 776–783.

Shehory, O. & Kraus, S. (1995), Task allocation via coalition formation among autonomous agents, *in* 'Proc. of the Intl. Joint Conf. on Artificial Intelligence (IJCAI)', pp. 655–661.

Smith, R. G. (1980), 'The Contract Net Protocol', *IEEE Transactions on Computers* **29**(12), 1104–1113.

Sycara, K., Decker, K., Pannu, A., Williamson, M. et al. (1996), 'Distributed intelligent agents', *IEEE Expert* **11**(6), 36–46.

Tambe, M. (1997), Agent architectures for flexible, practical teamwork, *in* 'Proc. of the Natl. Conf. on Artificial Intelligence (AAAI)', Providence, Rhode Island.

Varian, H. R. (1992), *Microeconomic Analysis*, 3^{rd} edn, W. W. Norton & Company, New York.

Walras, L. (1954), *Elements of Pure Economics*, Augustus M. Kelley Publishers. Originally published in 1874.

Walsh, W. E. & Wellman, M. P. (1998), A Market Protocol for Decentralized Task Allocation, *in* 'Proc. of the Intl. Conf. on Multi Agent Systems (ICMAS)', Paris, pp. 325–332.