

Resource Allocation via Message Passing

Ciamac C. Moallemi

Graduate School of Business, Columbia University, 3022 Broadway, Uris 416, New York, NY 10027, USA,
ciamac@gsb.columbia.edu

Benjamin Van Roy

Department of Management Science & Engineering, Stanford University, Terman 315, Stanford, CA 94305,
USA, bvr@stanford.edu

We propose a message-passing paradigm for resource allocation problems. This serves to connect ideas from the message passing literature, which has primarily grown out of the communications, statistical physics, and artificial intelligence communities, with a problem central to operations research. This also provides a new framework for decentralized management that generalizes price-based systems by allowing incentives to vary across activities and consumption levels. We demonstrate that message-based incentives, which are characterized by a new equilibrium concept, lead to system-optimal behavior for convex resource allocation problems, yet yield allocations superior to those from price-based incentives for non-convex problems. We describe a distributed and asynchronous message-passing algorithm for computing equilibrium messages and allocations, and demonstrate its merits in the context of a network resource allocation problem.

Key words: message-passing algorithms; min-sum algorithm; resource allocation; convex optimization; decentralized optimization

History: Initial version June 2007; submitted November 2008; revised June 2009, March 2010; accepted March 2010.

1. Introduction

Message-passing has emerged as an active research topic in a number of fields, including communications, statistical physics, artificial intelligence, probability theory, statistics, and signal processing. Interest was to a large extent triggered by the success of “turbo decoding” (Berrou et al., 1993; Benedetto et al., 1996; Frey, 1998; Richardson and Urbanke, 2001b), a message-passing algorithm now used routinely in communication systems. The decoding problem it aims to solve is NP-hard, and it was a surprise that this simple and efficient algorithm for all practical purposes solved the problem. Separately, inspired by ideas from

statistical physics, message-passing algorithms have been proposed for solving certain difficult combinatorial optimization problems such as satisfiability and graph coloring (Mézard et al., 2002; Braunstein et al., 2005; Aurell et al., 2005; Braunstein and Zecchina, 2004; Maneva et al., 2005). There has been much work dedicated to understanding whether message passing can be useful in other contexts. For example, a body of empirical work in various problem domains demonstrates promise in areas such as data-mining (e.g., McCallum, 2003; Pinto et al., 2003), computer vision (e.g., Freeman et al., 2000; Coughlan and Ferreira, 2002; Sun et al., 2002), and bioinformatics (e.g., Leone and Pagnani, 2005; Letovsky and Kasif, 2003; Sutton et al., 2004).

Message-passing algorithms can be applied to a wide variety of NP-hard optimization and inference problems. Further, their structure makes them amenable to distributed computation. One may be tempted to dismiss message passing as yet another ad hoc, heuristic method. However, for some significant problems such as decoding and satisfiability, message-passing algorithms represent the state-of-the-art method of solution. Moreover, message passing bears mathematical structure that has made it useful as an analytical tool. For example, analytical methods based on message passing have shed light on the asymptotic performance of certain classes of codes (Richardson and Urbanke, 2001a; Montanari et al., 2005; Montanari and Tse, 2006) and led to solutions and structural insights for certain large-scale combinatorial optimization problems (Aldous, 1992, 2001; Aldous and Steele, 2003; Talagrand, 2003; Gamarnik et al., 2006).

Despite their impressive successes, message-passing algorithms are poorly understood theoretically. Though a body of work is emerging, existing results are somewhat disparate and often customized to particular contexts. It is also surprising that such methods are largely unknown to the operations research community. In this paper, to advance understanding of message passing and to foster a connection with operations research, we will develop and interpret such an approach in the context of a classical resource allocation problem. Here, we will demonstrate that message passing can be viewed as a generalization of a classical decentralized solution technique, that of price-based decomposition. This will allow us to both elucidate the behavior of message passing and to demonstrate how it can offer significant benefits.

To begin, consider a system consisting of a set of activities and a set of resources. Each activity contributes utility to an overall system objective, as a function of the resources allocated to it, and each resource is of limited supply. The system manager's decision problem

is to allocate resources between the activities, so as to maximize overall utility. This resulting optimization program, whose objective and constraints are additively separable, is one of the oldest and most well-studied problems in operations research, economics, and engineering.

We will consider *decentralized* decision making methods for resource allocation. Such methods decompose the problem across the collection of agents that participate in the system. The spirit here is to allow activity managers, each responsible for a particular activity, to make their own resource consumption decisions. These decisions cannot be made in isolation, however. Since resources may be profitably used by other activities, consumption decisions by a single activity manager have an impact across the entire system. Decentralized methods address these decision externalities via coordination signals, or incentives, that influence resource consumption decisions. These incentives serve to align the objective of each individual activity manager to that of the system. Note that, in this paper, we are not considering “incentives” in a game theoretic or strategic sense, but rather as a coordination mechanism. We are assuming that activity managers are myopic with respect to the incentives they are provided, and do not seek to manipulate these incentives through strategic behavior. This would be the case, for example, if the agents are cooperative and jointly wish to maximize the global utility. It would also be the case in a price-taking or competitive equilibrium setting, where, even if agents are strategic, no single agent has sufficient market power to manipulate incentives, and myopic behavior is optimal.

One benefit of decentralized methods is that they allow for greater flexibility in the management of complex systems. This is illustrated in the following example:

Example 1. (Organizational Management) *Consider a large and complex firm. Activities represent divisions of the firm, and resources represent inputs to the processes of the firm, such as capital or raw materials, that are of limited supply. The firm’s resource allocation problem is to optimize the distribution of the resources across the divisions. Each division may, in turn, be faced with its own complicated internal decision making process. Given an allocation of resources, the benefit generated by a division’s activity may entail optimization of a large number of decisions that govern how the activity is conducted. Any model of the division that is tractable from the perspective of a central planner will necessarily be simplified or abstract. As such, the resource allocation decisions made by a central planner can constrain activities in ways that prevent the beneficial reallocation of resources between activities.*

An alternative to the centralized micromanagement of resources is to have resource consumption decisions made by each individual division. The activity managers will have the greatest expertise in and knowledge of their particular activities. Further, over time, the activities may be changing, or the managers may be learning how to better conduct their activities. Hence, activity managers are in the best position to accurately model and understand their resource needs on an ongoing basis. By having individual divisions make their own resource consumption decisions, decentralized methods allow for greater management flexibility, and more robust and efficient decision making.

Decentralized methods provide further benefits by reducing communication costs and distributing information processing tasks. This allows for their use in many settings, such as the following, where centralized solutions have prohibitive communication and computational requirements:

Example 2. (Network Rate Control) *Consider a communications network consisting of a set of links (resources), and a set of users (activities). Each user wishes to transmit data across a particular path (subset of links) in the network, and generates utility as a function of the transmission rate allocated to it. Each link in the network is capable of transmitting data at some finite capacity. The network manager's problem is to allocate the capacity along each link among the users requiring service from the link, so as to maximize the overall utility.*

In such a network, the users and links are geographically distributed and physically disparate. A central planner would require a global view of the network. This would entail significant additional communication that may degrade the performance of the network. Further, a central planner would require computational resources commensurate with the size of the network. Decentralized methods, on the other hand, allow users and links to coordinate their respective consumption and allocation decisions by purely local communication that occurs alongside the regular flow of network traffic. Neither the agents nor the network manager require knowledge of the entire network. Further, since the computational burden is shifted to the agents that comprise the network, the network manager does not require additional computational resources.

In the case where the utility functions are concave (often called the convex resource allocation problem), the classical theory of convex optimization establishes shadow prices (Lagrange multipliers) as proxies for decentralization. Given a proper set of prices for resources, each activity manager can optimize resource consumption so as to maximize the

utility generated by the activity minus the cost (as reflected through prices) of the consumed resources, so that the resulting decision will be optimal for the system manager’s problem. Price-based methods for decentralized resource allocation have been developed as far back as the 1950’s, dating to the pioneering work of Arrow, Hurwicz, and others (e.g. Arrow and Hurwicz, 1977). Such methods have the following benefits:

1. *A tractable representation of externalities that leads to system-optimal behavior.*

Prices provide a linear representation of externalities, and concisely summarize the impact of decisions across the system. They enable each activity manager to align their objective with that of the system manager.

2. *Distributed asynchronous algorithms for computing prices and allocations.*

Optimal prices and allocations can be computed iteratively via gradient methods. These methods require only communication between activity managers, which make resource consumption decisions, and resource managers, which determine prices. Further, each activity manager needs only to communicate with the resource managers for resources it requires. Neither communication with nor even knowledge of other activities and resources is necessary, nor is any other global coordination or synchronization required.

In convex resource allocation problems, fixed prices can provide appropriate incentives to induce system-optimal decisions within activities. This is not generally true for non-convex problems, where there may be no set of prices which supports a globally optimal allocation. Non-convexities appear in many practical problem instances for a host of reasons. The underlying resources may be discrete and indivisible. The activities may have increasing returns to scale, or inelastic demand for resources. In such cases, price-based decentralized algorithms may converge to local optima, or may fail to converge at all.

In this paper, we consider prices that vary across activities and consumption levels. We refer to such nonlinear price functions as *messages*, as they can be viewed as incentives communicated between resource managers and activity managers. Message-based incentives allow for a richer description of externalities than prices, while still maintaining computational tractability. We argue that messages extend many of the benefits of prices to non-convex resource allocation problems. The contributions of this paper are as follows:

1. *We propose a new equilibrium concept for message-based incentives.*

We define a set of equilibrium message-based incentives as the fixed points of a message-

passing operator. We establish that, under broad technical conditions, these equilibria exist, and that they can support optimal allocations even when prices can not.

2. *We demonstrate that messages lead to system-optimal behavior for convex problems.*

We demonstrate that in the convex case, message-passing equilibria lead to system-optimal behavior. Indeed, in this case, messages are locally equivalent to prices: the marginal incentives provided by a set of equilibrium messages at the optimal allocation are precisely optimal shadow prices.

3. *We argue that messages yield allocations superior to prices for non-convex problems.*

For non-convex problems, in general, message-based incentives will not guarantee system-optimal allocations. This is not surprising, because this class of problems includes many which are provably intractable. Any method which guarantees global optimality is not likely to be of practical use in large scale problems. Allocations resulting from message-based incentives will, however, satisfy a property which precludes the improvement of the system objective under certain types of transfers of resources between activities. This property is stronger than the local optimality guarantees which can be made for price-based incentives. Further, we present a computational case study involving inelastic network rate control in which message-based incentives yield far superior solutions to alternative heuristics that utilize price-based incentives or greedy search.

4. *We propose a distributed asynchronous algorithm for computing messages and allocations.*

Equilibrium messages can be computed via a successive approximations procedure. We show how this procedure decomposes into purely local communication between activity and resource managers. In the inelastic rate control example, this takes a particularly simple form where the algorithm operates alongside the normal flow of network traffic, and appends a single real number to each data packet.

The balance of the paper is organized as follows: in Section 2, we describe the resource allocation problem. In Section 3, we describe the decision externalities that occur because of decentralization. In Section 4, we define the concept of a message-passing equilibrium, and compare the optimality properties of the message-based incentives with those of price-based incentives. In Section 5, we describe a distributed asynchronous algorithm for computing

message-passing equilibria. Finally, in Section 6, we discuss the application of message passing to a network resource allocation problem. Proofs are provided in the appendices.

2. Problem Formulation

Consider the following prototypical resource allocation problem: a set of resources \mathcal{R} , each of finite capacity, is to be allocated among a set of activities \mathcal{A} . Each activity $a \in \mathcal{A}$ depends on some subset $\mathcal{R}(a) \subseteq \mathcal{R}$ of the resources. For each a and each $r \in \mathcal{R}(a)$, denote by $x_{ar} \geq 0$ the decision variable representing the quantity of resource r to be allocated to activity a . Denote the allocation decisions by $x \triangleq \{x_{ar} : a \in \mathcal{A}, r \in \mathcal{R}(a)\}$. Denote by $x_{\mathcal{R}(a)} \triangleq \{x_{ar} : r \in \mathcal{R}(a)\}$ the consumption bundle for activity a . A utility function $u_a(\cdot)$ specifies the contribution $u_a(x_{\mathcal{R}(a)}) \in \mathbb{R}$ of activity a to the overall system objective, as a function of the allocation $x_{\mathcal{R}(a)}$ it receives. For each resource r , denote by $\mathcal{A}(r) \triangleq \{a \in \mathcal{A} : r \in \mathcal{R}(a)\} \subseteq \mathcal{A}$ the set of activities which depend on resource r . Denote by $x_{\mathcal{A}(r)} \triangleq \{x_{ar} : a \in \mathcal{A}(r)\}$ the allocations of resource r . There is a finite quantity $b_r > 0$ of each resource r available, hence we require that $x_{ar} \in \mathcal{X}_r \triangleq [0, b_r]$, for all $a \in \mathcal{A}(r)$, and that $\sum_{a \in \mathcal{A}(r)} x_{ar} \leq b_r$. The relationships between activities and resources can be conveniently encoded using a graphical representation:

Definition 1. (Dependency Graph) Define the dependency graph \mathcal{D} to be an undirected bipartite graph consisting of vertices corresponding to the activities \mathcal{A} and the resources \mathcal{R} . An edge (a, r) is present if and only if activity a depends on resource r , that is, if $a \in \mathcal{A}(r)$.

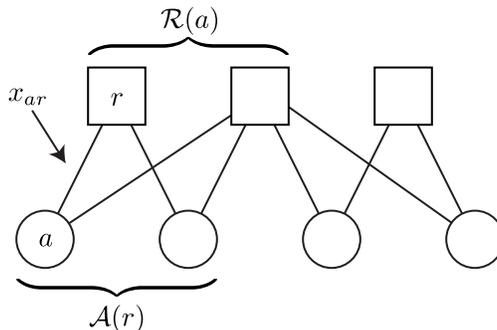


Figure 1: A dependency graph. Vertices in the graph correspond to activities and resources, edges in the graph correspond to decision variables.

An optimal allocation is determined by solving the following program:

$$(2.1) \quad \begin{array}{ll} \text{maximize} & U(x) \triangleq \sum_{a \in \mathcal{A}} u_a(x_{\mathcal{R}(a)}) \\ \text{subject to} & \sum_{a \in \mathcal{A}(r)} x_{ar} \leq b_r, \quad \forall r \in \mathcal{R}, \\ & x_{ar} \in \mathcal{X}_r, \quad \forall a \in \mathcal{A}, r \in \mathcal{R}(a). \end{array}$$

The function $U(\cdot)$ is called the *system objective function*, and the problem (2.1) is called the *system manager's problem*. Note that the system objective function is separable across activities but not across resources. If the utility functions are concave, this optimization problem can be addressed by methods of convex optimization, as we discuss in Section 3. Our primary motivation, however, is to consider cases where utility functions are not concave, as in the following example, which we revisit in Section 6.

Example 3. (Inelastic Rate Control) *Consider a communications network consisting of a set of links (resources), and a set of users (activities). Each user a wishes to transmit data across a particular path (subset of links) $\mathcal{R}(a)$ in the network. For each user a and each link $r \in \mathcal{R}(a)$, the decision variable x_{ra} represents the data transmission rate on the link r that is allocated to the user a . Each link in the network is capable of transmitting data at some finite capacity.*

The overall transmission rate for a user is constrained by the minimum transmission rate it is allocated along all the links in its path. Each user a desires some minimum overall transmission rate $w_a > 0$. If the user is able to transmit at that rate, the user derives utility $z_a > 0$. Otherwise, the user derives 0 utility. Hence, the utility function for user a takes the form

$$u_a(x_{\mathcal{R}(a)}) = \begin{cases} z_a & \text{if, for each } r \in \mathcal{R}(a), x_{ar} \geq w_a, \\ 0 & \text{otherwise,} \end{cases}$$

which is not concave. As we shall see in Section 6, in this case, the resource allocation problem is equivalent to a multi-dimensional knapsack problem.

3. Decentralization and Externalities

Under a decentralized decision making scheme, individual activity managers make their own resource consumption decisions. These individual decisions impact the entire system since, as a resource is consumed by one activity, the quantity of the resource available for other activities is reduced. A coordination mechanism is required to address these decision externalities.

One very general way that this can be accomplished is as follows: for each activity a , consider the optimization problem

$$(3.1) \quad \begin{aligned} & \text{maximize} && u_a(x_{\mathcal{R}(a)}) + E_a(x_{\mathcal{R}(a)}) \\ & \text{subject to} && x_{ar} \in \mathcal{X}_r, \quad \forall r \in \mathcal{R}(a). \end{aligned}$$

Here, the function $E_a(\cdot)$ is defined by

$$(3.2) \quad \begin{aligned} E_a(x_{\mathcal{R}(a)}) \triangleq & \text{maximize} && \sum_{a' \in \mathcal{A} \setminus a} u_{a'}(x_{\mathcal{R}(a)'}) \\ & \text{subject to} && \sum_{a' \in \mathcal{A}(r)} x_{a'r} \leq b_r, \quad \forall r \in \mathcal{R}, \\ & && x_{a'r} \in \mathcal{X}_r, \quad \forall a' \in \mathcal{A} \setminus a, r \in \mathcal{R}(a)'. \end{aligned}$$

In (3.2), the decision variables are the consumption decisions $\{x_{a'r} : a' \in \mathcal{A} \setminus a, r \in \mathcal{R}(a)'\}$ for all activities *except* a . Given a consumption decision $x_{\mathcal{R}(a)}$ for activity a , the quantity $E_a(x_{\mathcal{R}(a)})$ is the optimized value of utility across the rest of the system. Relative values of $E_a(\cdot)$ exactly capture the impact of consumption decisions for the activity a to the rest of the system. In other words, the function $E_a(\cdot)$ captures the externalities of decision-making for activity a . This function can be used as an incentive to the activity manager, aligning the objective (3.1) of the activity manager and the objective (2.1) of the system manager.

In general, however, such a mechanism is not practical. The function $E_a(\cdot)$ can be an arbitrary multidimensional nonlinear function. It is not clear how to tractably represent or compute such an object, much less in a decentralized manner. We discuss here two exceptions that provide tractable special cases. The first involves concave utility functions.

Example 4. (Concave Utility Functions) *It is well-known that if utility functions are strictly concave, then the optimal allocation is unique and supported by a set of prices. In particular, there exists an allocation x^* and a price vector $p^* \in \mathbb{R}_+^{\mathcal{R}}$, such that x^* is the unique optimal solution to the system manager's problem (2.1), and each $x_{\mathcal{R}(a)}^*$ is the unique maximizer of the optimization problem*

$$(3.3) \quad \begin{aligned} & \text{maximize} && u_a(x_{\mathcal{R}(a)}) - \sum_{r \in \mathcal{R}(a)} p_r^* x_{ar} \\ & \text{subject to} && x_{ar} \in \mathcal{X}_r, \quad \forall r \in \mathcal{R}(a). \end{aligned}$$

This program opens the door to decentralized management based on an incentive system. Instead of overseeing each activity's consumption, the manager of a resource can set a unit price and leave consumption decisions in the hands of activity managers. If the manager for activity a maximizes the utility his activity generates minus the cost of resources consumed, objectives are aligned and he chooses to consume exactly $x_{\mathcal{R}(a)}^$.*

One way to interpret a price-based incentive system is as a linear and separable approximation to the true externalities. If the utility functions are concave, the solution of (3.1) is

determined by first-order conditions. Hence, we need only to characterize the first-order behavior of $E_a(\cdot)$ around the optimal allocation $x_{\mathcal{R}(a)}^*$. This behavior is captured by the shadow price vector p^* , and the price-based incentives in the optimization program (3.3).

Unfortunately, the preceding story does not generally apply when utility functions are non-concave. Even if there is a unique optimal solution, there may be no price vector that leads activity managers to make optimal decisions. The solution concept presented in the next section generalizes price-based incentives in a way that addresses this.

Before moving on to our solution concept, let us discuss a second special case that allows for general utility functions but imposes a requirement on the structure of the dependency graph.

Example 5. (A Chain of Activities) Consider a case with resources $\mathcal{R} = \{r_1, \dots, r_{N+1}\}$ and activities $\mathcal{A} = \{a_1, \dots, a_N\}$, where each activity a_i can only consume the resources r_i and r_{i+1} . Here, the dependency graph forms a chain. The externalities imposed by the i th activity's consumption bundle $x_{\mathcal{R}(a_i)} = (x_{a_i, r_i}, x_{a_i, r_{i+1}})$ decompose according to $E_{a_i}(x_{\mathcal{R}(a_i)}) = V_{r_i \rightarrow a_i}(x_{a_i, r_i}) + V_{r_{i+1} \rightarrow a_i}(x_{a_i, r_{i+1}})$. Hence, the externalities can be represented as a sum of two one-dimensional functions. One of the two functions encodes the impact of activity a_i on activities a_1, \dots, a_{i-1} , while the other encodes impact on activities a_{i+1}, \dots, a_N . The chain structure allows for this decomposition since these two sets of activities are only coupled through decisions of activity a_i .

The functions $V_{r_i \rightarrow a_i}(\cdot)$ and $V_{r_{i+1} \rightarrow a_i}(\cdot)$ can be computed recursively via dynamic programming. Given these functions, optimal allocations for each activity a_i solve

$$(3.4) \quad \begin{aligned} & \text{maximize} && u_{a_i}(x_{\mathcal{R}(a_i)}) + V_{r_i \rightarrow a_i}(x_{a_i, r_i}) + V_{r_{i+1} \rightarrow a_i}(x_{a_i, r_{i+1}}) \\ & \text{subject to} && x_{a_i, r} \in \mathcal{X}_r, \quad \forall r \in \{r_i, r_{i+1}\}. \end{aligned}$$

So long as the solutions to such optimization problem are unique, activity managers can make optimal consumption decisions in a decentralized fashion.

For general dependency graphs, externalities do not decompose as they do in a chain. However, as we will see in the next section, our new solution concept approximates externalities using similarly separable decompositions.

4. Solution Concept

Our solution concept involves a general class of incentives, which we refer to as *messages*. These messages are exchanged between managers for each activity and each resource. For

each activity a , the activity manager receives a message from the resource manager for each resource $r \in \mathcal{R}(a)$. This message is a function $V_{r \rightarrow a}: \mathcal{X}_r \rightarrow \mathbb{R}$. The quantity $V_{r \rightarrow a}(x_{ar})$ can be thought of as a penalty imposed on activity a for consuming x_{ar} units from the finite supply of resource r that is available.

Similarly, for each resource r , the resource manager receives a message from each activity manager corresponding to an activity $a \in \mathcal{A}(r)$. This message is a function $V_{a \rightarrow r}: \mathcal{X}_r \rightarrow \mathbb{R}$. The quantity $V_{a \rightarrow r}(x_{ar})$ can be thought of as a benefit generated to the resource manager by allocating x_{ar} units from its finite supply to activity a .

The spirit here is to allow decisions to be made in a decentralized manner: for each activity a , the activity manager makes a consumption decision that optimizes

$$(4.1) \quad \begin{array}{ll} \text{maximize} & u_a(x_{\mathcal{R}(a)}) + \sum_{r \in \mathcal{R}(a)} V_{r \rightarrow a}(x_{ar}) \\ \text{subject to} & x_{ar} \in \mathcal{X}_r, \quad \forall r \in \mathcal{R}(a). \end{array}$$

Comparing with (3.1), the messages received by the manager of an activity a can be viewed as an additively separable approximation to the true externalities,

$$(4.2) \quad E_a(x_{\mathcal{R}(a)}) \approx \sum_{r \in \mathcal{R}(a)} V_{r \rightarrow a}(x_{ar}).$$

This approximation is motivated by the case where the dependency graph \mathcal{D} is a tree, that is, a graph with no cycles. In this case, the impact on the rest of the system that occurs when the activity consumes a particular quantity of a resource does not depend on the quantities of other resources consumed by the activity. Hence, the approximation (4.2) is exact. This is illustrated in Figure 2. There, the optimization problem (3.2) for the externalities of activity a decomposes into three independent subproblems, so that $E_a(x_{ar_1}, x_{ar_2}, x_{ar_3}) = V_{r_1 \rightarrow a}(x_{ar_1}) + V_{r_2 \rightarrow a}(x_{ar_2}) + V_{r_3 \rightarrow a}(x_{ar_3})$.

Comparing the incentives provided by the messages in (4.1) to those provided by the price-based incentives in (3.3), it is clear that messages generalize prices by allowing for nonlinear incentives. Further, with prices, there is a single price associated with each resource. Hence, the incentives corresponding to a single resource are identical to all the activities that require the resource. Messages provide additional flexibility by allowing these incentives to vary depending on the identity of the activity.

A related body of work in the economics literature also treats nonconvex resource allocation problems using as proxies for decentralization nonlinear incentives that can vary across activities (e.g., Spence, 1977; Berliant and Dunz, 1990; Aliprantis, 2001; Mordukhovich,

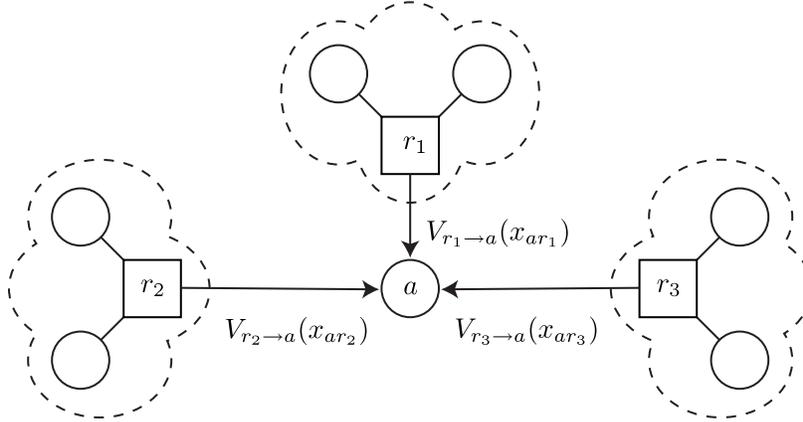


Figure 2: A dependency graph that is a tree. The externalities of consumption decisions for activity a decompose into three independent sub-problems.

2005). Similarly with our message-passing paradigm, this work characterizes nonlinear incentives that induce consumption of resources in ways that satisfy various optimality criteria. On the other hand, when there are multiple resources and activities, it is not clear how to address the associated solution concepts without computing global optima of complex non-convex functions. As we will see, our work on message passing differs in that the solution concept is motivated by the existence of a tractable heuristic that efficiently approximates solutions through a simple distributed protocol.

It is also worth mentioning a potential relation to augmented Lagrange multiplier functions (e.g., Rockafeller, 1974; Bertsekas, 1982). Here, the consumption of a resource is penalized by a function of the consumption level, which is a nonlinear function parameterized by a small number of multipliers. One important difference from our message-passing paradigm is that this function is not applied to the consumption of each agent but rather the total consumption of a resource by all agents. Yet the substantial and sophisticated literature on augmented Lagrange multiplier functions and algorithms motivates exploring whether some of this technology can help in the design and analysis of message-passing algorithms.

4.1 Message-Passing Equilibrium

Our solution concept requires that messages obey a notion of equilibrium. We explain this intuitively now and subsequently provide a precise definition. Think of $V_{r \rightarrow a}(x_{ar})$ as a penalty imposed on activity a for consuming x_{ar} units of resource r . The reason for penalizing the activity is that the resource can be profitably used by others. Interpret $V_{a' \rightarrow r}(x_{a'r})$ as the

benefit generated by allocating $x_{a'r}$ units of the resource r to an alternative activity a' . One part of our equilibrium condition states that the penalty $V_{r \rightarrow a}(x_{ar})$ should be commensurate with the sum of benefits $V_{a' \rightarrow r}(x_{a'r})$ among the alternative activities $a' \in \mathcal{A}(r) \setminus a$, assuming the remaining $b_r - x_{ar}$ units of the resource are allocated optimally among them. This is illustrated in Figure 3(a).

Note that, in addition to benefiting activity a , the choice of x_{ar} affects the activity's other consumption decisions $x_{ar'}$, for $r' \in \mathcal{R}(a) \setminus r$. The benefit $V_{a \rightarrow r}(x_{ar})$ should be commensurate with sum of the utility $u_a(x_{\mathcal{R}(a)})$ generated by activity a and the penalties $V_{r' \rightarrow a}(x_{ar'})$ for the activity's consumption of other resources, assuming that the other resource consumption decisions are made optimally. A second equilibrium condition appropriately accounts for this cascading influence of the choice of x_{ar} . This is illustrated in Figure 3(b).

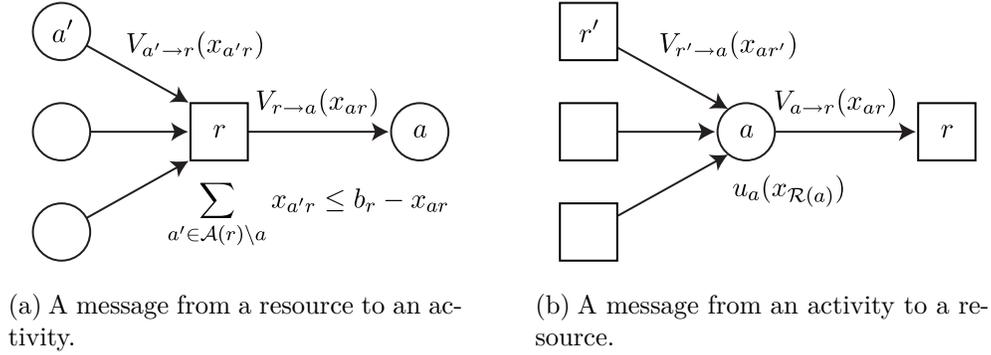


Figure 3: The equilibrium condition for messages.

To define our equilibrium conditions more precisely, we introduce an operator. Denote by V an entire set of messages, including the messages from activity managers to resource managers, $\{V_{a \rightarrow r}(\cdot) : \forall a \in \mathcal{A}, r \in \mathcal{R}(a)\}$, and messages from resource managers to activity managers, $\{V_{r \rightarrow a}(\cdot) : \forall r \in \mathcal{R}, a \in \mathcal{A}(r)\}$. The operator F maps one set of messages to another and is defined by

$$(4.3a) \quad (FV)_{a \rightarrow r}(x_{ar}) \triangleq \begin{aligned} & \text{maximize} && u_a(x_{\mathcal{R}(a)}) + \sum_{r' \in \mathcal{R}(a) \setminus r} V_{r' \rightarrow a}(x_{ar'}) \\ & \text{subject to} && x_{ar'} \in \mathcal{X}_{r'}, \quad \forall r' \in \mathcal{R}(a) \setminus r, \end{aligned}$$

$$(4.3b) \quad (FV)_{r \rightarrow a}(x_{ar}) \triangleq \begin{aligned} & \text{maximize} && \sum_{a' \in \mathcal{A}(r) \setminus a} V_{a' \rightarrow r}(x_{a'r}) \\ & \text{subject to} && \sum_{a' \in \mathcal{A}(r) \setminus a} x_{a'r} \leq b_r - x_{ar}, \\ & && x_{a'r} \in \mathcal{X}_r, \quad \forall a' \in \mathcal{A}(r) \setminus a. \end{aligned}$$

The first part of the definition (4.3a) relates the benefit of allocating resource r to activity a to the penalties associated with other resource constraints associated with the activity. The

second part of the definition (4.3b) relates the penalty imposed on activity a for consuming resource r to benefits that other activities could obtain.

In order to elucidate the structure of the operator F , consider the case where the dependency graph \mathcal{D} is a tree, and a set of messages V satisfies the fixed point equation $V = FV$. In this case, the messages V correspond to a dynamic programming decomposition of the decision externalities for all the activities, and the operator F is analogous to a Bellman operator.

In the case where the dependency graph has cycles, the operator F may not have any fixed points. This can be addressed with a minor modification: note that adding or subtracting a constant from any message does not influence incentives. Only the relative values of a message matter. As such, we restrict attention to messages that assign zero value to a null allocation. In other words, for each activity a and $r \in \mathcal{R}(a)$, we consider only messages for which $V_{a \rightarrow r}(0) = 0$ and $V_{r \rightarrow a}(0) = 0$. We introduce a modified version H of the operator F which subtracts an offset to accomplish this:

$$(HV)_{a \rightarrow r}(x_{ar}) \triangleq (FV)_{a \rightarrow r}(x_{ar}) - (FV)_{a \rightarrow r}(0), \quad (HV)_{r \rightarrow a}(x_{ar}) \triangleq (FV)_{r \rightarrow a}(x_{ar}) - (FV)_{r \rightarrow a}(0).$$

The subtraction of an offset is analogous to the modification of the Bellman operator in average cost dynamic programming necessary when moving from a finite horizon to an infinite horizon setting.

We call a set of messages V a *message-passing equilibrium* if $V = HV$. The following result, whose proof can be found in the online supplement, offers a general sufficient condition for existence.

Theorem 1. *Assume that the utility functions are Lipschitz continuous. Then, a message-passing equilibrium exists.*

This sufficient condition is broad and covers most models of practical interest. This is in contrast to conditions for existence of prices that support optimal allocations, which fail to hold in many contexts involving non-concave utility functions. The following example illustrates a simple situation where a message-passing equilibrium supports an optimal allocation but prices do not.

Example 6. (Equilibrium Messages When Prices Fail) *Consider a system with two activities $\mathcal{A} = \{a_1, a_2\}$ and a single resource $\mathcal{R} = \{r\}$. There is a unit quantity of the*

resource available; i.e., $b_r = 1$. The utility functions are

$$u_{a_1}(x_{a_1,r}) = \begin{cases} \frac{3}{2}x_{a_1,r} & \text{if } x_{a_1,r} \in [0, 1/4], \\ \frac{3}{8} & \text{if } x_{a_1,r} \in [1/4, 1], \end{cases} \quad \text{and} \quad u_{a_2}(x_{a_2,r}) = \begin{cases} 0 & \text{if } x_{a_2,r} \in [0, 1/2], \\ 2x_{a_2,r} - 1 & \text{if } x_{a_2,r} \in [1/2, 1]. \end{cases}$$

It is easy to see that in the unique optimal allocation, activity a_2 consumes the entire pool of the resource. However, no price supports this solution. In order to encourage activity a_2 to consume, the price must be no greater than 1. On the other hand, if the price is $3/2$ or less, activity a_1 will want to consume a quarter of the resource pool.

Equilibrium messages, however, do support the optimal solution. Equilibrium messages from the resource are given by

$$V_{r \rightarrow a_1}(x_{a_1,r}) = \begin{cases} -2x_{a_1,r} & \text{if } x_{a_1,r} \in [0, 1/2], \\ -1 & \text{if } x_{a_1,r} \in [1/2, 1], \end{cases} \quad \text{and} \quad V_{r \rightarrow a_2}(x_{a_2,r}) = \begin{cases} 0 & \text{if } x_{a_2,r} \in [0, 3/4], \\ \frac{9}{8} - \frac{3}{2}x_{a_2,r} & \text{if } x_{a_2,r} \in [3/4, 1]. \end{cases}$$

Given these messages, each activity manager's decentralized decision leads to the system optimum.

This example illustrates how message-passing equilibria can be more effective than prices as proxies for decentralization. It is a trivial example because there is only a single resource and the dependency graph is a chain. However, it is not difficult to generalize the idea to arrive at the same conclusion for examples involving greater numbers of activities and resources and more complex dependency graphs.

4.2 Optimality

Given a message-passing equilibrium V , an allocation can be selected by optimizing, for each activity a , the activity manager's problem (4.1). In this section, we characterize the optimality properties of this allocation.

Consider two feasible allocations x and x' . We can interpret the difference $x - x'$ as a set of *direct transfers* of resources between various activity managers and resource managers. These transfers involve pairs of activities and resources that are indexed by the set $\Delta(x, x') \triangleq \{(a, r) \in \mathcal{A} \times \mathcal{R} : x_{ar} \neq x'_{ar}\}$, which is the collection of decision variables that differ between the two allocations.

Given an allocation x , we say that a set of direct transfers is feasible if the allocation resulting from the combination of transfers is feasible. A *cycle* is a set of transfers for which the activity-resource pairs involved can be written as

$$(a_1, r_1), (a_2, r_1), (a_2, r_2), (a_3, r_2), \dots, (a_k, r_k), (a_1, r_k),$$

where each resource index and each activity index are distinct. The following theorem characterizes a set of transfers that cannot improve a solution delivered by a message-passing equilibrium:

Theorem 2. *Given a message-passing equilibrium V , assume that each activity manager’s problem (4.1) has a unique solution, and define x^* to be the resulting allocation. The objective value of this allocation cannot be increased by any set of transfers that involves at most one cycle.*

This theorem is a corollary of Theorem 6, which is stated and proved in the online supplement. Key elements of the argument are borrowed from the analyses of Freeman and Weiss (2001); Wainwright et al. (2004), but are translated to our resource allocation context.

Theorem 2 guarantees, for example, that the objective cannot be improved by transfers involving redistribution of only a single resource, as such transfers contain no cycles. If the original dependency graph \mathcal{D} contains at most one cycle, then *any* set of transfers contains at most one cycle. Hence, x^* is a global optimum. We comment further on this optimality property in Section 4.4.

Note that Theorem 2 requires that each activity manager’s problem (4.1) have a *unique* solution. Non-uniqueness of these solutions indicates that there are activity managers are indifferent between multiple consumption bundles. In such cases, it may be necessary to break ties in a consistent way across activity managers. This is typically done in a problem-specific way, as in the example in Section 6.3.

4.3 Concave Utility Functions

In this section, we analyze message-passing equilibria in a convex resource allocation setting: we assume that the utility functions are Lipschitz continuous and strictly concave. Under this assumption, the system manager’s problem (2.1) has a unique globally optimal allocation. Further, by the classical theory of Lagrange multipliers, a supporting price vector exists. We demonstrate that the message-passing approach yields equivalent results.

To begin, note that, without loss of generality, we can restrict ourselves to message sets with concave messages, by the following analog of Theorem 1, whose proof can be found in the online supplement.

Theorem 3. *There exists a message-passing equilibrium with concave and Lipschitz continuous messages.*

Given a message-passing equilibrium with concave messages, each activity manager's problem (4.1) has a strictly concave objective and a convex constraint set, and, hence, a unique optimal solution. Further, in this case, Theorem 2 can be strengthened to the following global optimality guarantee, whose proof can be found in the online supplement.

Theorem 4. *Consider a message-passing equilibrium with concave and Lipschitz continuous messages. The resulting allocation of resources is globally optimal for the system manager's problem (2.1).*

Now, let x^* be the globally optimal allocation, and assume that the system manager's objective $U(\cdot)$ is differentiable at x^* . Let $p^* \in \mathbb{R}_+^{\mathcal{R}}$ be the unique supporting price vector. For an activity a and resource $r \in \mathcal{R}(a)$, we can think of p_r^* as a marginal incentive for the manager of activity a , in the sense that $\frac{\partial}{\partial x_{ar}} u_a(x_{\mathcal{R}(a)}^*) = p_r^*$. We interpret this statement as saying that the marginal change in utility of deviating from the allocation x_{ar}^* is balanced by the incremental resource cost due to the price. The following theorem, whose proof can be found in the online supplement, shows that the derivatives of messages in message-passing equilibrium can be interpreted the same way.

Theorem 5. *Let x^* be the globally optimal allocation for the system manager's problem (2.1) and let p^* be a supporting price vector. Suppose that $U(\cdot)$ is differentiable at x^* . Consider a message-passing equilibrium V with concave and Lipschitz continuous messages. Then, for each activity a and resource r ,*

$$\frac{d}{dx_{ar}} V_{a \rightarrow r}(x_{ar}^*) = p_r^*, \quad \frac{d}{dx_{ar}} V_{r \rightarrow a}(x_{ar}^*) = -p_r^*,$$

where the existence of the above derivatives is guaranteed. Thus,

$$\frac{\partial}{\partial x_{ar}} u_a(x_{\mathcal{R}(a)}^*) = \frac{d}{dx_{ar}} V_{a \rightarrow r}(x_{ar}^*) = -\frac{d}{dx_{ar}} V_{r \rightarrow a}(x_{ar}^*) = p_r^*.$$

Theorem 5 implies that, subject to differentiability considerations, concave message-passing equilibria are unique in their first-order behavior at the optimal allocation, and this behavior corresponds to that of the unique shadow price vector.

4.4 Messages Versus Prices

As we have discussed, shadow prices and message-passing equilibria provide two different ways to decompose the system manager's problem (2.1) into a series of smaller problems,

one for each activity manager, which are of the form (3.3) and (4.1), respectively. These activity managers' problems are no longer coupled by resource constraints and can be solved independently. Both methods can be interpreted as providing incentives to each activity manager which capture decision externalities.

In the convex resource allocation case, the discussion in Section 4.3 suggests that these methods are equivalent. Both methods derive incentives that support the globally optimal allocation, and these incentives have equivalent structure in a local neighborhood of the globally optimal allocation.

For non-convex problems, however, prices may not support optimal allocations while message-passing equilibria still exist and the allocations they suggest satisfy certain optimality properties. In particular, allocations derived from message-passing equilibria satisfy the optimality property of Theorem 2.

Finally, it should be noted that the optimality properties of message-passing equilibria are not well understood from a theoretical perspective. Their performance on many difficult optimization problems is far better than suggested by the guarantee provided by Theorem 2. We shall see an example of this in Section 6.1.

5. Message-Passing Algorithms

Up to this point, we have described message-passing equilibrium as a solution concept and analyzed its properties. In this section, we consider the issue of computing message-passing equilibrium.

Since a message-passing equilibrium is a fixed point of the operator H , a natural approach to consider is the method of successive approximations. This is an iterative scheme which starts with some initial message set V , for example $V = 0$, and generates subsequent approximations to a message-passing equilibrium according to

$$(5.1) \quad V := (1 - \gamma)V + \gamma HV.$$

Here, the scalar $\gamma \in (0, 1]$ is a dampening factor. This procedure is repeated until it converges and a fixed point is reached. We generically call a successive approximation scheme of the form (5.1) a message-passing algorithm. As we will discuss in Section 5.3, this algorithm is not guaranteed to converge, though it is often an effective heuristic.

5.1 Tractability

Each iteration of the successive approximations method requires the solution of optimization problems of the following form

$$\begin{aligned}
 (5.2a) \quad & \text{maximize} && u_a(x_{\mathcal{R}(a)}) + \sum_{r' \in \mathcal{R}(a) \setminus r} V_{r' \rightarrow a}(x_{ar'}), \\
 & \text{subject to} && x_{ar'} \in \mathcal{X}_{r'}, \quad \forall r' \in \mathcal{R}(a) \setminus r, \\
 (5.2b) \quad & \text{maximize} && \sum_{a' \in \mathcal{A}(r) \setminus a} V_{a' \rightarrow r}(x_{a'r}), \\
 & \text{subject to} && \sum_{a' \in \mathcal{A}(r) \setminus a} x_{a'r} \leq b_r - x_{ar}, \\
 & && x_{a'r} \in \mathcal{X}_r, \quad \forall a' \in \mathcal{A}(r) \setminus a.
 \end{aligned}$$

for each activity a , resource r , and $x_{ar} \in \mathcal{X}_r$. Implicit in the application of this method is the assumption that these optimization problems can be solved efficiently. One case where in which this is true is if the dependency graph is *sparse*, that is, each resource is required by few activities, and each activity depends on few resources. In this case, both the problems (5.2a) and (5.2b) will be small and tractable.

More generally, there is special structure in the problems (5.2a) and (5.2b) that can be exploited for efficient solution by dynamic programming. First, note that the problem (5.2b) involves optimizing a separable objective function with an interval constraint on each variable and a single linear constraint on the sum of variable values. Such a problem can be efficiently solved as a series of one-dimensional optimization problems via dynamic programming.

Second, in many relevant cases, problem the (5.2a) can similarly be decomposed into a series of one-dimensional optimization problems. This is true, for example, if the utility function is additively separable or if utility depends only on the sum or minimum of allocated resources. We shall see one such example of this in Section 6.1. Further, note that if (5.2a) it not tractable, then each agent, individually and without regard to the rest of the system, is unable to maximize his utility. In such cases, it seems truly hopeless solve the much harder (global) resource allocation problem, which involves simultaneous optimization all of the agents' utilities coupled by the resource constraints, by *any* algorithm.

5.2 Distributed and Asynchronous Implementation

One important characteristic of the message-passing iteration (5.1) is that it naturally lends itself to a distributed and asynchronous implementation. Imagine an implementation where the activity and resource managers operate completely independently. Consider this from the perspective of an activity manager. At each point in time, the activity manager keeps track of the most recent message it has received from each neighboring resource manager in

the dependency graph. Occasionally, the activity manager can decide to send a new message to each neighboring resource manager, based on the most recent messages it has received from other resource managers. Resource managers behave in an analogous fashion. So long as each pair of activity and resource managers relay messages to one another an unbounded number of times, a fixed point of this distributed and asynchronous procedure is a message-passing equilibrium. Moreover, each manager only requires knowledge of and communication with neighboring managers in the dependency graph.

In general, messages are functions over a continuous domain. As such, the algorithm, as we have formulated it, cannot be implemented on digital computers. In some cases, such as the example we consider in Section 6.1, the messages lie in a finite dimensional space that is closed under the message-passing operator H . In such cases, messages can be transmitted by sending a vector of finitely many real numbers. Indeed, in the example in Section 6.1, each message can be encoded as a single real number.

In the more general case, it is necessary to approximate messages using representations that are parameterized by finitely many real numbers. Approximate message-passing algorithms are an interesting area for future research, and are more extensively considered by Moallemi and Van Roy (2007). We briefly discuss some possibilities here:

- Since each message is a function over an interval of real numbers, one simple discretization is to evaluate each message at a finite number of points in its domain. Values of the message between each pair of consecutive points can be approximated by linear interpolation. Such a scheme would thus approximate messages by piecewise linear functions.
- Another option for each activity manager to maintain an estimate of its optimal consumption decisions, by solving the problem (4.1) based on the most recent message received from each resource manager. Messages can then be approximated by quadratic functions based on a second-order Taylor expansion around the current estimate. The resulting algorithm can be viewed as a hybrid between message-passing algorithms and Newton's method.
- The problem of approximating messages is analogous to the situation in approximate dynamic programming (ADP), where low-dimensional approximations are made to high-dimensional value functions. Similar ideas can be applied, and one might consider

approximating messages as a linear combination of a finite collection of basis functions. An approximate message-passing equilibrium could be defined using a variation of the H operator that is projected onto the subspace spanned by the basis functions, and variations of ADP algorithms could be used to compute such equilibria.

5.3 Convergence

An immediate question is whether our message-passing algorithm converges to a message-passing equilibrium. In the context of Theorems 1 and 3, the operator H is continuous and compact. Hence, any sequence of iterates generated by successive approximation has limit points. However, these limit points may not, in general, be fixed points and thus equilibria—they may be contained in some invariant collection of message sets and may be, for example, periodically oscillating under the action of the operator H .

The question of convergence of the message-passing algorithm we have proposed for resource allocation remains open. If the dependency graph contains no cycles, message-passing can be seen to converge in a finite number of iterations by simple dynamic programming arguments. There is a body of work on convergence properties of various message-passing algorithms that are similar to ours, but designed for different problem contexts. Abstract conditions for convergence of a version of message passing across a range of problems have been developed (Tatikonda and Jordan, 2002), but these are difficult to verify in specific problem instances and do not apply in our context. Convergence has also been established for certain message-passing algorithms for special classes of optimization problems, such as maximum-weight matching (Bayati et al., 2005), and for certain random ensembles of optimization problems (Gamarnik et al., 2005).

One case that is well-understood, however, is for a message-passing algorithm applied to the optimization of unconstrained quadratic programs. Here, we and other authors (Weiss and Freeman, 2001; Rusmevichientong and Van Roy, 2001; Moallemi and Van Roy, 2009; Malioutov et al., 2006) have established convergence so long as the objective decomposes a particular way. Moreover, this convergence continues to hold in a distributed and asynchronous setting. In some cases, a rate of convergence analysis is also available (Moallemi and Van Roy, 2006).

We have recently extended these convergence results to a message-passing algorithm for optimization of unconstrained convex programs (Moallemi and Van Roy, 2007). Unfortunately, this analysis does not apply to the resource allocation context considered here.

However, in the following section, we see that our message-passing algorithm can still offer excellent solutions in the absence of convergence guarantees.

6. Network Rate Control

One feature of the message-passing algorithms described in the previous section is that they can be implemented in a distributed manner. This can be crucial in systems where information or computational resources are decentralized. In this section, we discuss an example involving transmission rate control in a communication network.

We consider a model put forth by Kelly (Kelly, 1997). There is a set \mathcal{R} of resources. Each resource represents a single link or bottleneck in a communication network. Each link r has a finite capacity $b_r > 0$. There is a set \mathcal{A} of activities, each representing a user who wishes to transmit data across the network. Each user a transmits data along a fixed route, which consisting of the set of links $\mathcal{R}(a) \subseteq \mathcal{R}$. This is illustrated in Figure 4. If the user is allocated capacity $x_{\mathcal{R}(a)}$ along these links, it can transmit at the rate $\min_{r \in \mathcal{R}(a)} x_{ar}$, and its utility is a function of this rate, $u_a(x_{\mathcal{R}(a)}) = \tilde{u}_a(\min_{r \in \mathcal{R}(a)} x_{ar})$. Here, we assume that the single-variable utility function $\tilde{u}_a: \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is non-decreasing. The objective is to allocate capacity in a way that maximizes the sum of utilities.

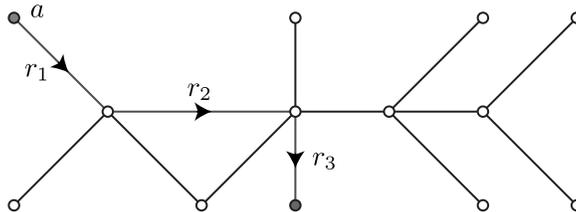


Figure 4: A network rate control example. Each edge in the graph is a constrained communications link. Each user is associated with a route in the network. For example, user a wishes to transmit data along the path consisting of the links $\mathcal{R}(a) = \{r_1, r_2, r_3\}$.

The numbers of users and links in modern communication networks are enormous. As such, it is not possible for a central authority to gather all the utility functions and link capacities as would be required to make centralized allocation decisions. Rather, the capacity of each link must be allocated based on locally available information. This information should be gathered from packets of data transmitted by users as they pass through the link. Further, links might mark the packets as they pass through to inform users of how much capacity they are allocated.

For the case of increasing strictly concave utility functions, referred to in the networking literature as the case of elastic traffic (Shenker, 1995), Kelly proposes an elegant distributed algorithm (Kelly, 1997). Our interest here is in designing a distributed message-passing scheme that effectively optimizes the allocation when utility functions are not concave, also known as the case of inelastic traffic. Such utility functions are required to model user preferences, for example, in real-time video and audio applications (Shenker, 1995). Optimization algorithms designed for elastic traffic, like that of Kelly, can lead to instabilities when applied in the presence of inelastic traffic (Lee et al., 2005). Several heuristics have been proposed to address inelastic traffic (Lee et al., 2005; Hande et al., 2005; Fazel and Chiang, 2005).

6.1 Inelastic Rate Control

Consider the extreme case of inelastic traffic described in Example 3. Here, each user a has a utility function $\tilde{u}_a(x_a) = z_a \mathbb{I}_{\{x_a \geq w_a\}}$. The quantity $w_a > 0$ is the minimum overall transmission rate desired by the user, and the $z_a > 0$ is the utility derived if allocated a rate w_a or larger.

In this setting, each user a is indifferent between transmitting at rate 0 or at any rate in the interval $(0, w_a)$, and is similarly indifferent between transmitting at rate w_a and at any rate larger than w_a . Hence, the system manager’s problem (2.1) is equivalent to the 0–1 integer program

$$(6.1) \quad \begin{array}{ll} \text{maximize} & \sum_{a \in \mathcal{A}} z_a y_a \\ \text{subject to} & \sum_{a \in \mathcal{A}(r)} w_a y_a \leq b_r, \quad \forall r \in \mathcal{R} \\ & y_a \in \{0, 1\}, \quad \forall a \in \mathcal{A}. \end{array}$$

Here, for each user a , the binary decision variable y_a determines the overall transmission rate allocation to user a : if $y_a = 1$, the user is allocated the desired transmission rate w_a . Otherwise, the user is allocated zero transmission rate. The program (6.1) is a multidimensional 0–1 knapsack problem, which is NP-hard (Fréville, 2004).

There are a number of heuristics available for solving the program (6.1) (see Kellerer et al. 2004, for example, for a survey). We consider the class of “primal greedy” heuristics. Such algorithms start with all users receiving a zero rate allocation. The users are then considered sequentially according to some ordering. When a user a is considered, the user receives the desired transmission rate w_a if such an allocation would preserve feasibility, given the allocations already made to previously considered users. Otherwise, the user is allocated zero transmission rate.

Critical to the success of such a greedy method is the ordering in which the users are considered. Typically, a measure of efficiency, or “bang-per-buck”, is defined for each user. This metric represents some estimate of the contribution of the user to the overall utility relative to the cost of the resource consumption of the user. We consider a prototypical efficiency metric, $e_a = z_a / \left(\sum_{r \in \mathcal{R}(a)} w_a / b_r \right)$, for each user a . The users are then considered in order of decreasing efficiency, and are greedily allocated their desired capacity so long as feasibility is maintained. We call this method the *greedy heuristic*.

Alternatively, one may consider the linear programming relaxation of (6.1),

$$(6.2) \quad \begin{aligned} & \text{maximize} && \sum_{a \in \mathcal{A}} z_a y_a \\ & \text{subject to} && \sum_{a \in \mathcal{A}(r)} w_a y_a \leq b_r, \quad \forall r \in \mathcal{R} \\ & && 0 \leq y_a \leq 1, \quad \forall a \in \mathcal{A}. \end{aligned}$$

This is equivalent to approximating the utility function of each user a by the concave piecewise linear function $\tilde{u}_a(x_a) \approx z_a \min(1, x_a/w_a)$. Note that this relaxation is very weak, and naive application of such an approximation leads to poor solutions. There may be many users who are allocated non-zero transmission rates that are less than their minimum desired transmission rates. These users consume capacity on the network, yet generate zero utility. Equivalently, these users correspond to decision variables with fractional values in the relaxation (6.2). Much better solutions can be generated from this approximation by examining the resulting vector p of shadow prices for the link constraints in (6.2) (Kellerer et al., 2004). These prices can be used as proxies for the cost of capacity on a link. Then, an efficiency metric can be defined according to $e_a = z_a / \left(\sum_{r \in \mathcal{R}(a)} w_a p_r \right)$, for each user a . An allocation decision can then be made as in the case of the greedy heuristic, by sequentially considering users in order of decreasing efficiency. We call this method *concave approximation*.

In Section 6.3, we compare the performance of message passing to the greedy heuristic and concave approximation methods described above. One motivation for choosing these particular methods is that they naturally lend themselves to distributed implementation. While our description of them has been sequential in nature, one can easily imagine decentralized implementations. A second motivation is that both the greedy heuristic and the concave approximation methods will yield locally optimal allocations. That is, the objective value cannot be improved through small deviations from the prescribed allocation. We can compare the quality of these allocations to those resulting from message passing. Finally, our consideration of the concave approximation method will highlight the fact that inelastic rate control provides a class of fundamentally non-convex resource allocation problems, that is,

problems that cannot be reasonably approximated using concave utility functions. We will see that the message-passing approach is able to cope with this fundamental non-convexity.

6.2 Distributed Message-Passing

Consider a distributed message-passing algorithm for a network with inelastic traffic. Since the messages $V_{r \rightarrow a}(x_{ar})$ and $V_{a \rightarrow r}(x_{ar})$ represent incentives, only their values at $x_{ar} \in \{0, w_a\}$ matter. Hence, we can parameterize these messages by $V_{a \rightarrow r}(x_{ar}) \triangleq v_{a \rightarrow r} \mathbb{I}_{\{x_{ar} \geq w_a\}}$ and $V_{r \rightarrow a}(x_{ar}) \triangleq v_{r \rightarrow a} \mathbb{I}_{\{x_{ar} \geq w_a\}}$, given parameters $v_{a \rightarrow r} \geq 0$ and $v_{r \rightarrow a} \leq 0$. Denote by v the set of all parameters $\{v_{a \rightarrow r}, v_{r \rightarrow a}\}$. This parametrization is closed under the operator H , and H can be expressed directly in terms of the parameter set v by

$$(6.3a) \quad (Hv)_{a \rightarrow r} = z_a + \sum_{r' \in \mathcal{R}(a) \setminus r} v_{r' \rightarrow a},$$

$$(6.3b) \quad (Hv)_{r \rightarrow a} = \begin{array}{ll} \text{maximize} & \sum_{a' \in \mathcal{A}(r) \setminus a} v_{a' \rightarrow r} y_{a'r} \\ \text{subject to} & \sum_{a' \in \mathcal{A}(r) \setminus a} w_{a'} y_{a'r} \leq b_r - w_a, \\ & y_{a'r} \in \{0, 1\}, \end{array} \quad \forall a' \in \mathcal{A}(r) \setminus a.$$

Given a set of parameters v , each activity a needs to solve the activity manager's problem (4.1). This is equivalent to selecting to consume quantities x_{ar} , for all $r \in \mathcal{R}(a)$, by $x_{ar} = w_a \mathbb{I}_{\{z_a + \sum_{r \in \mathcal{R}(a)} v_{r \rightarrow a} > 0\}}$.

Since the application setting here is naturally decentralized, it is important to be able to compute the message-passing update equations (6.3a)–(6.3b) and the resulting allocation in a distributed and possibly asynchronous fashion. We describe one particularly parsimonious implementation now. Consider a setting where, at each time t , each link r maintains a set of incoming and outgoing message parameters $\{v_{r \rightarrow a}^{(t)}, v_{a \rightarrow r}^{(t)}\}$ for each user $a \in \mathcal{A}(r)$. Assume that a user a transmits a data packet at time t , along the route $\mathcal{R}(a)$. A single real number m_a^+ is appended to this data packet, and the user initially sets $m_a^+ \triangleq z_a$. When the packet passes through a link $r \in \mathcal{R}(a)$, the value of m_a^+ is observed. This value is then updated by setting $m_a^+ \triangleq m_a^+ + v_{r \rightarrow a}^{(t)}$, before it is forwarded to the next link. When the packet arrives at the destination, an acknowledgment message is sent back the source, containing a single real number m_a^- . This number is initialized to $m_a^- \triangleq 0$. As it passes through a link $r \in \mathcal{R}(a)$, it is observed, and then updated according to $m_a^- \triangleq m_a^- + v_{r \rightarrow a}^{(t)}$, until it reaches the source. Now, at any link $r \in \mathcal{R}(a)$ along the route, the observed values m_a^+ and m_a^- can be combined to compute $m_a^+ + m_a^- = z_a + \sum_{r' \in \mathcal{R}(a) \setminus r} v_{r' \rightarrow a}^{(t)} = (Hv^{(t)})_{a \rightarrow r}$. Thus, the link can update its stored incoming message from user a by setting $v_{r \rightarrow a}^{(t+1)} \triangleq m_a^+ + m_a^-$. New

outgoing messages $v_{r \rightarrow a'}^{(t+1)}$, for each activity $a' \in \mathcal{A}(r) \setminus a$, can then be computed according to the update equation (6.3b). Similarly, when the user a receives the acknowledgment packet, it can compute the value $z_a + m_a^- = z_a + \sum_{r \in \mathcal{R}(a)} v_{r \rightarrow a}^{(t)}$. Then, it can make a consumption decision via by examining if $z_a + \sum_{r \in \mathcal{R}(a)} v_{r \rightarrow a}^{(t)} > 0$.

The spirit of this implementation is that the computation of a message-passing equilibrium and the associated allocation decisions can be accomplished with very little overhead. All communication occurs along the normal flow of network traffic, and only a single real number is appended to every data packet.

6.3 Numerical Results

In this section, we compare the performance of message passing to the heuristics described in Section 6.1 as well as the optimal solution across as a set of random problem instances. These problem instances are described by a problem size parameter n . Each problem instance of size n consists of n users and n links. The assignment of users to links is made by uniformly sampling a bipartite graph of degree 10, so that each user is assigned a route along 10 links, each link is in the route of 10 users. Each link r is assigned a fixed capacity $b_r = 5$. The utility function of each user a is generated randomly, by setting z_a to an IID exponential random variable of mean 1, and setting $w_a \triangleq z_a$. This type of utility function corresponds to a “strongly correlated” regime for the multidimensional 0–1 knapsack problem (6.1) (Fréville, 2004). Here, the combinatorial nature of the underlying packing problem is most apparent and the problem is thought to be most difficult.

In these simulations, message passing is run for 1000 iterations, independent of the problem size. During each iteration t , a set of message-passing parameters $v^{(t)}$ is updated according to $v^{(t)} = (1 - \gamma)v^{(t-1)} + \gamma H v^{(t-1)}$, where a dampening factor of $\gamma = 0.5$ is used. An allocation decision $x^{(t)}$ is made by solving each activity manager’s problem, with one important modification: to ensure feasibility of the resulting allocation, the users are considered in order of decreasing values of $z_a + \sum_{r \in \mathcal{R}(a)} v_{r \rightarrow a}^{(t)}$. Each user is then greedily allocated capacity w_a if this is feasible, and is otherwise allocated capacity 0. This procedure is analogous to the greedy rounding procedures described in Section 6.1, and can similarly be implemented in a distributed fashion. The objective value of the best allocation seen in the 1000 iterations is reported.

Table 1 provides data on the performance of message passing versus the greedy and

Problem Size (n)	Message-Passing	Concave Approximation	Greedy
25	1.35% \pm 2.14	15.69% \pm 9.18	18.25% \pm 14.57
50	0.81% \pm 1.18	17.94% \pm 7.79	20.62% \pm 11.45
75	1.10% \pm 0.98	17.99% \pm 5.64	20.19% \pm 8.29
100	1.38% \pm 0.93	19.34% \pm 5.88	20.49% \pm 6.27
125	1.65% \pm 0.78	19.18% \pm 4.83	22.29% \pm 6.51

Table 1: A comparison of algorithms for Inelastic Rate Control, where the algorithms are compared for a collection of random problem instances of varying size. In each case, the average optimality gap (optimal = 0%) and the standard deviation of the optimality gap across problem instances is reported.

concave approximation heuristics. The algorithms are compared across a set of problem instances of various sizes. For each problem size, we sampled fifty instances and report the average percentage optimality gap relative to the globally optimal allocation, which is determined using a mixed integer solver. With each average we provide the standard deviation across instances to capture variation among samples.

Message-passing performs significantly better than either heuristic. Moreover, the optimality gap for message passing is very consistent, and typically is within 3% of the optimal objective value. The heuristics, on the other hand, have highly variable performance across problem instances.

For this class of problems, the efficiency metric employed by the greedy algorithm is constant: $e_a = 0.5$ for each user a . Hence, the greedy heuristic is particularly trivial: consider the users in an arbitrary order, and greedily assign capacity while maintaining feasibility. The concave approximation heuristic, which requires solution of a linear program, does not perform noticeably better.

Finally, note that our experiments involve networks with at most 125 users. These are the largest problems for which our mixed integer solver (ILOG CPLEX 9.1) could compute a global optimum. Message-passing can comfortably scale to much larger problem instances, up to 100,000's of users on a desktop workstation. Indeed, message passing could handle much larger problem instances than even our commercial LP solver (ILOG CPLEX 9.1), which was used in computing concave approximation solutions.

Acknowledgments

This research was partially stimulated by discussions with Stephen Boyd and Garrett van Ryzin. The first author was supported by a Benchmark Stanford Graduate Fellowship. This

research was supported in part by the National Science Foundation under Grant CMMI-0653876.

References

- Aldous, D. 1992. Asymptotics in the random assignment problem. *Probab. Th. Rel. Fields* **93** 507–534.
- Aldous, D. 2001. The $\zeta(2)$ limit in the random assignment problem. *Random Structures and Algorithms* **18** 381–418.
- Aldous, D., J. M. Steele. 2003. The objective method: Probabilistic combinatorial optimization and local weak convergence. H. Kesten, ed., *Discrete Combinatorial Probability*. Springer-Verlag.
- Aliprantis, C. D. 2001. A theory of value with non-linear prices. *Journal of Economic Theory* **100** 22–72.
- Arrow, K. J., L. Hurwicz, eds. 1977. *Studies in Resource Allocation*. Cambridge University Press, Cambridge, UK.
- Aurell, E., U. Gordon, S. Kirkpatrick. 2005. Comparing beliefs, surveys, and random walks. *Advances in Neural Information Processing Systems 18*. MIT Press, Cambridge, MA.
- Bayati, M., D. Shah, M. Sharma. 2005. Maximum weight matching via max-product belief propagation. *International Symposium of Information Theory*. Adelaide, Australia.
- Benedetto, S., D. Divsalar, G. Montorsi, F. Pollara. 1996. Soft-output decoding algorithms in iterative decoding of turbo codes. *JPL TDA Progress Report* **42** 63–87.
- Berliant, M., K. Dunz. 1990. Nonlinear supporting prices: The superadditive case. *Journal of Mathematical Economics* **19** 357–367.
- Berrou, C., A. Glavieux, P. Thitimajshima. 1993. Near Shannon limit error-correcting coding and decoding. *Proc. Int. Communications Conf.*. Geneva, Switzerland, 1064–1070.
- Bertsekas, D. P. 1982. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, New York.

- Braunstein, A., M. Mézard, R. Zecchina. 2005. Survey propagation: An algorithm for satisfiability. *Random Struct. Algorithms* **27** 201–226.
- Braunstein, A., R. Zecchina. 2004. Survey propagation as local equilibrium equations. Preprint. URL: <http://arxiv.org/abs/cond-mat/0312483>.
- Coughlan, J., S. Ferreira. 2002. Finding deformable shapes using loopy belief propagation. *European Conference on Computer Vision*.
- Fazel, M., M. Chiang. 2005. Network utility maximization with nonconcave utilities using sum-of-squares method. *Proceedings of the 44th Conference on Decision and Control*.
- Freeman, W. T., E. C. Pasztor, O. T. Carmichael. 2000. Learning low-level vision. *Intl. J. Computer Vision* **40** 25–47.
- Freeman, W. T., Y. Weiss. 2001. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory* **47** 736–744.
- Fréville, A. 2004. The multidimensional 0–1 knapsack problem: An overview. *European Journal of Operational Research* **155** 1–21.
- Frey, B. J. 1998. *Graphical Models for Machine Learning and Digital Communication*. M.I.T. Press, Cambridge, MA.
- Gamarnik, D., T. Nowicki, G. Swirszcz. 2005. Maximum weight independent sets and matchings in sparse random graphs. exact results using the local weak convergence method. *Random Structures and Algorithms* **28** 76–106.
- Gamarnik, D., T. Nowicki, G. Swirszcz. 2006. Maximum weight independent sets and matchings in sparse random graphs. exact results using the local weak convergence method. To appear in *Random Structures and Algorithms*.
- Hande, P., S. Zhang, M. Chiang. 2005. Distributed rate allocation for inelastic flows. Submitted to IEEE/ACM Transactions of Networking.
- Kellerer, H., U. Pferschy, D. Pisinger. 2004. *Knapsack Problems*. Springer, Berlin.

- Kelly, F. 1997. Charging and rate control for elastic traffic. *European Transactions on Telecommunications* **8** 33–37.
- Lee, J. W., R. R. Mazumdar, N. B. Shroff. 2005. Non-convex optimization and rate control for multi-class services in the Internet. *IEEE/ACM Trans. on Networking* **13** 841–853.
- Leone, M., A. Pagnani. 2005. Predicting protein functions with message passing algorithms. *Bioinformatics* **21** 239–247.
- Letovsky, S., S. Kasif. 2003. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics* **19 Suppl 1** i197–204.
- Malioutov, D. M., J. K. Johnson, A. S. Willsky. 2006. Walk-sums and belief propagation in Gaussian graphical models. *Journal of Machine Learning Research* **7** 2031–2064.
- Maneva, E. N., E. Mossel, M. J. Wainwright. 2005. A new look at survey propagation and its generalizations. *Proceedings of the Symposium on Discrete Algorithms*.
- McCallum, A. 2003. Efficiently inducing features of conditional random fields. *Uncertainty in Artificial Intelligence: Proceedings of the Nineteenth Conference*.
- Mézard, M., G. Parisi, R. Zecchina. 2002. Analytic and algorithmic solutions to random satisfiability problems. *Science* **297** 812–815.
- Moallemi, C. C., B. Van Roy. 2006. Consensus propagation. *IEEE Transactions on Information Theory* **52** 4753–4766.
- Moallemi, C. C., B. Van Roy. 2007. Convergence of the min-sum algorithm for convex optimization. Tech. rep., Management Science & Engineering Department, Stanford University.
- Moallemi, C. C., B. Van Roy. 2009. Convergence of min-sum message passing for quadratic optimization. *IEEE Transactions on Information Theory* **55** 2413–2423.
- Montanari, A., B. Prabhakar, D. Tse. 2005. Belief propagation based multi-user detection. *Proceedings of the Allerton Conference on Communication, Control, and Computing*.
- Montanari, A., D. Tse. 2006. Analysis of belief propagation for non-linear problems: The example of CDMA. *Proceedings of the IEEE Information Theory Workshop*.

- Mordukhovich, B. S. 2005. Nonlinear prices in nonconvex economies with classical Pareto and strong Pareto optimal allocations. *Positivity* **9** 541–568.
- Pinto, D., A. McCallum, X. Wei, W. B. Croft. 2003. Table extraction using conditional random fields. *Proceedings of the ACM SIGIR*.
- Richardson, T., R. Urbanke. 2001a. The capacity of low-density parity check codes under message-passing decoding. *IEEE Transactions on Information Theory* **47** 599–618.
- Richardson, T., R. Urbanke. 2001b. An introduction to the analysis of iterative coding systems. *Codes, Systems, and Graphical Models, IMA Volume in Mathematics and Its Applications*. Springer, 1–37.
- Rockafellar, R. T. 1970. *Convex Analysis*. Princeton University Press, Princeton, NJ.
- Rockafellar, R. T. 1974. Augmented lagrange multiplier functions and duality in nonconvex programming. *Siam Journal of Control* **12** 268–285.
- Rusmevichientong, P., B. Van Roy. 2001. An analysis of belief propagation on the turbo decoding graph with Gaussian densities. *IEEE Transactions on Information Theory* **47** 745–765.
- Shenker, S. 1995. Fundamental design issues for the future Internet. *IEEE Journal on Selected Areas in Communications* **13** 1176–1188.
- Spence, M. 1977. Nonlinear prices and welfare. *Journal of Public Economics* **8** 1–18.
- Sun, J., Shum H. Y., N. N. Zheng. 2002. Stereo matching using belief propagation. *European Conference on Computer Vision*.
- Sutton, C., K. Rohanimanesh, A. McCallum. 2004. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *Proceedings of the International Conference for Machine Learning*.
- Talagrand, M. 2003. *Spin glasses: a challenge for mathematicians*. Springer.
- Tatikonda, S., M. I. Jordan. 2002. Loopy belief propagation and Gibbs measures. *Uncertainty in Artificial Intelligence: Proceedings of the Eighteenth Conference*.

Wainwright, M. J., T. Jaakkola, A. S. Willsky. 2004. Tree consistency and bounds on the performance of the max-product algorithm and its generalizations. *Statistics and Computing* **14** 143–166.

Weiss, Y., W. T. Freeman. 2001. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation* **13** 2173–2200.