

Discussion Paper No. 865

**DECENTRALIZED DYNAMIC PROCESSES
FOR
FINDING EQUILIBRIUM***

by

Stanley Reiter**
Northwestern University
and
Carl P. Simon***

April 1990

*We have benefitted from conversations with many friends and colleagues. Among them are J. Jordan, L. Hurwicz and E. Green. We owe special thanks to Michael Chwe, who has programmed and run some computer trials. They are, of course, not responsible for what we have done or not done with their suggestions. This reasearch has had the support of the National Science Foundation.

**Northwestern University

***The University of Michigan and I.M.A.

Discussion Paper No. 865
DECENTRALIZED DYNAMIC PROCESSES

FOR
FINDING EQUILIBRIUM*

by
Stanley Reiter**

and
Carl P. Simon***

*We have benefitted from conversations with many friends and colleagues. Among them are J. Jordan, L. Hurwicz and E. Green. We owe special thanks to Michael Chwe, who has programmed and run some computer trials. They are, of course, not responsible for what we have done or not done with their suggestions. This research has had the support of the National Science Foundation.

**Northwestern University

***The University of Michigan and I.M.A.

DECENTRALIZED DYNAMIC PROCESSES FOR FINDING EQUILIBRIUM

STANLEY REITER*† AND CARL P. SIMON†

Abstract. This paper describes a class of decentralized dynamic processes designed to converge to equilibrium when the equilibrium equations are linear. These processes can also be viewed as distributed algorithms for solving systems of linear equations, or as learning algorithms. The class includes processes that use a message space larger by one binary digit than the space in which the equilibrium exists. However, memory and time requirements increase exponentially with the number of agents (equations).

Section 1.

We present a class of decentralized dynamic processes designed to converge to message equilibrium. These processes may also be regarded as distributed algorithms for computing solutions of the distributed system of equations that characterize equilibrium. They may also be interpreted as algorithms for learning. Unlike most dynamic processes that have been used to study stability of economic mechanisms, these processes are not given by differential or difference equations that determine trajectories in the space in which the message equilibria live and in which convergence to equilibrium is to take place. The case considered here is that in which the equilibrium conditions are linear. In that case equilibrium is stabilized with a message space bigger by just one binary digit than the minimal message space needed for static equilibrium. Before discussing the processes in detail, we shall briefly discuss the background of the problem and the earlier work on stability of equilibrium.

Comparison of ways of organizing economic activity should include comparison of dynamic performance. Leo Hurwicz's original formalization [8] of "adjustment process" was explicitly dynamic. An adjustment process is a system of first order temporally homogeneous difference equations, together with an outcome function. Thus, in one version of Hurwicz's original formulation (not the most general) there are N agents: each agent has a space of characteristics, denoted E^i for agent i . The space of environments is

$$E = E^1 \times \cdots \times E^N.$$

*Northwestern University.

†We have benefitted from conversations with many friends and colleagues. Among them are J. Jordan, L. Hurwicz and E. Green. We owe special thanks to Michael Chwe, who has programmed and run some computer trials. They are, of course, not responsible for what we have done or not done with their suggestions. This research has had the support of the National Science Foundation.

†The University of Michigan and I.M.A.

Each agent has a message space M^i and the messages sent by agent i are chosen according to the response function

$$f^i : M \times E^i \rightarrow M^i \quad \text{for } i = 1, \dots, N,$$

where

$$M = M^1 \times \dots \times M^N$$

is the message space of the adjustment process. Here it is implicit that agent i knows his characteristic e^i , but does not know e^j for $j \neq i$. The process is given by a system of difference equations

$$m^i(t+1) = f^i(m(t), e^i), \quad \text{for } i = 1, \dots, N,$$

where $m^i \in M^i$, and $e^i \in E^i$. The equilibria are the stationary points of this system of difference equations, and may be described by the system of equilibrium equations

$$g^i(m, e^i) = f^i(m, e^i) - m^i = 0.$$

(The outcome function, which maps equilibrium messages into outcomes, is not relevant for our analysis and so is omitted here.)

In spite of this explicitly dynamic beginning, between 1960 and 1979 research on mechanisms was almost completely devoted to studying their equilibrium or static properties. Equilibrium can be defined directly without reference to a dynamic process¹. On the informational side, research concentrated on static realization, that is, on decentralized mechanisms capable of realizing a given goal function on a specified class of environments (e.g., Pareto optimal allocations on environments with nonconvexities, such as increasing returns) and on the size of the message space required for realization.

However, there had already been research into the dynamics of specific processes. For example, the work of Arrow and Hurwicz [1], and Arrow, Hurwicz and Block [2] on the price tatonnement antedates formal study of decentralized mechanisms in the more general framework. These authors established global stability of the price tatonnement on a subclass of the classical environments (gross substitutes), but no stability result was obtained for the full class. Scarf's example [17] showed that this could not be done, i.e., that the price tatonnement is not stable on the full set of classical economic environments.

Smale [18] studied a globalization of the Newton process, which is convergent but uses a good deal more information than the usual price tatonnement — essentially all the partial derivatives of the demand functions.

¹See for example Mount and Reiter [11]

Saari and Simon [15] showed that for mechanisms defined by systems of differential equations almost none of this additional information can be ignored while ensuring convergence. Saari [14] showed the analogous result for mechanisms defined by systems of difference equations.

Hurwicz, Radner and Reiter ([6] and [7]) introduced the B -process, a stochastic decentralized mechanism that is dynamic (a Markov process) and that converges with probability one to Pareto optima on a class of environments that includes the classical ones, but also includes environments with indivisibilities or other nonconvexities. In terms of informational requirements, while the messages used in the B -process are essentially approximations of points in the commodity space, the process involves the use of small portions of indifference surfaces in some cases. Analogues of the classical welfare theorems are proved for these environments.

Recent work by Bala [3] and subsequently by Bala, Majumdar and Mitra [4] studies the speed of convergence of the B -process. These authors also prove convergence when the commodity space includes some indivisible and some divisible commodities, and introduce modifications that improve informational properties. However, the B -process is not formally an adjustment process in the sense of Hurwicz's 1960 definition.

Reiter [13] studied an example with 2 persons, 2 goods, and a Walrasian goal function on a class of exchange environments with quadratic utility functions. It was known that in this example the minimum message space needed for static realization is 2-dimensional ([9] and [11]). The question studied was whether there could be an adjustment process using a 2-dimensional message space whose equilibria are stable. By analyzing the characteristic roots, Reiter [13] showed that for difference equations with non-zero linear part, there is an open set of parameter values (in $E = R^4$) on which the system is not locally stable. This result was suggestive, but not complete, since it did not rule out difference equations with vanishing linear part. In addition, a (globally) stable adjustment process was constructed for this example, but using a larger space. This stable process can be regarded as an adjustment process with a 4-dimensional message space, or as a process of the kind later formalized by Jordan [10] using the two dimensional static message space as the space of *state messages*, and another two dimensional space of *control messages*.

Subsequent papers by Jordan [10], Mount and Reiter [12] and Williams [19] studied differential equations rather than difference equations. Jordan introduced a broader class of processes, already mentioned above, which accommodates the traditional price tatonnement, a process not formally included in the class of adjustment processes as defined in [8]. These papers differ in the class of processes allowed, in the concept of stability used, and in other assumptions and conclusions. Their impact is that a decentralized process capable of stabilizing the equilibria of a static mechanism (that realizes a given goal function) requires more information, i.e., a message space larger in dimension than the minimum needed for static realization or verification of equilibrium. The analysis in these papers is not restricted to linear equilibrium conditions.

These papers also made it clear that there are open sets of environments on which there are stable processes. Saari and Williams [16] showed that the agents could provide enough information (in a decentralized way) to identify a region of the parameter space in which the environment lives and thus to permit choice of an appropriate differential equation system. This involves a 2-stage process, thus not temporally homogeneous, and an increase in the size of the message space to include the finite number of messages needed in the first stage².

Almost all of the analysis of dynamic processes has been done by studying differential equations. This seems quite natural, since there is considerable mathematical knowledge about differential equation systems (and about difference equations), as compared to other types of dynamic processes. Consideration of examples referred to earlier suggests that difference or differential equations use information in a more sensitive and responsive way than is necessary to achieve convergence to equilibria. In the presence of decentralization requirements, this excessive sensitivity is potentially a source of instability. A differential equation process needs to know at every point in the space a direction or set of directions in which to move. When the class of environments is large, the minimal (static) message space does not allow the *decentralized* transfer of enough information to pinpoint the correct decision at each point. Therefore, a larger message space is needed. Indeed, the work of Saari and Simon [15], Jordan [10] and Williams [19] shows that any stable process given by differential equations must use nearly all the partial derivatives of the equilibrium equations.

Processes like the price tatonnement involve “black boxes”, in the sense that differential equations can describe the results of the functioning of a system that is not otherwise specified, and whose existence is not established. There may be no fully specified decentralized institutional arrangement that exhibits the dynamic behavior described by the differential equation.

It is a commonplace metaphor to see an economic process as a way in which the economic organization computes an equilibrium. One can reverse the perspective and consider any algorithm or computational procedure that is capable of computing economic equilibria as a possible dynamic process for the economy. Decentralization requirements rule out many such processes, including various fixed point algorithms. Considerations of this sort led to the reconsideration of the example of [13], and to generalizations of that example in which the number of agents (and therefore equations) may be more than two, while the linearity of the individual equations is retained.

The class of processes presented in this paper include some that are globally stable on

²Temporal homogeneity and convergence from arbitrary initial conditions are important properties of dynamic economic processes, because together they ensure convergence to the new equilibrium without additional information when an underlying environmental parameter changes and equilibrium is displaced. These properties do not have the same importance when the process is an algorithm designed to converge to the solution of a given system of equations characterizing a mixed equilibrium.

the full class of environments considered. In particular, the examples of Scarf [17] and of Reiter [13] are stabilized by such processes.

The processes in this class use a message space that is the minimal message space needed for static realization augmented by one bit. That is, if M is the minimal space of equilibrium messages, then $M \sqcup \{1, -1\}$ (the disjoint union of M and $\{1, -1\}$) suffices for the dynamics³.

But, the size of the message space required, which may be interpreted as a measure of capital costs of communication associated with a dynamic mechanism, is not the only relevant dimension of informational costs. Computational complexity is also important. While the computations required vary among members of this class of processes (algorithms), the class includes at least one process for which the requirements are:

- (1) a density function to generate points of the space;
- (2) the evaluation of the sign of a function at a point;
- (3) the computation of the average of two points.

The dynamic processes we describe have a number of advantages over the other mechanisms we have mentioned. First, as we have emphasized, they are distributed so that each agent uses only his own private information and not that of other agents. Second, no private agent is required to evaluate her equilibrium function, but only to determine its sign at a given point m . This is, of course, a much weaker requirement, whether interpreted as a computation or as an act of behavior. For example, in the Walrasian setting, it is more natural for an agent to estimate whether a given allocation gives him more or less than he wants (positive or negative excess demand) than it is to quantify the precise amount by which the allocation differs from his demands (the exact value of the excess demand function). Finally, the procedure which underlies this algorithm entails more natural behavior and decision processes from the agents and coordinator than do other mathematical procedures in the literature, such as the global Newton method or various triangulation schemes.

These processes can also be interpreted as distributed algorithms for solving systems of linear equations, where each equation of the system is known by only one processor (computer) in a network. Since the standard methods for solving linear equations work on the whole matrix, i.e., are not distributed, it is difficult to compare this class of processes with the standard ones.

The processes presented in this paper can also be interpreted as ways of learning about sets determined by distributed information.

The rest of this paper is organized as follows. Section 2 describes the class of processes, and in particular one process and a variant of it, in the context of the two agent example

³In the formal development of these processes, the messages in $M \sqcup \{1, 0, -1\}$ are used, 0 being used to identify equilibrium, but the processes use only the additional messages $\{1, -1\}$; the element 0 comes in only in the limit.

mentioned above. The presentation is informal and intuitive. In Section 3 the processes are formally defined for N agents, i.e., for a distributed system of N linear equations, where $N \geq 2$ and a proof of convergence is given.

Section 2. A Computational Dynamic Process.

We first describe the process in informal, intuitive terms in the context of the two agent, two commodity example. This may seem rather special, but in fact the transition to higher dimensions brings few new significant qualitative properties⁴.

The example consists of two (private) agents and two commodities, with each agent characterized by two parameters so that $E^i = \mathbf{R}^2$, $i = 1, 2$, with $e^i = \theta^i = (\theta_1^i, \theta_2^i) \in \mathbf{R}^2$ for $e^i \in E^i$. The equilibrium message correspondence of private agent i is given by equation:

$$(1) \quad g^i(m, \theta^i) = 0 \quad i = 1, 2,$$

so that the equilibrium message $m^* = (m_1^*, m_2^*)$ is the solution of the equation system (1). It is known for the (Walrasian) performance function underlying this example that the minimal message space for (static) realization is two-dimensional, and that the equation for each agent is linear (affine) in the message components with the parameters of agent i as coefficients, (and also linear in the coefficients). We will assume further that each of the functions

$$g^i : M \times \Theta^i \rightarrow R \quad i = 1, 2,$$

changes sign on the sets given by (1) and only there, so that $g^i(m, \theta^i) > 0$ for all m in one half-space determined by the corresponding equation of (1), and $g^i(m, \theta^i) < 0$ for all m in the other.

Agent i knows his own parameter value θ^i and his own function g^i . In addition there is the role of *coordinator*, a role that may be played by one of the agents or by a special additional agent recruited or created for this purpose. The coordinator can address messages to each agent and identify individual messages received from them. If one of the private agents serves as coordinator, she knows what the coordinator knows in addition to her information as a private agent.

We consider this example in two cases: first, when one of the private agents, say agent 2, plays the role of coordinator; second, when an additional agent, agent 0, is the coordinator.

It is convenient to describe the process in two phases, Phase I and Phase II⁵. In each Phase the coordinator sends messages to the private agents. In Case 1 the coordinator, who is agent 2, sends messages to agent 1, and receives agent 1's response to each message

⁴In the case of Algo I, higher dimensions bring no new properties, while for Algo II, a variant of I, a significant difference is pointed out on page 16.

⁵It is not strictly necessary to distinguish two phases. The behavior of agents can be characterized by a single set of conditional rules, not time dependent, so that the process is temporally homogeneous.

sent. In Case 2 the coordinator sends separate messages to agents 1 and 2 and receives their responses. We consider Case 1 first.

Agent 2, acting as coordinator and as private agent, knows that any equilibrium must satisfy her equilibrium condition,

$$(2) \quad g^2(m, \theta^2) = 0,$$

and can use this knowledge in the process. Therefore her message at any step t of the process is a point satisfying her equilibrium condition. Let $m(t)$ denote the message sent by the coordinator to agent 1 at step t . The response of agent 1 at t is:

$$\begin{aligned} 1 & \quad \text{if } g^1(m(t), \theta^1) > 0, \\ 0 & \quad \text{if } g^1(m(t), \theta^1) = 0, \\ -1 & \quad \text{if } g^1(m(t), \theta^1) < 0. \end{aligned}$$

We may visualize the Agent 2 as having two *storage registers* in which messages sent are recorded according to the responses of agent 1. These are labeled “1” and “-1” respectively. The message $m(t)$ is stored in register 1, replacing whatever was there, if the response of agent 1 to it at time t is 1, and in register -1 if the response is -1. The initial state of these registers is “empty”. If at any time the response of agent 1 is 0, then the message sent by the coordinator at that time is the equilibrium message, and the process stops.

Phase I.

The coordinator has a density function on \mathbf{R}^2 that is positive on the line given by (2) and 0 elsewhere. In other words, the coordinator works only with points in \mathbf{R}^2 which satisfy (2). The coordinator’s message is chosen independently at each time according to this density function. Phase I ends at the first time t at which both storage registers are nonempty. At that time there are two messages, s^1 and s^{-1} in registers 1 and -1 respectively. These form the initial state for Phase II. Denote this state by $s(0) = (s^1, s^{-1})$. At this point it is known by the coordinator that the solution to (1) (i.e., the equilibrium) lies in the interval $[s^1, s^{-1}]$.

It is convenient to represent this structure by a function

$$\lambda = \lambda_1 : M \rightarrow \{1, 0, -1\},$$

called a *labeling function* for agent 1. A set of messages $\{m', m''\}$ in the line

$$\{x \in \mathbf{R}^2 \mid g^2(x, \theta^2) = 0\}$$

is *completely labeled* if one of the messages is labeled 1 and the other -1. An initial state for Phase II is a completely labeled set of messages.

Phase II.

Phase II consists of a procedure (an algorithm) for replacing a completely labeled set by a “smaller” one, in such a way that the sequence of completely labeled sets converges to a point. That point must be the solution of the system of equations (1). There are many different procedures or algorithms for transforming a completely labeled set into another smaller one, thereby generating a class of processes. These processes would generally have different informational and computational requirements. The present case illustrates a common feature of such processes.

In the (essentially one-dimensional) example under discussion, one method of replacing a completely labeled set $\{s^1, s^{-1}\}$ by a smaller one is to choose a point in the interval (line segment)

$$[s^1, s^{-1}] = \{x \in \mathbb{R}^2 \mid x = \alpha s^1 + (1 - \alpha)s^{-1}, 0 < \alpha < 1\}.$$

This can be done by specifying a value for α , e.g., $\alpha = \frac{1}{2}$, and choosing

$$m = \alpha s^1 + (1 - \alpha)s^{-1}$$

The coordinator then transmits m to Agent 1, who replies⁶

$$\lambda_1(m) = \begin{cases} +1 & \text{if } g^1(m, \theta^1) > 0 \\ -1 & \text{if } g^1(m, \theta^1) < 0. \end{cases}$$

This gives a new completely labeled state, in which m replaces s^1 if $\lambda_1(m) = 1$ and s^{-1} if $\lambda_1(m) = -1$. The new completely labeled set is smaller in the sense that

$$[m, s^x] \subset [s^1, s^{-1}],$$

where $x \in \{1, -1\}$ and $x \neq \lambda_1(m)$. Figure 1 shows the situation graphically. This is, of course, the familiar but essentially one-dimensional method of bisection. The method of bisection is a well-studied (see, for example, Conte and de Boor [5]) method for finding the zeroes of a real-valued continuous function f of one-variable using only information about the sign of f and not its actual values. The mechanisms we describe in this report can be considered as n -dimensional *distributive* generalizations of the method of bisection.

⁶Here we ignore the possibility that $\lambda_1(m) = 0$, i.e., that m satisfies $g^1(m, \theta^1) = 0$.

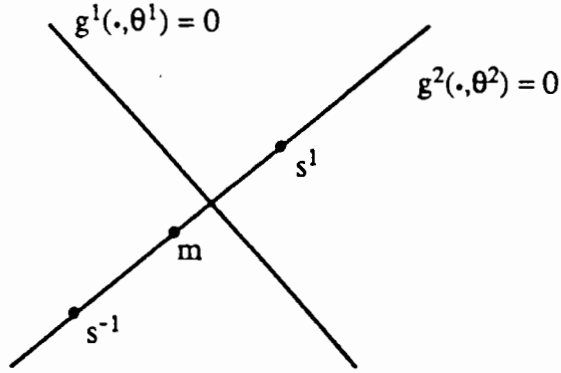


Figure 1.

It is intuitively evident that this procedure generates a decreasing sequence of intervals that converges to the solution m^* .

While this essentially one-dimensional case is overly special, it still permits the existence of several different algorithms for going from one completely labeled set to another. These include probabilistic choice of a point in the interval $[s^1, s^{-1}]$.

Notice that for this special case of two agents with one acting as coordinator, the linearity of g^1 and g^2 is an unnecessary assumption. One only needs that g^1 and g^2 are continuous and that the set $\{x \mid g^2(x, \theta^2) = 0\}$ is a parametrized curve in the plane.

In Case 2, the coordinator is Agent 0, and there are two private agents, Agents 1 and 2. This case allows almost all the significant features of the general case⁷.

In this case the coordinator knows only that the space in which the solution is sought in \mathbf{R}^2 . For $i = 1, 2$, agent i knows only the function $g^i(\cdot, \theta^i)$, and given any point x of \mathbf{R}^2 can determine the sign of $g^i(x, \theta^i)$ ⁸.

Phase I.

In Phase I the coordinator uses a density that is positive everywhere in \mathbf{R}^2 to select the initial trial points. At each step of Phase I a chosen point x is transmitted to each agent i , who responds with $\lambda_i(x)$, the sign of $g^i(x, \theta^i)$.

Phase I ends when the coordinator has accumulated a completely labeled set. In \mathbf{R}^2 , a completely labeled set consists of four points, say v_1, v_2, v_3, v_4 , such that $\lambda(\{v_1, v_2, v_3, v_4\}) = \{1, -1\}^2$.

Some aspects of Phase I deserve further comment. First, in an unbounded domain, such as \mathbf{R}^2 , it might take a long time to find a completely labeled set, using a fixed density

⁷Case 2 can also arise as Case 1 for $N = 3$ with the role of coordinator played by Agent 3.

⁸As we mentioned earlier, this is a weaker requirement than being able to evaluate $g^i(x, \theta^i)$. (For example, it would be sufficient to be able to calculate the sign of the first digit of $g^i(x, \theta^i)$.) In some setting it would be natural to consider determining the sign of $g^i(x, \theta^i)$ to be a behavioral act rather than an act of computation.

positive everywhere on \mathbf{R}^2 . Several devices suggest themselves to deal with this problem. For example, choose a bounded region in \mathbf{R}^2 , such as the square $S(s)$ of side $s \geq 1$ centered at the origin. Begin with a density positive on this square and zero outside it. If after a certain time no completely labeled set is found, sample from the square $S(s^2)$ of side s^2 centered at the origin (perhaps with extra weight given to $S(s^2) \setminus S(s)$), and so on. Generally speaking, one could increase the variance of the density exponentially if it is taking too long to find a completely labeled set.

Second, the problem of convergence to equilibrium is naturally posed under the assumptions that the equilibrium exists and is unique. But in the computational setting this assumption is less natural. In the linear case Phase I can be used to help detect singularity, even though the matrix of the system of equations is not known by anyone. If certain labels are not observed by the coordinator for sufficiently many rounds of Phase I, then the probability that the corresponding region is non-empty becomes very small.

For example, suppose, for $N = 2$, that

$$g^2(m, \theta^2) = g^1(m, \theta^1) + c,$$

as shown in Figure 2, where the $+$ and $-$ signs indicate respectively the positive and negative half spaces defined by $g^i(\cdot, \theta^i) = 0$.

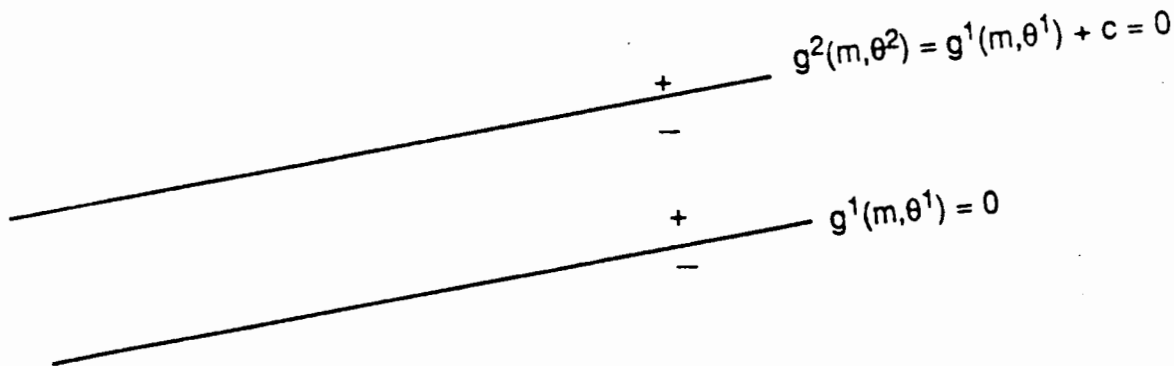


Figure 2.

In this case the label $(1, -1)$ can never be observed. Even using a fixed positive density on \mathbf{R}^2 the coordinator can eventually conclude that there is no solution, with prespecified probability of error. On the other hand, the coordinator may have to use many trials of Phase I to distinguish the case where the lines given by $g^i(m, \theta^i) = 0$, $i = 1, 2$ coincide, from the case in which they intersect but are very close to being parallel. We remind the reader that the singular matrices (matrices A for which the system $Ax = 0$ does not have a unique solution) are rare in that they form a set of measure zero in the set of all square matrices.

Finally, as noted above, Phase I provides the initial state for Phase II. If the Phase II algorithms are regarded as algorithms that converge only from certain initial conditions, i.e., minimal completely labeled sets, but not from all initial conditions, then Phase I can be conceptually separated from Phase II, and the problem of finding an initial completely labeled set left open.

Phase II.

We describe two different processes for Phase II in this example: Algo I and Algo II. They share a common feature, which can be described first.

A completely labeled set in \mathbf{R}^2 consists of four points, say $V = \{v_1, v_2, v_3, v_4\}$. Let $\lambda = (\lambda_1, \lambda_2) : \mathbf{R}^2 \rightarrow \{-1, +1\}^2$ be the labeling function. For $\ell_2 = +1$ or -1 , write $(*, \ell_2)$ for the two labels $(-1, \ell_2)$ and $(+1, \ell_2)$. Think of $(*, \ell_2)$ as determining an edge in the unit square $[-1, 1] \times [-1, 1]$ in \mathbf{R}^2 , $[-1, +1] \times \{\ell_2\}$,

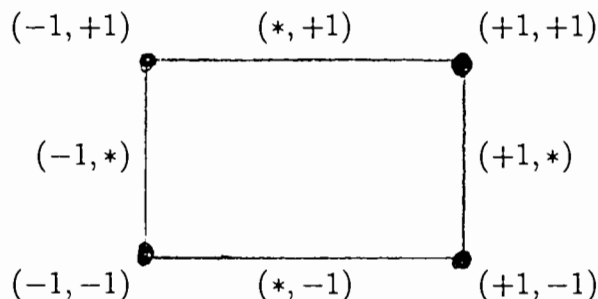


Figure 3.

as illustrated in Figure 3. (By abuse of notation, we also consider $(*, \ell_2)$ as denoting the two vertices on this edge: $(-1, \ell_2)$ and $(+1, \ell_2)$.) We use λ to carry these labels back to the completely labeled set V , so that we write $v \in V$ as $+-$, say, if $\lambda_1(v) = +1$ and $\lambda_2(v) = -1$. The edges of $[-1, 1] \times [-1, 1]$ correspond in a natural way to line segments in \mathbf{R}^2 , which we still call “edges”. For example, if $\lambda(v_1) = (+1, +1)$ and $\lambda(v_2) = (+1, -1)$, then we write $[v_1, v_2]$ to denote the line segment from v_1 to v_2 and note that it corresponds to the edge $(+1, *)$ in Figure 3. We call $[v_1, v_2]$ an *edge for agent 2* or a “2-edge”, because only the second agent’s function changes sign along $[v_1, v_2]$, that is, $[v_1, v_2]$ corresponds to $(+1, *)$ where the $*$ is in the *second* component. In Figure 4, the edges for agent 1 are $[v_1, v_2]$ (corresponding to $(*, +)$) and $[v_3, v_4]$ (corresponding to $(*, -)$). The edges for agent 2 are $[v_1, v_3]$ (for $(+, *)$) and $[v_2, v_4]$ (for $(-, *)$). Agent 1’s zero-set ($g^2(x, \theta^1) = 0$) cuts the two 1-edges, while agent 2’s do not; agent 2’s zero set ($g^2(x, \theta^2) = 0$) cuts the two 2-edges, while agent 1’s do not.

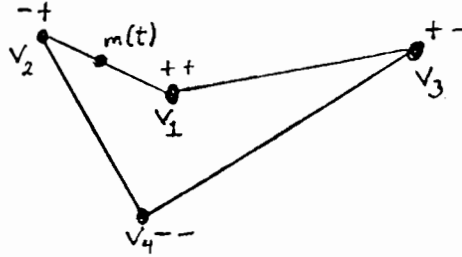


Figure 4

In each algorithm, the coordinator

- i) calculates the midpoint $m(t) = \frac{1}{2}(v_j + v_{j'})$ where $[v_j, v_{j'}]$ is an i -edge of the current completely labeled set of vertices, $V(t)$,
- ii) announces $m(t)$ to agent i , if the edge $[v_j, v_{j'}]$ is an edge of agent i , and
- iii) replaces with $m(t)$ the vertex whose label is the same as that of $m(t)$.

For example, in Figure 4, the edge $[v_1, v_2]$ has midpoint

$$m = \frac{1}{2}(v_1 + v_2)$$

Since this is an edge of agent 1,

$$\lambda_2(m) = \lambda_2(v_1) = \lambda_2(v_2).$$

Suppose that $\lambda_1(m) = \lambda_1(v_1) = +1$.

Form the new set of vertices $V(t+1) = \{m, v_2, v_3, v_4\}$, replacing v_1 with m , resulting in the polyhedron shown in Figure 5.

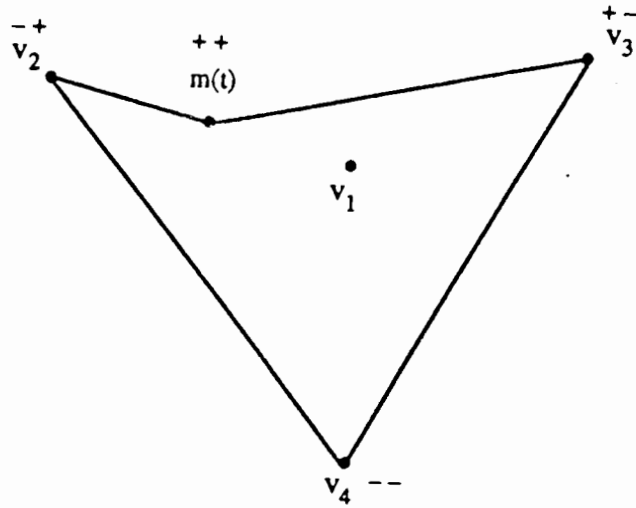


Figure 5

On the other hand, if $\lambda_1(m) = \lambda_1(v_2)$, then m replaces v_2 in $V(t+1)$ and the new set of vertices is $\{m, v_1, v_3, v_4\}$ and the new polyhedron is as in Figure 6.

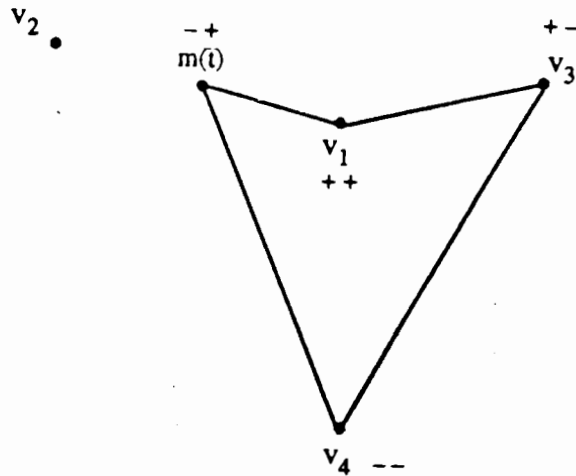


Figure 6

Call the replacement of v_i by m as in Figures 5 and 6 a *step* in the algorithm.

The two algorithms differ in the order in which edges are considered.

Algo I.

Let $V(t) = \{v_1(t), v_2(t), v_3(t), v_4(t)\}$ denote the completely labeled set at step t where $\lambda(v_i(t))$ is independent of t for each i .

First, the edges of agent 1 are considered in the order

$$[v_1(0), v_2(0)], [v_3(0), v_4(0)]$$

Let

$$m(0) = \frac{1}{2}(v_1(0) + v_2(0))$$

and

$$\rho(m(0)) = \begin{cases} v_1(0) & \text{if } \lambda_1(m(0)) = \lambda_1(v_1(0)) \\ v_2(0) & \text{otherwise .} \end{cases}$$

The set $V(1)$ of vertices at time 1 is

$$V(1) = (V(0) \setminus \{\rho(m(0))\}) \cup \{m(0)\}.$$

In words, the midpoint m of v_1 and v_2 replaces the vertex for which $\lambda(m) = \lambda(v_i)$; so that $V(1)$ is still a completely labeled set.

After considering both edges of agent 1, a process which we call agent 1's *round*, the coordinator works with the edges of agent 2, starting with the completely labeled set $V(2)$, and replacing vertices with midpoints of edges of agent 2 in turn.

Thus,

$$V(3) = (V(2) \setminus \{\rho(m(2))\}) \cup \{m(2)\}$$

where $m(2)$ is the mid-point of the first edge of agent 2 considered. This is the edge given by the vertices with labels, say, $(1, 1)$ and $(-1, 1)$, and

$$V(4) = (V(3) \setminus \{\rho(m(3))\}) \cup \{m(3)\}$$

where $m(3)$ is the mid-point of the segment whose vertices are those with labels $(1, -1)$ and $(-1, -1)$.

The process continues by successive rounds of cuts of the edges of agents 1 and 2 in turn.

The vertices $V(t)$ and the four associated line segments define polygons, which we call *boxes*. These boxes are, however, not necessarily convex. It is intuitively evident that eventually these boxes shrink to a point. However, that process of shrinking is not monotonic with respect to set inclusion; as Figures 4 and 5 show, the box corresponding to $V(t + 1)$ may properly include the box corresponding to $V(t)$. It is also the case that, although at each step the length of some edge is cut in half, the distance between the vertices that span that edge does not decrease monotonically in t . In fact, it is difficult to find any measure of size which decreases monotonically step by step in this process.

Before sketching Algo II, it is interesting to consider the information and computational requirements of the class of processes introduced here.

The process defined by either Algo I or II in Case 1 or Case 2 stabilizes the equilibrium in the example under discussion for all environments (θ 's) for which the system (1) is not degenerate. It does so with messages from the coordinator that are points of \mathbf{R}^2 , while messages from the private agents are one binary digit, i.e., "yes" or "no". Thus,

while conclusions of the various “impossibility” theorems that deal with informational requirements of local stability are not violated in the sense that the message space used by the class of processes presented here is larger than \mathbf{R}^2 , it is larger by only one binary digit⁹. This tells us that the processes presented here minimize communication, as measured by size of the message space.

It may also be of interest that the process presented here stabilizes the equilibrium in Scarf’s example of an environment with a unique *unstable* competitive equilibrium.

The computations required to carry out the Phase II algorithms are surprisingly small. The coordinator must record points of \mathbf{R}^2 and their labels. This requires two units of memory. The coordinator must calculate the average of two points of the space, which entails summing 2 pairs of numbers and dividing each sum by 2. Each private agent must determine whether his function is positive or negative at a given point of the space. In an economic, rather than a computational setting, this evaluation may represent a behavioral rather than a computational act. It is difficult to imagine a computational procedure or an adjustment process capable of finding equilibrium in this class of problems that involves less computation per step than this.

Algo II.

To anticipate higher dimension, the size of a completely labeled set goes up with the number N of agents as 2^N . Therefore, the idea of saving dimensions is attractive. In two dimensions this can easily be done as follows:

Choose $\varepsilon > 0$. Reduce the edges of agent 1 for T_1 rounds of cuts until they are all less than ε in length, as in Figure 7, without working with agent 2’s edges.

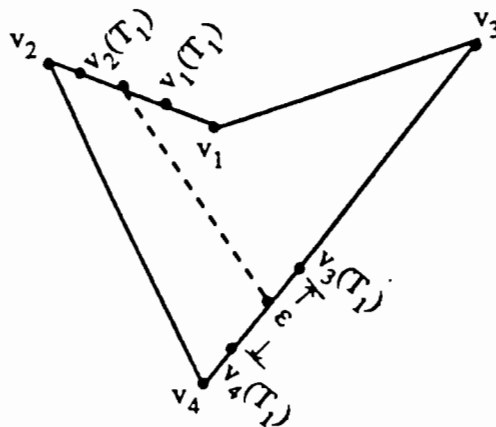


Figure 7

⁹This is true even in Case 1, where one of the agents acts as coordinator. Although the problem was reduced to one dimension, the coordinator must still send points of her line as points of \mathbf{R}^2 , since agent 1 must test them as points of \mathbf{R}^2 .

Then, the midpoints of the intervals $[v_1(T_1)_1 v_2(T_1)]$ and $[v_3(T_1)_1 v_4(T_1)]$ determine a line segment

$$\left[\frac{1}{2}(v_1(T_1) + v_2(T_1)), \frac{1}{2}(v_3(T_1) + v_4(T_1)) \right]$$

shown as a dotted line in Figure 7. The dialogue with agent 2 can be confined to this line, that is, to one-dimension, because the labels of agent 2 for these midpoints constitute a completely labeled set in that one-dimensional line. Reducing agent 2's interval to length less than ϵ , and taking the midpoint of that interval would guarantee that the result is within $\epsilon\sqrt{2}$ of the solution.

However, the same idea for $N = 3$ would encounter an additional difficulty. Agent 1 would have four edges, as shown in Figure 8, after reduction to length at most ϵ .

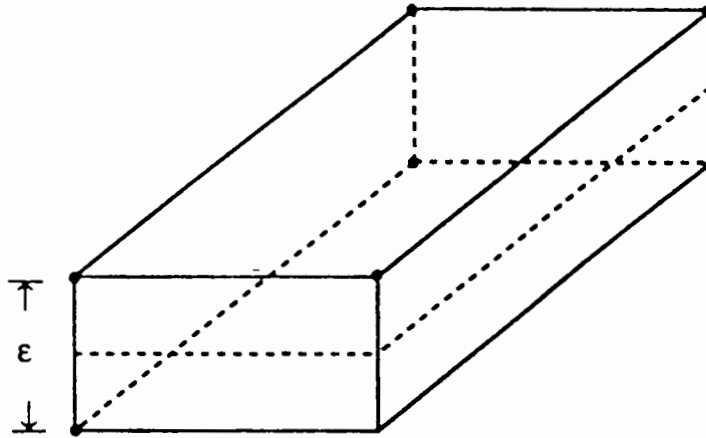


Figure 8

If we knew four points, one on each of these edges, that lay in a single plane, then in this plane the label of agent 1 could be ignored and the remaining $N - 1 = 2$ labels would be such that the four points would constitute a completely labeled set of vertices in that plane. In that case the remaining problem would be reduced from three to two dimensions.

But in general, the midpoints of the edges of Agent 1 do not lie in a plane. In order to carry out this reduction of dimension, one would have to use three midpoints to determine a plane, and then solve for the intersection of this plane with the fourth edge. This is a significant increase in computation. Alternatively, one could determine a plane using three midpoints and then go back to Phase I to determine a completely labeled set in an appropriately determined *plane* segment. But this too would entail additional computation. Depending on the particular details, it might or might not be worth doing those computations in exchange for the decrease in dimension.

In any case, the procedure of Algo II is the same as Algo I, except that several rounds of cuts are carried out for one agent before turning to the edges of the next. This may

be a more efficient procedure especially when the equations are sufficiently close to being orthogonal.

One can imagine many other ways to “shrink” a completely labeled set of vertices to achieve convergence. Each such way would in general involve different computational tasks and result in a different rate of convergence.

Section 3. Convergence of Phase II of Algo I and Algo II.

This section contains a formal description of Phase II of Algo I and a proof of its convergence. The proof for Algo II is similar.

We suppose that for the current environment θ the equilibrium functions

$$x \mapsto g^i(x, \theta^i)$$

are linear functions of x for each $i = 1, \dots, N$. We write $g_i(x, \theta^i)$ simply as $L_i(x)$. We further assume that the linear mapping $L = (L_1, \dots, L_N) : \mathbf{R}^N \rightarrow \mathbf{R}^N$ is nonsingular so that it has a unique zero in \mathbf{R}^N . In N dimensions, the labeling function $\lambda = (\lambda_1, \dots, \lambda_n) : \mathbf{R}^N \rightarrow \{-1, 0, 1\}^N$ is defined by

$$\lambda_i(x) = \begin{cases} -1 & \text{if } L_i(x) < 0 \\ 0 & \text{if } L_i(x) = 0 \\ +1 & \text{if } L_i(x) > 0. \end{cases}$$

for $i = 1, \dots, N$. For any subset A of \mathbf{R}^N , write $\lambda_A : A \rightarrow \{-1, 0, +1\}^N$ for λ restricted to A . A subset A of \mathbf{R}^N is called *completely labeled* if:

- (1) A consists of 2^N points
- (2) λ_A is onto $\{-1, 1\}^N$, or there exists a function $\bar{\lambda} : A \rightarrow \{-1, +1\}^N$ that is onto such that $\lambda_i(x) = \bar{\lambda}_i(x)$ for all x and i such that $\lambda_i(x) \neq 0$.

In other words, a set of 2^N points is completely labeled if either its points together take on all possible non-zero labels, or, if some labels are zero, they can be replaced by labels 1 or -1 in such a way that the result is a set taking all possible non-zero labels. With this definition, if the algorithm happens to hit upon a point that satisfies one of the equations $L_i(x) = 0$, and so gets the label 0, the coordinator can interpret that 0 as either 1 or -1 so as to make the set involved a completely labeled one. To simplify exposition, the possibility of hitting upon such (zero labeled) points will be ignored from now on. With this understanding, the function λ maps any completely labeled set of vertices onto $\{1, -1\}^N$ in a one-to-one fashion.

Let $\{-1, +1\}^{\sim i}$ denote the product of $(N - 1)$ copies of $\{-1, +1\}$ with the i^{th} factor replaced by the set $\{*\}$, i.e.,

$$\{-1, +1\}^{\sim i} = \underbrace{\{-1, 1\} \times \dots \times \{-1, 1\}}_{(i-1) \text{ times}} \times \{*\} \times \underbrace{\{-1, 1\} \times \dots \times \{-1, 1\}}_{(N-i) \text{ times}}.$$

Let $\lambda_{\sim i} = (\lambda_1, \dots, \lambda_{i-1}, *, \lambda_{i+1}, \dots, \lambda_N) : \mathbf{R}^N \rightarrow \{-1, 1\}^{\sim i}$ denote the corresponding labeling function which ignores L_i . Let $\pi^{\sim i} : \{-1, 1\}^N \rightarrow \{-1, 1\}^{\sim i}$ be the corresponding projection which replaces the i^{th} coordinate by a $*$.

For any ℓ in $\{-1, 1\}^{\sim i}$, there are two points in $(\pi^{\sim i})^{-1}(\ell)$, one with a $+1$ in the i^{th} component, which we call ℓ^+ , and one with a -1 in the i^{th} component, which we call ℓ^- . Geometrically, we think of the elements of $\{-1, 1\}^N$ as the 2^N vertices of the hypercube $[-1, +1]^N$ in \mathbf{R}^N and the elements ℓ in $\{-1, 1\}^{\sim i}$ to be edges joining ℓ^+ and ℓ^- in $[-1, +1]^N$. See Figure 3.

For any completely labeled set V of vertices in \mathbf{R}^N , with labeling function $\lambda_V : V \rightarrow \{-1, +1\}^N$, we say two points a and b in V determine an edge for agent i , or simply an i -edge, if $\lambda_{\sim i}(a) = \lambda_{\sim i}(b)$, i.e., $L_j(a)$ and $L_j(b)$ have the same sign for all $j \neq i$, but $L_i(a)$ and $L_i(b)$ have different signs.

Write $[a, b]$ for the line segment joining a and b in \mathbf{R}^N . The function L_i changes sign on $[a, b]$ and has a unique zero there. Each of the other $(N - 1)$ L_j 's has the same (non-zero) sign throughout $[a, b]$.

Algo I.

Let V_0 be the initial completely labeled set of vertices (perhaps given from Phase I) and let $\lambda_{V_0} : V_0 \rightarrow \{-1, +1\}^N$ be the labeling function. The coordinator chooses a sequence i_1, \dots, i_N determining the order in which the agents (or equations) will be considered. This choice can change as the algorithm proceeds, or to keep Algo I temporally homogeneous, it can be fixed for all time. For simplicity of notation, we will assume the natural order:

$$i_j = j, \quad j = 1, 2, \dots, N.$$

Agent 1 has 2^{N-1} edges, one for each label ℓ in $\{-1, 1\}^{\sim 1}$. For one such ℓ , let ℓ^+ and ℓ^- be the corresponding points in $(\pi^{\sim 1})^{-1}(\ell)$ in $\{-1, +1\}^N$. Let $a^+ = \lambda_{V_0}^{-1}(\ell^+)$ and $a^- = \lambda_{V_0}^{-1}(\ell^-)$ be the corresponding points of V_0 , so that $[a^+, a^-]$ is an edge for agent 1 in \mathbf{R}^N . The coordinator calculates in the first step, the midpoint of $[a^+, a^-]$

$$m = \frac{1}{2}(a^+ + a^-)$$

and communicates the point m to agent 1. Agent 1 responds with $\lambda_1(m)$. Define $\rho(m)$ by

$$\rho(m) = \begin{cases} a^+, & \text{if } \lambda_1(m) = \lambda_1(a^+) \\ a^-, & \text{if } \lambda_1(m) = \lambda_1(a^-) \end{cases}$$

The coordinator forms a new completely labeled set V_1 by replacing $\rho(m)$ by m :

$$V_1 = (V_0 - \{\rho(m)\}) \cup \{m\}.$$

This process is called a *step* in the algorithm. It is completed for each ℓ in $\{-1, +1\}^{\sim 1}$, i.e. , for each edge of agent 1. The implementation of these 2^{N-1} steps is called agent 1's *round*. The coordinator then works with the 2^{N-1} edges of agent 2, taking the midpoint of each and then forming a new completely labeled set by using the midpoint of each edge to replace the vertex of that edge which has exactly the same labels as the midpoint. In this way the coordinator carries out a round for each of the N agents, thus completing a *cycle* of Algo I. To prove that Algo I converges, we would like to find a function which decreases at a uniform rate at each step, like a Liapunov function for a system of differential equations. It turns out that the obvious candidates, e.g., volume of the enclosed polyhedron, sum of the lengths of all the edges, or aggregate distance of the vertices from the actual zero sets, do not work.

We will work with the function

$$K_i(V_t) = \max_j (|L_i(e_j^+)| + |L_i(e_j^-)|)$$

where V_t is the set of completely labeled vertices after round t , $\{e_1, \dots, e_{2^{N-1}}\}$ are the edges for agent i corresponding to V_t , and e_j^+ and e_j^- are the vertices of edge e_j ; so that $V_t = \{e_1^+, e_1^-, \dots, e_{2^{N-1}}^+, e_{2^{N-1}}^-\}$. Then, $K_i(V_t)$ is the length (in L_i -terms) of the largest of the images under L of the edges for agent i after round t . It is easy to see that $K_i(V_{t+1}) = \frac{1}{2}K_i(V_t)$ if the $(t+1)^{\text{st}}$ round is a round for agent i . However, during agent j 's rounds (for $j \neq i$), $K_i(V_t)$ can actually increase. We will see that the halving at agent i 's round uniformly dominates the combined increases during the other agents' rounds in any cycle so that there is a uniform decrease in K_i after each cycle.

Before presenting the general proof, let's see how the algorithm and proof works for an example with two agents. In Figure 9A, the lines represent the zero sets of L_1 and L_2 , and the four points $\{v_1, v_2, w_1, w_2\}$ form the initial set V_0 of completely labeled vertices. We'll work with $K_2(V)$ and focus on L_2 . Let $M_2 = -L_2$ for vertices $\{w_1, w_2\}$ below the $\{L_2 = 0\}$ set.

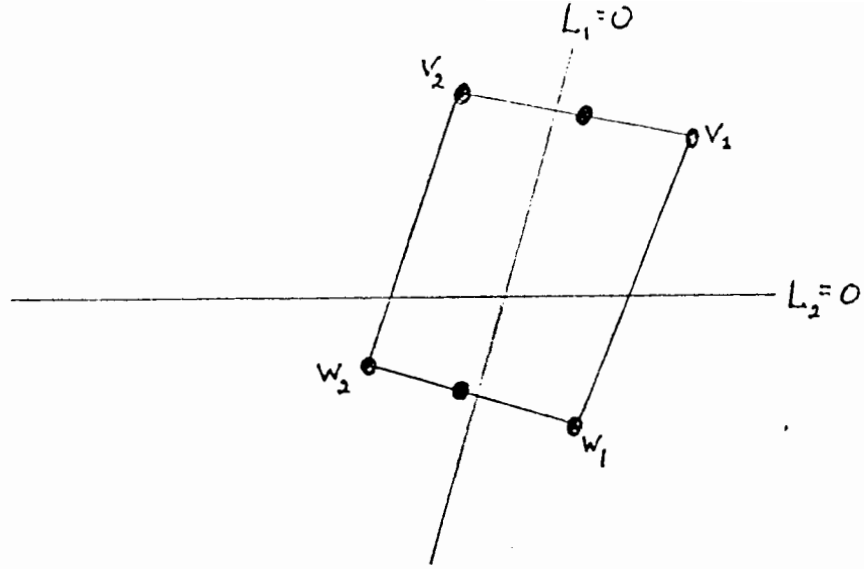


Figure 9A

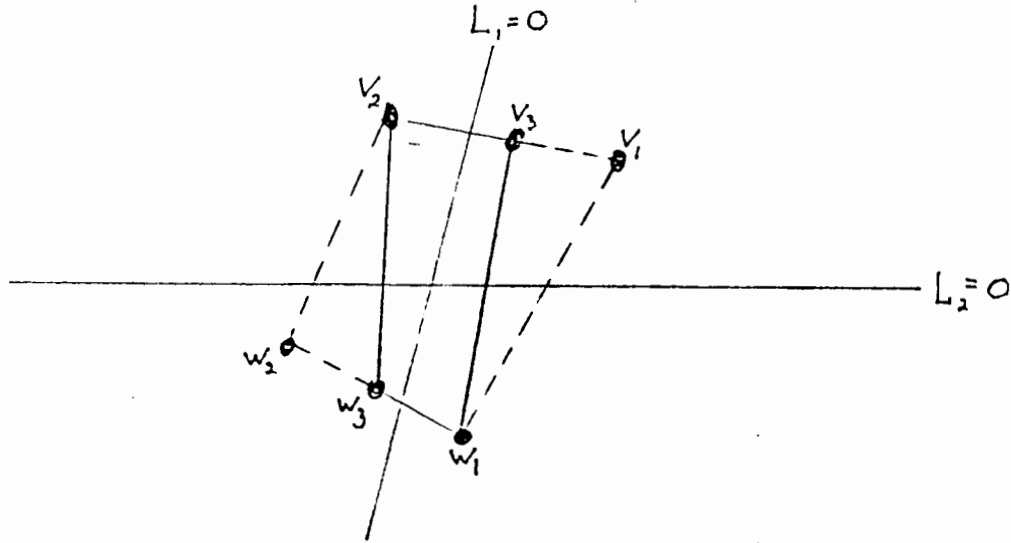


Figure 9B

Let $a_1 = L_2(v_1)$, $a_2 = L_2(v_2)$, $b_1 = M_2(w_1)$, $b_2 = M_2(w_2)$, all positive.

In round 1, the coordinator takes the midpoints $v_3 = \frac{v_1+v_2}{2}$ and $w_3 = \frac{w_1+w_2}{2}$ of agent 1's edges $[v_1, v_2]$ and $[w_1, w_2]$ respectively. Then, he forms a new completely labeled set replacing v_1 or v_2 by v_3 and w_1 or w_2 by w_3 . For example, in Figure 9B, v_3 replaces v_1 and w_3 replaces w_2 . Suppose that

$$K_2(V(0)) = |L_2(v_1)| + |L_2(w_1)| = a_1 + b_1,$$

so that $a_1 + b_1 \geq a_2 + b_2$. Then, if $V(1) = \{v_2, v_3, w_1, w_2\}$ as in Figure 9B, the new edges

for agent 2 are $[v_2, w_3]$ and $[v_3, w_1]$. Since

$$\begin{aligned}
|L_2(v_2)| + |L_2(w_3)| &= L_2(v_2) + M_2\left(\frac{1}{2}w_1 + \frac{1}{2}w_2\right) \\
&= L_2(v_2) + \frac{1}{2}M_2(w_1) + \frac{1}{2}M_2(w_2) \\
&= a_2 + \frac{1}{2}b_1 + \frac{1}{2}b_2 \\
&\leq \frac{1}{2}(a_1 + b_1) + (a_2 + b_2) \\
&\leq \frac{1}{2}(a_1 + b_1) + (a_1 + b_1) \\
&= \frac{3}{2}(a_1 + b_1) \\
&= \frac{3}{2}k_2(V(0)),
\end{aligned}$$

and

$$\begin{aligned}
|L_2(v_3)| + |L_2(w_1)| &= L_2\left(\frac{1}{2}v_1 + \frac{1}{2}v_2\right) + M_2(w_1) \\
&= \frac{1}{2}a_1 + \frac{1}{2}a_2 + b_1 \\
&\leq a_1 + b_1 + \frac{1}{2}(a_2 + b_2) \\
&\leq a_1 + b_1 + \frac{1}{2}(a_1 + b_1) \\
&= \frac{3}{2}(a_1 + b_1) \\
&= \frac{3}{2}K_2(V(0)),
\end{aligned}$$

it follows that $K_2(V(1)) \leq \frac{3}{2}K_2(V(0))$.

Now, carry out the round for agent 2. In Figure 9C, y_1 is the midpoint of 2-edge $[v_2, w_3]$ and y_2 is the midpoint of 2-edge $[v_3, w_1]$; y_1 replaces v_2 and y_2 replaces w_1 .

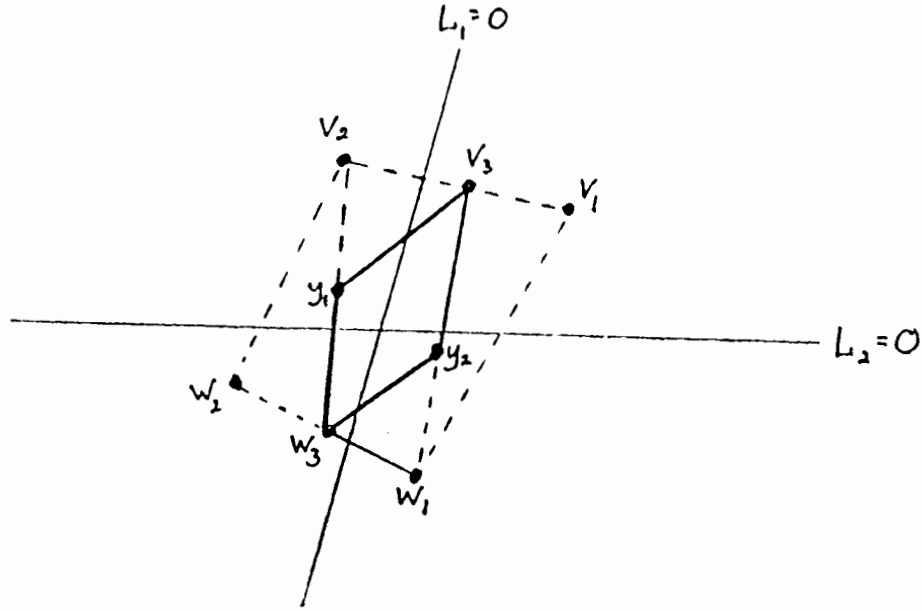


Figure 9C

The new completely labeled set $V(2) = \{y_1, v_3, w_3, y_2\}$.

$$\begin{aligned}
 |L_2(y_1)| + |L_2(w_3)| &= L_2\left(\frac{1}{2}v_2 + \frac{1}{2}w_3\right) - L_2(w_3) \\
 &= \frac{1}{2}L_2(v_2) - \frac{1}{2}L_2(w_3) \\
 &= \frac{1}{2}(|L_2(v_2)| + |L_2(w_3)|).
 \end{aligned}$$

Similarly,

$$\begin{aligned}
 |L_2(v_3)| + |L_2(y_2)| &= L_2(v_3) - L_2\left(\frac{1}{2}v_3 + \frac{1}{2}w_1\right) \\
 &= \frac{1}{2}L_2(v_3) - \frac{1}{2}L_2(w_1) \\
 &= \frac{1}{2}(|L_2(v_3)| + |L_2(w_1)|).
 \end{aligned}$$

Therefore,

$$\begin{aligned}
 K_2(V(2)) &= \max\{|L_2(y_1)| + |L_2(w_3)|, |L_2(v_3)| + |L_2(y_2)|\} \\
 &= \frac{1}{2} \max\{|L_2(v_2)| + |L_2(w_3)|, |L_2(v_3)| + |L_2(w_1)|\} \\
 &= \frac{1}{2}K_2(V(1)).
 \end{aligned}$$

Since $K_2(V(2)) = \frac{1}{2}K_2(V(1))$ and $K_2(V(1)) \leq \frac{3}{2}K_2(V(0))$, $K_2(V(2)) \leq \frac{3}{4}K_2(V(0))$.

After the first cycle, K_2 is at most three fourths its value before the cycle. Similarly, K_1 loses at least 1/4 of its value during the first cycle. A similar reduction occurs during each cycle so that after c cycles, each $K_i \leq (\frac{3}{4})^c K_0$ which tends to 0 as $c \rightarrow \infty$. It follows that the lengths of the edges of the hypercubes go to zero and that the process converges to a point \hat{z} at which all the $g_i(\cdot, \theta^i)$'s are zero.

We now proceed to the general proof in \mathbf{R}^N . Let $M \equiv 2^{N-1}$. Let $V(R) = \{v_1, \dots, v_M, w_1, \dots, w_M\}$ be a completely labeled set in \mathbf{R}^N after round R . For any edge $e_j = [e_j^+, e_j^-]$ for agent k , let $\|e_j\|_k = |L_k(e_j^+)| + |L_k(e_j^-)|$, so that $\ell_k(R) = \max_j \{\|e_1\|_k, \|e_2\|_k, \dots, \|e_M\|_k\}$. We assume that we are focusing on agent 1, that $L_j(v_j) > 0$ and $L_j(w_j) < 0$ for $j = 1, \dots, M$, and that $e_j = [v_j, w_j]$ are the M edges for agent 1.

Suppose first that the $(R+1)$ 'st round belongs to agent 2, not agent 1. Since L_1 does not change sign on any edge of agent 2, each edge of agent 2 is a $[v_j, v_k]$ or a $[w_j, w_k]$. So, after the $(R+1)$ 'st round, the completely labeled set $V(R+1)$ will be comprised of vertices of the form:

$$v_h, \dots, v_i, \frac{v_j + v_k}{2}, \dots, \frac{v_s + v_t}{2}, w_j, \dots, w_h, \frac{w_s + w_k}{2}, \frac{w_t + w_i}{2}.$$

If the $(R+2)$ 'nd round does not belong to agent 1 either, then $V(R+2)$ will be comprised of points of the form:

$$v_h, \dots, \frac{v_k + v_i}{2}, \dots, \frac{v_s}{2} + \frac{v_i + v_j}{4}, \dots, \frac{v_a + v_b + v_c + v_d}{4}, w_i, \dots, \text{etc.}$$

Similarly, if none of rounds $R+1, R+2, \dots, R+S$ belong to agent 1, then the completely labeled set $V(R+S)$ will contain only points of the form

$$(3) \quad \alpha_1^S v_1 + \dots + \alpha_M^S v_M, \dots, \beta_1^S w_1 + \dots + \beta_M^S w_M$$

where

$$(4) \quad \alpha_1^S + \dots + \alpha_M^S = \beta_1^S + \dots + \beta_M^S = 1$$

and

$$(5) \quad \text{each non-zero } \alpha_i^S \text{ or } \beta_j^S \text{ equals } \frac{k}{2^S} \text{ for some integer } k, 1 \leq k \leq 2^S.$$

For each of rounds $R+1, \dots, R+S$ as above, let

$$V(R+j) = \{v_1^j, \dots, v_M^j, w_1^j, \dots, w_M^j\},$$

where $v_i^0 = v_i$ and $w_i^0 = w_i$ and the points in $V(R+j)$ are ordered so that:

$$\begin{aligned} \text{each } L_h(v_i^j) &= L_h(v_i^0) & \text{for } h = 1, \dots, N, i = 1, \dots, M, \\ \text{each } L_h(w_i^j) &= L_h(w_i^0) & \text{for } h = 1, \dots, N, i = 1, \dots, M. \end{aligned}$$

Claim 1: Each $v_1^j = \alpha_{j1}v_1 + \dots + \alpha_{jM}v_M$ where $\alpha_{j1} > 0$ and $\alpha_{j1} + \dots + \alpha_{jM} = 1$. The second part follows directly from (4). To see that α_{j1} is non-zero, follow the course of v_1^i as i progresses. In the formation of $V(R+1)$, assume without loss of generality that $[v_1, v_2]$ is an edge of $V(R)$. Then, v_1^1 is either v_1 or $(v_1 + v_2)/2$, both of which have a non-zero coefficient of v_1 . In the formation of $V(R+2)$ from $V(R+1)$, assume that $[v_1^1, v_1^3]$ is an edge in $V(R+1)$. Then, v_1^2 equals v_1^1 or $\frac{v_1^1 + v_1^3}{2}$. Since $v_1^1 = v_1$ or $\frac{v_1 + v_2}{2}$, we see that the coefficients of v_1 in v_1^2 is either 1, 1/2 or 1/4; and so on.

Claim 2: For rounds $V(R+1), \dots, V(R+S)$ as above, none of which belong to agent 1,

$$K_1(V(R+S)) \leq \left(\frac{2^S - 1}{2^S}\right) K_1(V(R)).$$

As in the two dimensional example above, let $a_i = L_1(v_i)$ and $b_i = -L_j(w_i) \equiv M_1(w_i)$ for $i = 1, \dots, M$. Suppose without loss of generality that

$$(6) \quad a_1 + b_1 \geq a_j + b_j \text{ for all } j.$$

Recall that the completely labeled set for $V(R+S)$ is

$$(7) \quad \begin{aligned} v_1^S &= \sum_i \alpha_{1i}^S v_i, \dots, v_M^S = \sum_i \alpha_{Mi}^S v_i \\ w_1^S &= \sum_i \beta_{1i}^S w_i, \dots, w_M^S = \sum_i \beta_{Mi}^S w_i. \end{aligned}$$

and that $e_i^S \equiv [v_i^S, w_i^S]$ is an edge for agent 1. Therefore,

$$\begin{aligned} \|e_i^S\|_1 &= |L_1(v_i^S)| + |L_1(w_i^S)| \\ &= L_1(\sum \alpha_{1i}^S v_i) - L_1(\sum \beta_{1i}^S w_i) \\ &= \sum_i \alpha_{1i}^S a_i + \sum_i \beta_{1i}^S b_i \\ &\leq \sum_i \max\{\alpha_{1i}^S, \beta_{1i}^S\} (a_i + b_i) \\ &\leq \max\{\alpha_{11}^S, \beta_{11}^S\} (a_1 + b_1) \\ &\quad + \sum_{i>1} (\alpha_{1i}^S + \beta_{1i}^S) (a_i + b_i) \\ &\leq \max\{\alpha_{11}^S, \beta_{11}^S\} (a_1 + b_1) \\ &\quad + \sum_{i>1} (\alpha_{1i}^S + \beta_{1i}^S) (a_1 + b_1) \quad \text{by (6)} \\ &= K_1(V(R)) [\max\{\alpha_{11}^S, \beta_{11}^S\} + \sum_{j>1} (\alpha_{1j}^S + \beta_{1j}^S)] \\ &= K_1(V(R)) [2 - \min\{\alpha_{11}^S, \beta_{11}^S\}], \quad \text{by (4)}. \end{aligned}$$

By (5) and Claim 1,

$$\min\{\alpha_{11}^S, \beta_{11}^S\} \geq \frac{1}{2^S}.$$

Therefore, $\|e_i^S\| \leq K_1(V(R)) \cdot [2 - \frac{1}{2^S}]$. It follows that

$$K_1(V(R+S)) = \max \|e_i^S\| \leq \frac{2^{S+1} - 1}{2^S} K_1(V(R)).$$

If rounds $R+1, \dots, R+n-1$ do not belong to agent 1,

$$K_1(V(R+n-1)) \leq \frac{2^{n+1} - 1}{2^n} K_1(V(R)).$$

On the other hand, if $V(R+S)$ is as in (7), and round $R+S+1$ belongs to agent 1, then for each edge $e_i^S = [v_i^S, w_i^S]$ of agent 1, the new $e_i^{S+1} = [v_i^S, \frac{v_i^S + w_i^S}{2}]$ or $[\frac{v_i^S + w_i^S}{2}, w_i^S]$. In the former case,

$$\begin{aligned} \|e_i^{S+1}\|_1 &= L_1(v_i^S) - L_1\left(\frac{v_i^S + w_i^S}{2}\right) \\ &= \frac{1}{2}L_1(v_i^S) - \frac{1}{2}L_1(w_i^S) \\ &= \frac{1}{2}\|e_i^S\|_1. \end{aligned}$$

In the latter case,

$$\begin{aligned} \|e_i^{S+1}\|_1 &= L_1\left(\frac{v_i^S + w_i^S}{2}\right) - L_1(w_i^S) \\ &= \frac{1}{2}L_1(v_i^S) - \frac{1}{2}L_1(w_i^S) \\ &= \frac{1}{2}\|e_i^S\|_1. \end{aligned}$$

It follows that

$$K_1(V(R+S+1)) = \frac{1}{2}K_1(V(R+S)).$$

Conclusion: Consider a cycle whose last round (and no other round in the cycle) belongs to agent 1. Then, $K_1(V)$ increases by a factor $\leq \frac{2^{N+1}-1}{2^N}$ during the first $N-1$ rounds and decreases by $1/2$ during the last round. Throughout the cycle, K_1 changes by a factor $\leq \frac{2^{N+1}-1}{2^{N+1}}$. Let $V(c)$ be the completely labeled set after c rounds. We have shown that $L(V(c))$ lies in a hypercube whose edges have length $\leq \left(\frac{2^{N+1}-1}{2^{N+1}}\right)^c$. Since L is a diffeomorphism (one-to-one, onto, continuous map) and since $\left(\frac{2^{N+1}-1}{2^{N+1}}\right)^c \rightarrow 0$ as the

number of cycles c goes to infinity, $V(c)$ tends to a unique point \hat{z} . Since each L_i takes on both signs on every $V(c)$, $L(\hat{z}) = \mathbf{0}$, that is, $V(c)$ tends to the equilibrium.

Remark. The convergence of $\left(\frac{2^N-1}{2^N}\right)^c$ as $c \rightarrow \infty$ is, of course, very slow. This rate is the “worst case” rate for the sake of the above proof. On the one hand, one can describe completely labeled sets $V(0)$ so that $K_i(1) = \frac{2^N-1}{2^N}K_i(0)$, after one cycle; so our estimate is the best possible. On the other hand, for a random $V(0)$, on the average K_i remains constant during the rounds of agents other than agent i . So, mean convergence is $\left(\frac{1}{2}\right)^c$ as $c \rightarrow \infty$, a rather fast rate especially if the algorithm is performed on a parallel processor.

REFERENCES

- [1] ARROW, K.J., H.D. BLOCK AND L. HURWICZ, *On the stability of the competitive equilibrium, I*, *Econometrica* 26 (1958), pp. 522–552.
- [2] ARROW, K.J., H.D. BLOCK AND L. HURWICZ, *On the stability of the competitive equilibrium, II*, *Econometrica* 27 (1959), pp. 82–109.
- [3] BALA, V., *Rates of convergence of B processes: some simulation results*, *Journal of Computer Science in Economics and Management Science* (Forthcoming); Cornell University Working Paper, 1988.
- [4] BALA, V., M. MAJUMDAR, AND T. MITRA, *Resource allocation on non-Walrasian environments: some analytical and simulation results*, *Journal of Economic Behavior and Organization* (Forthcoming).
- [5] CONTE, S.D., AND C. DEBOOR, *Elementary Numerical Analysis*, New York: McGraw-Hill (1980).
- [6] HURWICZ, L., R. RADNER AND S. REITER, *A stochastic decentralized resource allocation process: Part I*, *Econometrica* 43 (1975), pp. 187–221.
- [7] HURWICZ, L., R. RADNER AND S. REITER, *A stochastic decentralized resource allocation process: Part II*, *Econometrica* 43 (1975), pp. 363–393.
- [8] HURWICZ, L., *Optimality and Informational Efficiency in Resource Allocation Processes*, *Mathematical Methods in the Social Sciences*, edited by Karlin S., and Patrick Supes, Stanford University Press, 1960; also in: K.J. Arrow and T. Scitovsky, eds., *Readings in Welfare Economics*, Irwin, 1969.
- [9] HURWICZ, L., *On the dimensional requirements of informationally decentralized Pareto-satisfactory processes*, *Studies in Resource Allocation Processes*, Ed. K.J. Arrow and L. Hurwicz, Cambridge University Press, 1977.
- [10] JORDAN, J.S., *The informational requirement of local stability in decentralized allocation mechanisms*, *Information, Incentives and Economic Mechanism*, essays in honor of Leonid Hurwicz, Ed. T. Groves, R. Radner, and S. Reiter (1987).
- [11] MOUNT, K.R., AND S. REITER, *The Informational Size of Message Spaces*, *Journal of Economic Theory* (1974), pp. 161–192.
- [12] MOUNT, K.R., AND S. REITER, *On the existence of a locally stable dynamic process with a statically minimal message space*, *Information, Incentives and Economic Mechanisms*, essays in honor of Leonid Hurwicz, Ed. T. Groves, R. Radner, and S. Reiter (1987).
- [13] REITER, S., *There is no adjustment process with 2-dimensional message spaces for “counter-examples”*, Northwestern University, Mimeo, 1979.
- [14] SAARI, D.G., *Iterative price mechanisms*, *Econometrica* 53 (1985), pp. 1117–1133.
- [15] SAARI, D.G., AND C.P. SIMON, *Effective price mechanisms*, *Econometrica* 46 (1978), pp. 1097–1125.
- [16] SAARI, D.G., AND S.R. WILLIAMS, *On the local convergence of economic mechanisms*, *Journal of Economic Theory* 40 (1986), pp. 152–167.
- [17] SCARF, H., *Some examples of global instability of the competitive equilibrium*, *International Economic Review* 1 (1960), pp. 157–172.
- [18] SMALE, S., *A convergent process of price adjustment and global Newton methods*, *Journal of Mathematical Economics* 3 (1976), pp. 107–120.

- [19] WILLIAMS, S., *Necessary and sufficient conditions for the existence of a locally stable message process*, Journal of Economic Theory 35 No. 1 (1983), pp. 127–154.