January 1987 LIDS-P-1644

Distributed Computation for Linear Programming Problems Satisfying Certain Diagonal Dominance Condition*

by

Paul Tseng**

Abstract

An iterative ascent method for a class of linear programming problems whose constraints satisfy certain diagonal dominance condition is proposed. This method partitions the original linear program into subprograms where each subprogram corresponds uniquely to a subset of the decision variables of the problem. At each iteration, one of the subprograms is solved by adjusting its corresponding variables, while the other variables are held constant. The algorithmic mapping underlying this method is shown to be monotone and contractive. Using the contractive property, the method is shown to converge even when implemented in an asynchronous, distributed manner, and that its rate of convergence can be estimated from the synchronization parameter.

*This research was conducted first at the M.I.T. Laboratory for Information and Decision Systems with partial support provided by the National Science Foundation under Grant NSF-ECS-8217668 and later at the U.B.C. Faculty of Commerce and Business Administration with support provided by NSERC of Canada under Grant NSERC-U0505.

**The author is presently with the Faculty of Commerce and Business Administration, University of British Columbia, Vancouver, Canada.

1. Introduction

The infinite horizon, discounted dynamic programming problem with finite state and control spaces can be shown (see Bertsekas [3]; Denardo [6]) to be equivalent to a very large linear programming problem whose constraints satisfy some diagonal dominance condition. However, the number of constraints in this linear program grows as a product of the size of the state space and the size of the control space. This number is typically very large, thus rendering conventional linear programming methods impractical for solving this problem. In this paper, a method for solving a more general case of the above linear programming problem is proposed. The advantages of this method are that (i) it exploits the diagonal dominance structure of the problem, and (ii) its computation can be distributed over many processors in parallel. In this way, even very large problems can be solved in a reasonably short time. More specifically, this method partitions the original linear program into subprograms where each subprogram is associated with a processor. At each iteration, one of the subprograms is solved by adjusting its corresponding variable(s), while the other variables are held fixed. The algorithmic mapping underlying this method is shown to be contractive and, using the contractive property, we show convergence even if the method is implemented in an asynchronous, distributed manner and furthermore we obtain rate of convergence estimate as a function of the synchronization parameter.

2. Problem Definition

Consider linear program of the following special form :

$$
(\mathcal{P}) \qquad \begin{array}{lll} \underset{x}{\text{Maximize}} & a^T x & \\ \text{subject to} & C^1 x \le d^1 & \\ & \vdots & \\ & C^K x \le d^K & \end{array}
$$

where a and d^k ,k=1,...,K, are given vectors in R^m ; C^k ,k=1,...,K, are given mxm real matrices. We make the following assumptions regarding $\mathcal{P}$:

Assumption A : a is nonnegative.

Assumption B : Each C^k (k=1,...,K) is a mxm diagonally dominant matrix whose diagonal entries are positive and whose off-diagonal entries are nonpositive.

We denote the (i,j)th entry of C^k by C_{ij}^k, the ith entry of d^k by d_i^k, the jth entry of a and x by a_j and x_j respectively. We also denote the index set $\{1,2,\ldots,m\}$ by M. For any vector x we will use $\|x\|_\infty$ to denote the sup norm of x , i.e. $\|x\|_\infty = \max_j |x_j|$ and for two vectors x and y of equal dimension we will use $x \le y$ to mean $x_j \le y_j$ for all j. Note that for a given $\mathcal{P}$ the point $(\lambda,\lambda,\ldots,\lambda)$ is feasible for $\mathcal{P}$ for all λ sufficiently negative.

We may interpret $\mathcal{P}$ physically as a production problem involving m production centers, each of which is responsible for producing an item, some fractional amount of which is used by the other production centers as resource to produce their own items, and the efficiency of resource usage as well as external resouce supply are both random variables. More precisely, let the amount of the jth item produced be denoted by x_j which carries with it a nonnegative utility of a_jx_j. There are K possible scenarios that may be realized. Under the kth scenario, the amount of the ith item that can be produced is limited by the amount of external materials available, given by d_i^k/C_{ii}^k, plus the sum of a fraction, given by C_{ij}^k/C_{ii}^k, of the jth item produced summed over all $j \ne i$. We wish to plan a production level $(x_1,x_2,\ldots,x_m)$ that maximizes the total utility $a_1x_1+\ldots+a_mx_m$ such that none of the production constraints is violated regardless of which scenario is realized in the future.

$\mathcal{P}$ is a special case of linear programming problems and can be solved using any linear programming method, such as the simplex method. However, if the size of $\mathcal{P}$ is very large, and noting that $\mathcal{P}$ is not necessarily sparse in structure, the time required to solve $\mathcal{P}$ would likely be large even for very efficient linear programming methods. As an example, if m=100 and K=100, then $\mathcal{P}$ has 100 decision variables and 10,000 inequality constraints. A special case of $\mathcal{P}$, the infinte horizon, discounted dynamic programming problem with finite state and control spaces, typically has $n,K \geq 100$ for real applications.

It is therefore important to design methods for solving $\mathcal{P}$ that can take advantage of its special structure. Such approach has been successful for other special cases of linear programs such as network flow problems and Leontief systems. In fact, the constraint matrix for $\mathcal{P}$ and its transpose are both Leontief (a matrix E is Leontief if E has exactly one positive entry per column and there exists a $x \geq 0$ such that $Ex > 0$). It is known that if E is Leontief, then there exists a x^*, called the least element, such that x^* solves the following problem

$$\begin{array}{ll} \text{Maximize} & a^T x \\ \text{subject to} & E^T x \leq d \end{array}$$

for all $a > 0$ and d such that the above problem is bounded (see [7]). In our work we only require that a be nonnegative but the existence of a least element still holds and is crucial for the method proposed here to work.

The infinite horizon, discounted dynamic programming problem with finite state and control spaces is described below. This problem frequently arises in the areas of inventory control, investment planning, and Markovian decision theory. It is traditionally solved by the successive approximation method or the policy iteration method (see [3] or [6]). However neither method has a theoretical rate of convergence as good as that of the method proposed here.

Special case of $\mathcal{P}$

The infinite horizon, discounted dynamic programming problem with finite state and control spaces is equivalent (see for example [3]) to the following special type of linear program :

$$\text{Maximize} \qquad \sum_{j=1}^{m} x_j$$

$$\text{subject to} \qquad x_i \le g(i,u) + \alpha \sum_{j=1}^{m} p_{ij}(u)x_j \ , \ \forall u \in U(i),\ i=1,\ldots,m$$

where $\alpha \in (0,1)$ is called the discount factor, $\{1,2,\ldots,m\}$ denotes the state space, $U(i)$ denotes the set of possible controls when in state i (size of $U(i)$ is finite), $p_{ij}(u)$ denotes the probability that the next state is j given that the current state is i and the control u is applied, and $g(i,u)$ is the average reward per stage when in state i and control u is applied.

We can make the identification with $\mathcal{P}$ more explicit by rewriting the above program as

$$\text{Maximize} \qquad \sum_{j=1}^{m} x_j$$

$$\text{subject to} \quad (1-\alpha p_{ii}(u))x_i - \sum_{j=1}^{m} \alpha p_{ij}(u)x_j \le g(i,u), \quad \forall u \in U(i), \ i=1,\ldots,m.$$

Then given that $\alpha \in (0,1)$, and augmenting the constraint set with duplicate constraints if necessary, we can easily verify that the above problem is a special case of $\mathcal{P}$.

3. The Sequential Relaxation Method

Consider an arbitrary nonempty subset of M, denoted S, and for each $x \in R^m$ define the following maximization subproblem associated with S and x :

$$\text{Maximize} \quad \sum_{j\in S} a_j \xi_j$$

$\mathcal{P}_S(x)$

$$\text{subject to} \quad \sum_{j\in S} C_{ij}^k \xi_j \le d_i^k - \sum_{j\notin S} C_{ij}^k x_j \ , \ k=1,\ldots,K$$

where ξ_j , $j\in S$, are the decision variables. Note that $\mathcal{P}_N(x)$ is just the original problem $\mathcal{P}$. For any nonempty S and x the problem $\mathcal{P}_S(x)$ is clearly feasible since the vector $(\lambda,\lambda,\ldots,\lambda)$ of dimension $|S|$ is a feasible solution. However using the following lemma we can show that $\mathcal{P}_S(x)$ in fact has an optimal solution.

<u>Lemma 1</u> Suppose A is a n by n diagonally dominant matrix with positive diagonal entries and nonpositive off-diagonal entries. Then the following holds :

(a) A^{-1} exists and is a nonnegative.

(b) If B is a nonnegative matrix of n rows such that $[A \ \ -B]$ has all row sums greater than zero then $A^{-1}B$ is nonnegative and has all row sums less than one.

<u>Proof</u> :

We prove (a) first. That A is invertible follows from the Gershgorin Circle Theorem (see for example [10]). To prove (a) we write A as $A = D-B$ where D is the diagonal matrix whose diagonal entries are the diagonal entries of A. Then

$$A^{-1} = (I-D^{-1}B)^{-1}D^{-1} \ . \tag{1}$$

$D^{-1}B$ has zero diagonal entries, nonnegative off-diagonal entries, and row sums each less than 1. Then by Gershgorin Circle Theorem $D^{-1}B$ has spectral radius less than 1 and from (1) it follows that

$$A^{-1} = (I+(D^{-1}B)+(D^{-1}B)^2+\cdots)D^{-1} .$$

Since D^{-1} and $D^{-1}B$ are both nonnegative it follows that A^{-1} is nonnegative.

We now prove (b). We are given that $Ae - Be' > 0$ where e and e' denote vectors of appropriate dimensions whose entries are all 1's. Multiplying both sides by A^{-1} and using (a) we obtain that $e - A^{-1}Be' > 0$, from which (b) follows. Q.E.D.

To show that $\mathcal{P}_S(x)$ has an optimal solution we note that its constraints written in vector notation has the form

$$\tilde{C}^k \xi \quad \le \quad \tilde{d}^k \quad , \quad k=1,\ldots,K$$

where ξ is the vector with components ξ_j, $j\in S$, and each $\tilde{C}^k$ by Assumption B is a $|S|\times|S|$ diagonally dominant matrix whose diagonal entries are positive and whose off-diagonal entries are nonpositive. Then Lemma 1 (a) implies that all feasible solutions ξ of $\mathcal{P}_S(x)$ must satisfy $\xi \le (\tilde{C}^k)^{-1}\tilde{d}^k$, $k=1,\ldots,K$, which together with Assumption A imply that $\mathcal{P}_S(x)$ has an optimal solution.

The following lemma shows that the optimal solution set of $\mathcal{P}_S(x)$ has certain special properties :

Lemma 2 For each nonempty subset S of M and each $x\in R^m$ the following holds :

(a) There exists an (unique) optimal solution ξ^* of $\mathcal{P}_S(x)$ such that $\xi^* \ge \xi$, $j\in S$, for all other optimal solutions ξ of $\mathcal{P}_S(x)$ (ξ^* will be called the largest optimal solution of $\mathcal{P}_S(x)$).

(b) The ξ^* of part (a) has the property that there exists a set of indices $\{k_i\}_{i\in S}$ such that

$$\Sigma_{j\in S} C_{ij}^{k_i} \xi_j^* = d_i^{k_i} - \Sigma_{j\notin S} C_{ij}^{k_i} x_j \quad , \text{ for all } i\in S ,$$

where ξ_j^* denotes the jth coordinate of ξ^*.

<u>Proof</u> :

We first prove part (a). Let Ξ denote the set of optimal solutions of $P_S(x)$. If Ξ is a singleton then (a) follows trivially. Otherwise let ξ and ξ' denote any two distinct elements of Ξ. It is straightforward to verify that $\xi"$ given by

$$\xi"_j = \max \{ \xi_j , \xi'_j \} \quad , \quad j\in S$$

is also feasible for $P_S(x)$. Since all the a_j's are nonnegative, $\xi"$ has an objective value that is greater than or equal to the objective value of either ξ or ξ'. Since Ξ is easily seen to be bounded from above part (a) follows.

We now prove part (b). Suppose that (b) does not hold. Then for some $i\in S$

$$\Sigma_{j\in S} C_{ij}^k \xi_j^* < d_i^k - \Sigma_{j\notin S} C_{ij}^k x_j \quad , \; k=1,2,\ldots,K$$

in which case ξ given by

$$\xi_j = \xi_j^* \qquad , \; j\in S\setminus\{i\}$$

$$\xi_i = \max_{k=1,2,\ldots,K} \{ d_i^k - \Sigma_{j\notin S} C_{ij}^k x_j - \Sigma_{j\in S\setminus\{i\}} C_{ij}^k \xi_j^* \}$$

is feasible for $\mathcal{P}_S(x)$, has an objective value greater than or equal to that of ξ^*, and is strictly greater than ξ^* in the ith entry. This contradicts the definition of ξ^*. Q.E.D.

For each nonempty subset S of M and each $x \in R^m$ we define the mapping $T_S(x): R^m \to R^{|S|}$ by

$$T_S(x) \quad = \quad \text{largest optimal solution of } \mathcal{P}_S(x).$$

That $T_S(x)$ is well defined follows from Lemma 2 (a). Now consider an arbitrary partitioning of the index set M into a collection $\mathcal{C}$ of disjoint subsets. Define the mapping $T_{\mathcal{C}}: R^m \to R^m$ by

$$T_{\mathcal{C}}(x) = (\ldots \; T_S(x) \ldots)_{S \in \mathcal{C}}.$$

The sequential version of the proposed method can be described by an initial estimate x^0 and an infinite sequence of collections $\{\mathcal{C}^0, \mathcal{C}^1, \ldots\}$. The solution sequence $\{x^t\}$ thus generated is given by

$$x^t \quad = \quad T_{\mathcal{C}^{t-1}}(T_{\mathcal{C}^{t-2}}(\ldots(T_{\mathcal{C}^0}(x^0))\ldots)) \qquad , \; t=1,2,\ldots$$

We will show in the next section that this method converges, in the sense that as $t \to \infty$, x^t approaches an optimal solution of $\mathcal{P}$ for any starting point x^0 and any sequence $\{\mathcal{C}^0, \mathcal{C}^1, \ldots\}$. In the special case where $\mathcal{C} = \{\{1\},\{2\},\ldots,\{m\}\}$ $T_{\mathcal{C}}$ is the algorithmic mapping associated with the single coordinate relaxation (Gauss-Seidel) method for solving $\mathcal{P}$. By considering other possible $\mathcal{C}$ the proposed method may be viewed as a generalized <u>relaxation</u> method that allows several coordinates to be relaxed simultaneously.

4. Convergence analysis

To prove convergence of the sequential relaxation method we will first show that for any collection $\mathcal{C}$ the mapping $T_{\mathcal{C}}$ is contractive and thus $T_{\mathcal{C}}$ possesses an unique fixed point. We will next show that this fixed point is a solution of $\mathcal{P}$ and is independent of the choice of $\mathcal{C}$. In addition we obtain rate of convergence estimate as a function of the size of the subsets in $\mathcal{C}$.

For each nonempty subset S of M and each mapping $\sigma:S\rightarrow\{1,2,\ldots,K\}$ we define the following matrices

$$A(S,\sigma) = \left[\ c_{ij}^{k}\ \right]_{k=\sigma(i),\, i\in S,\, j\in S}$$

$$B(S,\sigma) = \left[\ -c_{ij}^{k}\ \right]_{k=\sigma(i),\, i\in S,\, j\notin S}\ .$$

We note that $A(S,\sigma)$ is diagonally dominant with positive diagonal entries and nonpositive off-diagonal entries. Furthermore by the definition of T_S and Lemma 2 we can express $T_S(x)$ as

$$T_S(x) \quad = \quad d(S,\sigma) \quad + \quad A(S,\sigma)^{-1}B(S,\sigma)x_{M\setminus S}$$

where $d(S,\sigma)$ denotes $(\cdots d_i^{\sigma(i)}\cdots)_{i\in S}$, $x_{M\setminus S}$ denotes $(\cdots x_j\cdots)_{j\in M\setminus S}$ and σ is some mapping from S to $\{1,2,\ldots,K\}$. We have the following lemma :

<u>Lemma 3</u> For any arbitrary subset S of M and two arbitrary vectors x and y in R^m, we have, for some $\sigma:S\rightarrow\{1,2,\ldots,K\}$ and $\sigma':S\rightarrow\{1,2,\ldots,K\}$, that

$$A(S,\sigma)^{-1}B(S,\sigma)(w-z) \le T_S(x) - T_S(y) \le A(S,\sigma')^{-1}B(S,\sigma')(w-z)$$

where $w=x_{M\setminus S}$ and $z=y_{M\setminus S}$.

<u>Proof</u> :

Let $\xi = T_S(x)$ and $\psi = T_S(y)$. By Lemma 2 (b) we have

$$\Sigma_{j\in S}\, C_{ij}^{k_i}\, \xi_j = d_i^{k_i} - \Sigma_{j\notin S}\, C_{ij}^{k_i}\, x_j \ ,\ \forall\ i\in S \qquad (2)$$

for some set of scalars $\{k_i\}_{i\in S}$, and similarly

$$\Sigma_{j\in S}\, C_{ij}^{k_i'}\, \psi_j = d_i^{k_i'} - \Sigma_{j\notin S}\, C_{ij}^{k_i'}\, y_j \ ,\ \forall\ i\in S \qquad (3)$$

for some set of scalars $\{k_i'\}_{i\in S}$. Since ξ is feasible for $\mathcal{F}_S(x)$ and ψ is feasible for $\mathcal{F}_S(y)$ it follows that

$$\Sigma_{j\in S}\, C_{ij}^{k_i'}\, \xi_j \le d_i^{k_i'} - \Sigma_{j\notin S}\, C_{ij}^{k_i'}\, x_j \ ,\ \forall\ i\in S \qquad (4)$$

$$\Sigma_{j\in S}\, C_{ij}^{k_i}\, \psi_j \le d_i^{k_i} - \Sigma_{j\notin S}\, C_{ij}^{k_i}\, y_j \ ,\ \forall\ i\in S\ . \qquad (5)$$

By defining $\sigma(i)=k_i$ and $\sigma'(i)=k_i'$ for all $i\in S$, we can rewrite (2), (3) as

$$A(S,\sigma)\xi = d + B(S,\sigma)w \qquad (6)$$

$$A(S,\sigma')\psi = d' + B(S,\sigma')z \qquad (7)$$

and (4), (5) as

$$A(S,\sigma')\xi \le d' + B(S,\sigma')w \qquad (8)$$

$$A(S,\sigma)\psi \quad \le \quad d \;+\; B(S,\sigma)z \;, \tag{9}$$

where $d=(\ldots d_i^{k_i}\ldots)_{i\in S}$, $d'=(\ldots d_i^{k_i'}\ldots)_{i\in S}$. Equations (6),(9) together with Lemma 1 (a) imply that

$$\xi-\psi \quad \ge \quad A(S,\sigma)^{-1}B(S,\sigma)(w-z)$$

while (7),(8) together with Lemma 1 (a) imply that

$$\xi-\psi \quad \le \quad A(S,\sigma')^{-1}B(S,\sigma')(w-z) \;.$$

Q.E.D.

Let e denote the vector whose entries are all 1's and u^i ($i\in M$) the vector whose ith coordinate is 1 and the other coordinates are 0's (the dimension of e and u^i will be clear from the context). For each nonempty subset S of M we define

$$\beta_S \;=\; \max_{\text{all } \sigma} \; \max_{i\in S} \; \left\{ (u^i)^T A(S,\sigma)^{-1}B(S,\sigma)e \right\} \;.$$

It immediately follows from Lemma 3 that

$$\|T_S(x)-T_S(y)\|_\infty \quad \le \quad \beta_S\|x-y\|_\infty \quad \text{, for all } x \text{ and } y. \tag{10}$$

From Lemma 1 (b) it is easily seen that $0 \le \beta_S < 1$ so that T_S is in fact a contraction mapping. Let

$$\beta \;=\; \max_{i\in M} \beta_{\{i\}} \;=\; \max\{\, -(\Sigma_{j\ne i}C_{ij}^k)/C_{ii}^k \;\big|\; k=1,2,\ldots,K \;,\; i\in M \,\}.$$

We also define

$$\alpha_S = \min_{\text{all } \sigma} \min_{i \in S} \left\{ (u^i)^T A(S,\sigma)^{-1} B(S,\sigma) e \right\}$$

and correspondingly

$$\alpha = \min_{i \in M} \alpha_{M \setminus \{i\}} .$$

Lemma 1 (b) implies that $0 \le \alpha, \alpha_S, \beta_S, \beta < 1$. However we can show the following stronger result :

<u>Proposition 4</u> For any two disjoint nonempty subsets S and T of M we have

$$\beta_{S \cup T} \le \max\{\beta_S, \beta_T\} \tag{11}$$

and

$$\alpha_{S \cup T} \le \min\{\alpha_S, \alpha_T\} . \tag{12}$$

<u>Proof</u> :

Consider an arbitrary σ . We can write $A(S \cup T,\sigma)$ and $B(S \cup T,\sigma)$ as

$$A(S \cup T,\sigma) = \begin{bmatrix} A & -B \\ -C & D \end{bmatrix} , \quad B(S \cup T,\sigma) = \begin{bmatrix} E \\ F \end{bmatrix} \tag{13}$$

where $A = A(S,\sigma)$, $D = A(T,\sigma)$, and B, C, E, and F are nonnegative matrices of appropriate dimensions such that $B(S,\sigma) = [B \;\; E]$ and $B(T,\sigma) = [C \;\; F]$. We will show that

$$A(S \cup T,\sigma)^{-1} B(S \cup T,\sigma) e'' \le \begin{bmatrix} A^{-1}(Be' + Ee'') \\ D^{-1}(Ce + Fe'') \end{bmatrix} \tag{14}$$

where e , e', and e" are vectors whose entries are all 1's, with dimensions of $|S|$, $|T|$, and $m-|S|-|T|$ respectively. Since the choice of σ was arbitrary and the inequality in (14) holds coordinate-wise (11) and (12) will follow almost immediately. To prove (14) we first express $A(S\cup T,\sigma)^{-1}$ in the following form [cf. (13)]

$$A(S\cup T,\sigma)^{-1} = \begin{bmatrix} (A-BD^{-1}C)^{-1} & (A-BD^{-1}C)^{-1}BD^{-1} \\ D^{-1}C(A-BD^{-1}C)^{-1} & D^{-1}[I+C(A-BD^{-1}C)^{-1}BD^{-1}] \end{bmatrix}. \tag{15}$$

It is straightforward to verify that (15) is valid. Direct multiplication using (15) yields

$$A(S\cup T,\sigma)^{-1}\begin{bmatrix} E \\ F \end{bmatrix}e'' = \begin{bmatrix} (A-BD^{-1}C)^{-1}(Ee'' + BD^{-1}Fe'') \\ D^{-1}C(A-BD^{-1}C)^{-1}Ee'' + D^{-1}[I+C(A-BD^{-1}C)^{-1}BD^{-1}]Fe'' \end{bmatrix}.$$

We will now show that

$$(A-BD^{-1}C)^{-1}(Ee''+BD^{-1}Fe'') \le A^{-1}(Be'+Ee'') . \tag{16}$$

To prove (16) we consider the difference

$$\begin{aligned} &(Ee''+BD^{-1}Fe'') - (A-BD^{-1}C)A^{-1}(Be'+Ee'') \\ &= BD^{-1}Fe'' - Be' + BD^{-1}CA^{-1}(Be'+Ee'') \\ &\le BD^{-1}Fe'' - Be' + BD^{-1}Ce = -B(e'-D^{-1}(Fe''+Ce)) \\ &\le 0 . \end{aligned} \tag{17}$$

The last inequality follows from the fact that $D^{-1}(Fe''+Ce)$ is a nonnegative vector whose entries are strictly less than one [cf. Lemma 1 (b)] and B is a nonnegative matrix.

By Lemma 1 (b), both $A^{-1}B$ and $D^{-1}C$ are nonnegative and have their row sums less than one so that $I-(A^{-1}B)(D^{-1}C)$ is diagonally dominant with positive diagonal entries and nonpositive off-diagonal entries. Since $(A-BD^{-1}C)^{-1} = A^{-1}[I-(A^{-1}B)(D^{-1}C)]^{-1}$ it follows from Lemma 1 (a) that $(A-BD^{-1}C)^{-1}$ is nonnegative and therefore [cf. (17)]

$$(A-BD^{-1}C)^{-1}\left[(Ee''+BD^{-1}Fe'') - (A-BD^{-1}C)A^{-1}(Be'+Ee'')\right] \le 0 .$$

This proves (16).

To complete the proof of (14) we express $A(S\cup T,\sigma)^{-1}$ in a form analogous to (15) [cf. (13)]

$$A(S\cup T,\sigma)^{-1} = \begin{bmatrix} A^{-1}[I+B(D-CA^{-1}B)^{-1}CA^{-1}] & A^{-1}B(D-CA^{-1}B)^{-1} \\ (D-CA^{-1}B)^{-1}CA^{-1} & (D-CA^{-1}B)^{-1} \end{bmatrix} . \qquad (18)$$

Direct multiplication using (18) yields

$$A(S\cup T,\sigma)^{-1}\begin{bmatrix} E \\ F \end{bmatrix}e'' = \begin{bmatrix} A^{-1}[I+C(D-CA^{-1}B)^{-1}CA^{-1}]Ee''+A^{-1}B(D-CA^{-1}B)^{-1}Fe'' \\ (D-CA^{-1}B)^{-1}(CA^{-1}Ee'' + Fe'') \end{bmatrix} .$$

By an argument analogous to that used in the proof of (16) we obtain that

$$(D-CA^{-1}B)^{-1}(CA^{-1}Ee''+Fe'') \le D^{-1}(Ce+Fe'')$$

which together with (16) proves (14).

To prove (11) and (12) we note that (14) implies

$$[(u^i)^T \;\; 0\,]A(S\cup T,\sigma)^{-1}B(S\cup T,\sigma)e'' \le (u^i)^T A(S,\sigma)^{-1}B(S,\sigma)\begin{bmatrix} e' \\ e'' \end{bmatrix}$$

$$[\,0 \;\; (u^j)^T]A(S\cup T,\sigma)^{-1}B(S\cup T,\sigma)e'' \le (u^j)^T A(T,\sigma)^{-1}B(T,\sigma)\begin{bmatrix} e \\ e'' \end{bmatrix}$$

for all $i\in S$, $j\in T$ and all σ. Taking the maximum on both sides over

all $i \in S$, $j \in T$ and all σ we obtain (11). Similarly taking the minimum on both sides over all $i \in S$, $j \in T$ and all σ we obtain (12).
Q.E.D.

<u>Corollary 4</u> For any nonempty strict subset S of M we have

$$\alpha \quad \leq \quad \alpha_S \quad \leq \quad \beta_S \quad \leq \quad \beta \; .$$

The result of Proposition 4 can be sharpened by using Corollary 4 :

<u>Proposition 5</u> For any two disjoint nonempty subset S and T of M we have

$$\beta_{S \cup T} \quad \leq \quad \max\{\beta_S, \beta_T\} \; - \; \alpha(S,T)(1-\beta)/(1+\beta^2) \qquad (19)$$

and

$$\alpha_{S \cup T} \quad \leq \quad \min\{\alpha_S, \alpha_T\} \; - \; \alpha(S,T)(1-\beta)/(1+\beta^2) \; , \qquad (20)$$

where we define the nonnegative scalar

$$\alpha(S,T) \; = \; \min_\sigma \; \{ \; A(S,\sigma)^{-1}\left[-C_{ij}^{\sigma(i)} \right]_{i\in S, j\in T} \, ,$$
$$A(T,\sigma)^{-1}\left[-C_{ij}^{\sigma(i)} \right]_{i\in T, j\in S} \; \} \; .$$

<u>Proof</u> :

Consider an arbitrary σ . We can write $A(S \cup T,\sigma)$ and $B(S\cup T,\sigma)$ as [cf. (13)]

$$A(S \cup T,\sigma) \; = \; \begin{bmatrix} A & -B \\ -C & D \end{bmatrix} \; , \quad B(S \cup T,\sigma) \; = \; \begin{bmatrix} E \\ F \end{bmatrix}$$

where $A = A(S,\sigma)$, $D = A(T,\sigma)$, and B, C, E, and F are nonnegative matrices of appropriate dimensions such that $B(S,\sigma) = [B \quad E]$ and $B(T,\sigma) = [C \quad F]$. We will show that

$$A(S\cup T,\sigma)^{-1}B(S\cup T,\sigma)e'' \leq \begin{bmatrix} A^{-1}(Be'+Ee'') \\ D^{-1}(Ce+Fe'') \end{bmatrix} - \frac{\alpha(S,T)(1-\beta)}{1+\beta^2} \begin{bmatrix} e \\ e' \end{bmatrix} \qquad (21)$$

where e , e', and e" are all vectors whose entries are 1's, with dimensions of $|S|$, $|T|$, and $m-|S|-|T|$ respectively. Since the choice of σ was arbitrary and the inequality in (21) holds coordinate-wise (19) and (20) then follows. To prove (21) we first express $A(S\cup T,\sigma)^{-1}B(S\cup T,\sigma)e''$ in the form given by (13) and (15) :

$$A(S\cup T,\sigma)^{-1}\begin{bmatrix} E \\ F \end{bmatrix}e'' = \begin{bmatrix} (A-BD^{-1}C)^{-1}(Ee''+BD^{-1}Fe'') \\ D^{-1}C(A-BD^{-1}C)^{-1}Ee''+D^{-1}[I+C(A-BD^{-1}C)^{-1}BD^{-1}]Fe'' \end{bmatrix}.$$

Then (21) will be partially proven if we can show

$$(A-BD^{-1}C)^{-1}(Ee''+BD^{-1}Fe'') - A^{-1}(Be'+Ee'') \leq -\alpha(S,T)(1-\beta)/(1+\beta^2)e. \qquad (22)$$

To prove (22) we first bound the difference

$$\begin{aligned} &(Ee''+BD^{-1}Fe'') - (A-BD^{-1}C)A^{-1}(Be'+Ee'') \\ &= BD^{-1}Fe'' - Be' + BD^{-1}CA^{-1}(Be'+Ee'') \\ &\leq BD^{-1}Fe'' - Be' + BD^{-1}Ce = -B(e'-D^{-1}(Fe''+Ce)) \\ &\leq -Be'(1-\beta) . \end{aligned} \qquad (23)$$

where the last inequality follows from Corollary 4. Now consider $(A-BD^{-1}C)^{-1} = A^{-1}[I-(A^{-1}B)(D^{-1}C)]^{-1}$. By Lemma 1 (b) and Corollary 4 both $A^{-1}B$ and $D^{-1}C$ are nonnegative matrices whose row sums are all less than or equal to β. It then follows that their product is a nonnegative matrix whose row sums are all less than or equal to β^2 and therefore the eigenvalues of $I-(A^{-1}B)(D^{-1}C)$ are (by Gershgorin Circle Theorem) in the interval $[1-\beta^2, 1+\beta^2]$. Furthermore by Lemma 1 (a) both A^{-1} and $[I-(A^{-1}B)(D^{-1}C)]^{-1}$ are nonnegative matrices so that (23) implies that the left side quantity of (22) is

$$\le - A^{-1}[I-(A^{-1}B)(D^{-1}C)]^{-1}Be'(1-\beta)$$

$$\le - A^{-1}Be' \ (1-\beta)/(1+\beta^2)$$

$$\le - e\ \alpha(S,T)(1-\beta)/(1+\beta^2)\ .$$

where the last inequality follows from the definition of $\alpha(S,T)$. This proves (22).

To complete the proof of (21) we express $A(S\cup T,\sigma)^{-1}B(S\cup T,\sigma)e"$ in the alternate form given by (13) and (18) :

$$A(S\cup T,\sigma)^{-1}\begin{bmatrix} E \\ F \end{bmatrix}e" = \begin{bmatrix} A^{-1}[I+C(D-CA^{-1}B)^{-1}CA^{-1}]Ee"+A^{-1}B(D-CA^{-1}B)^{-1}Fe" \\ (D-CA^{-1}B)^{-1}(CA^{-1}Ee" + Fe") \end{bmatrix}.$$

By an argument analogous to that used in the proof of (22) we obtain that

$$(D-CA^{-1}B)^{-1}(CA^{-1}Ee"+Fe") - D^{-1}(Ce+Fe") \le -\alpha(S,T)(1-\beta)/(1+\beta^2)e'$$

which together with (22) prove (21). Q.E.D.

The coefficient $\alpha(S,T)$ in some sense estimates the amount of interaction between those variables with index in S and those variables with index in T as imposed by the problem constraints (if $\alpha(S,T) > 0$ then an interaction surely exists). Unfortunately $\alpha(S,T)$ is difficult to compute in general. Using Proposition 5 we can show that, for any collection $\mathbf{C}$, $T_{\mathbf{C}}$ is a contraction mapping :

Proposition 6 There exists a set of scalars $\{\beta_{\mathfrak{C}}\}_{\text{all } \mathfrak{C}}$, each between 0 and 1, such that

$$\| T_{\mathfrak{C}}(x) - T_{\mathfrak{C}}(y) \|_\infty \quad \leq \quad \beta_{\mathfrak{C}} \| x - y \|_\infty \qquad \text{, for all } x \text{ and } y. \tag{24}$$

where $\{\beta_{\mathfrak{C}}\}$ satisfies

$$\beta_{\{1\},\dots,\{m\}} \quad = \quad \beta \ . \tag{25}$$

Furthermore if $\mathfrak{C}$ and $\mathfrak{C}'$ are two partitions such that each element of $\mathfrak{C}$ is strictly contained in an element of $\mathfrak{C}'$, then

$$\beta_{\mathfrak{C}'} \quad \leq \quad \beta_{\mathfrak{C}} \ - \ \gamma(\mathfrak{C},\mathfrak{C}')(1-\beta)/(1+\beta^2) \ . \tag{26}$$

where we define

$$\gamma(\mathfrak{C},\mathfrak{C}') = \min \{ \ \alpha(S,T) \ | \ S \in \mathfrak{C} \ , \ T \in \mathfrak{C} \ \text{ and } S \cup T \in \mathfrak{C}' \ \} \ .$$

Proof :

The proof is by construction (of the set of scalars $\{\beta_{\mathfrak{C}}\}$) since $\beta_{\mathfrak{C}}$ given by

$$\beta_{\mathfrak{C}} \quad = \quad \max \{ \ \beta_S \ | \ S \in \mathfrak{C} \ \}$$

by Proposition 5 and the definition of $\gamma(\mathfrak{C},\mathfrak{C}')$ satisfies (24),(25), and (26). Q.E.D.

The contractiveness of $T_{\mathfrak{C}}$ implies that $T_{\mathfrak{C}}$ has an unique fixed point (see for example [8];[10]). The following proposition shows that this fixed point belongs to the set of optimal solutions of $\mathcal{P}$.

<u>Proposition 7</u> For any collection $\mathcal{C}$ of disjoint subsets of M whose union is M, x^* is a fixed point of $T_{\mathcal{C}}$ (i.e. $x^* = T_{\mathcal{C}}(x^*)$) if and only if x^* is the largest optimal solution of $\mathcal{P}$.

<u>Proof</u> :

Suppose that $x^* = T_{\mathcal{C}}(x^*)$, then from the definition of $T_{\mathcal{C}}$ and Lemma 2 we have

$$\Sigma_{j\in S}\, C_{ij}^{k}\, x_j^* \;\le\; d_i^{k} - \Sigma_{j\notin S}\, C_{ij}^{k}\, x_j^*\ ,\ k=1,\ldots,K,\ \forall i\in S,\ \forall S\in\mathcal{C}$$

and

$$\Sigma_{j\in S}\, C_{ij}^{k_i}\, x_j^* \;=\; d_i^{k_i} - \Sigma_{j\notin S}\, C_{ij}^{k_i}\, x_j^*\ ,\ \forall i\in S,\ \forall S\in\mathcal{C}$$

for some set of indices $\{k_i\}$. Clearly x^* is feasible for $\mathcal{P}$. To show that x^* is an optimal solution we assume that the constraints of $\mathcal{P}$ have been ordered such that $\mathcal{P}$ has the form :

$$(\mathcal{P}') \qquad \begin{array}{ll} \text{Maximize} & a^T x \\ \text{subject to} & Cx \le c \\ & Dx \le d \end{array}$$

where $C = [\ C_{ij}^{k_i}\ ]_{i\in M, j\in M}$ and $c = [\ d_i^{k_i}\ ]_{i\in M}$. The dual problem of $\mathcal{P}'$ is :

$$(\mathcal{D}') \qquad \begin{array}{ll} \text{Minimize} & c^T u + d^T v \\ \text{subject to} & C^T u + D^T v = a \\ & u \ge 0\ ,\ v \ge 0 \end{array}$$

Let $u^* = (C^T)^{-1}a$, $v^* = 0$. Since C is diagonally dominant with positive diagonal entries and nonpositive off-diagonal entries, by Lemma 1 (a) (and the nonnegativity of a) u^* is nonnegative.

Therefore (u^*, v^*) is feasible for $\mathcal{D}'$. Furthermore, since $Cx^*=c$, we have

$$(u^*)^T(Cx^*-c) = 0$$
$$(v^*)^T(Dx^*-d) = 0$$

and thus the complementary slackness condition is satisfied. It follows from classical duality theory that x^* is an optimal solution of $\mathcal{P}'$. To show that x^* is the largest optimal solution of $\mathcal{P}'$ we note that any optimal solution x' of $\mathcal{P}'$ necessarily satisfies $Cx' \le c$, or equivalently [cf. Lemma 1 (a)] $x' \le C^{-1}c$. Since $x^* = C^{-1}c$ then x^* must be the largest optimal solution of $\mathcal{P}'$. Q.E.D.

In what follows we will use x^* to denote the largest optimal solution of $\mathcal{P}$. Combining Proposition 6 with 7 we obtain our main convergence result :

<u>Proposition 8</u> For any arbitrary sequence $\{\mathcal{C}^0, \mathcal{C}^1, \ldots\}$ and starting point x^0 we have

$$\lim_{t\to\infty} x^t = x^* \qquad \text{and} \qquad \|x^t - x^*\|_\infty \le \mu \|x^0 - x^*\|_\infty ,$$

where x^t is given by

$$x^t = T_{\mathcal{C}^{t-1}}(T_{\mathcal{C}^{t-2}}(\ldots(T_{\mathcal{C}^0}(x^0))\ldots)) \quad , t=1,2,\ldots$$

and $$\mu = \max_{t=0,1,\ldots} \beta_{\mathcal{C}^t} . \tag{27}$$

The diagonal dominance of the constraint matrices C^k's is necessary for the mapping $T_{\mathcal{C}}$ to be contractive. One can easily construct examples for which the diagonal dominance assumption is violated and for which the mapping $T_{\mathcal{C}}$ is not contractive. Note that the classical Gauss-Seidel method (see [10]) for solving a system of linear equalities $Ex=b$ is very similar in nature to the special case of the proposed method with $\mathcal{C}^t = \{\{1\},\{2\},\ldots,\{m\}\}$ for all t. The Gauss-Seidel method also requires the diagonal dominance assumption on the matrix E to ensure convergence. Furthermore, at each iteration, it adjusts one of the coordinate variables, say x_i, to satisfy the ith equality constraint (while the other x_j's , $j \neq i$, are held fixed), at the expense of violating other equality constraints. The relaxation method proposed here does much the same, except that each equality constraint is replaced by a set of inequality constraints and that several coordinates may be relaxed simultaneously. Drawing upon this analogy we see that the concept of relaxing several coordinates simultaneously and the associated convergence theory [cf. Proposition 6] are equally applicable to solving a system of linear equalities.

Equations (25) and (26) suggests that if groups of coordinates are relaxed simultaneously then the rate of convergence of the proposed method, as estimated by $\beta_{\mathcal{C}}$ for some partition $\mathcal{C}$, can only improve. This improvement is likely to be strict if the coordinates in each group are in some sense strongly coupled (i.e. $\gamma(\mathcal{C}',\mathcal{C}) > 0$ where $\mathcal{C}'$ denotes the partition $\{\{1\},\{2\},\ldots,\{m\}\}$).

The mapping $T_{\mathcal{C}}$ apart from being contractive has the additional property of being monotone (i.e. if $y \le x$ then $T_{\mathcal{C}}(y) \le T_{\mathcal{C}}(x)$). This is not hard to see using equations (6), (9) and the fact that $A(S,\sigma)^{-1}$ and $B(S,\sigma)$ are both nonnegative matrices for all S and σ. The monotonicity property is often useful for proving convergence of algorithms (see for example [3],[4]) although in our case the contractiveness of $T_{\mathcal{C}}$ is alone sufficient for establishing all the convergence results needed.

In the special case where the cost vector a has positive entries it is easily verified that the set of optimal solutions of $\mathcal{P}$ is a singleton. As a final remark, all our results still hold if the linear cost $a^T x$ is replaced by

$$\sum_j a_j(x_j)$$

where each $a_j : R \to R$ is a subdifferentiable function with nonnegative slopes.

5. Asynchronous distributed implementation

In this section, we consider the asynchronous, distributed implementation of the sequential relaxation method described in Section 3 and show that the rate of convergence for this implementation can be estimated as a function of the synchronization parameter.

Distributed implementation is of interest because the rapid increase in the speed and the computing power of processors has made distributing the computational load over many processors in parallel very attractive. In the conventional scenario for distributed implementation, the computational load is divided among several processors during each iteration; and, at the end of each iteration, the processors are assumed to exchange all necessary information regarding the outcome of the current iteration. Such an implementation where a round of information exchange, involving all processors, occurs at the end of each iteration is called synchronous. However, for many applications in the areas of power systems, manufacturing, and data communication, synchronization is impractical. Furthermore, in such a synchronous environment, the faster processors must always wait for the slower ones. Asynchronous, distributed implementation permits the processors to compute and exchange (local) information essentially independent of each other. A minimum amount of coordination among the processors is required, thus alleviating the need for initialization and synchronization protocols.

A study of asynchronous, distributed implementation is given in [1]. An example of asynchronous, distributed implementation on a "real" system is the ARPANET (see for example [9]) data communication network, where nodes and arcs on the network can fail withoug warning. However, convergence analysis in such a chaotic setting is typically difficult and restricted to simple problems. The recent work of Bertsekas [4] on distributed computation of fixed points and of Tsitsiklis [11] show that convergence is provable for a broad class of problems, among which is the problem of computing the fixed point of a contractive (with respect to sup norm) mapping.

The model for asynchronous, distributed implementation considered here is similar to that considered in [4]. In [4], convergence is shown under the assumption that the time between successive computations at each processor and the communication time between each pair of processors are finite. here we further assume that this time is bounded by some constant. Using this boundedness assumption, we estimate the rate of convergence of the distributed relaxation method as a function of the bounding constant. This rate of convergence result is similar to that given by Baudet [2] and it holds for the fixed point computation of any contractive (with respect to the sup norm) mapping. The argument used here however is still interesting in that it is a simpler and more intuitive than that given in [2].

Description of the implementation

For simplicity we will assume that the same collection $\mathbf{C}$ is used throughout the method (i.e. $\mathbf{C} = \mathbf{C}^0, \mathbf{C}^1, \ldots$) and denote the subsets of nodes belonging to $\mathbf{C}$ by $S_1, S_2, \ldots, S_R$. Now we consider finding the fixed point of $T_{\mathbf{C}}$ by distributing the computation over R processors, where the communication and the computation done by the processors are not coordinated.

Let processor r, denoted by P_r, be responsible for updating the value of the coordinates in S_r. In other words, P_r takes the current value of x it possess, applies the mapping T_{S_r} to x, and then sends the coordinates of $T_{S_r}(x)$ to the other processors. Each P_r upon receiving a value, say that of coordinate j, from some P_q ($j \in S_q$), $q \neq r$, replaces its value of x_j by the received value. We assume that P_r does not apply T_r unless a new value is received since P_r had last computed. In what follows, we will count each application of T_r by some P_r as a computation.

Let the communication time between any pair of processors be upper bounded by L_1, where L_1 is in units of "consecutive computations". In other words, at most L_1 consecutive computations can pass before a value sent by P_r to P_q is received by P_q, for all r, q such that $r \neq q$. We also assume that each P_r always uses the most recently received values in its computations (note that due to communication delay P_r may not receive values from P_q by the order in which they were sent).

Let L_2 denote the upper bound on the number of consecutive computations that can pass before each P_r has made at least a computation.

The assumption that both L_1 and L_2 are finite is reasonable for any useful system; for otherwise the system may either wait arbitrarily long time to hear from a processor, or leave some processor out of the computation altogether. Let $L = L_1 + L_2$. Then we have that every processor always computes using values all of which were computed within the last L computations.

Convergence of the relaxation method under distributed implementation

The following proposition is the main result in this section.

Proposition 9 The iterates generated by the asynchronous, distributed version of the relaxation method converge to the fixed point of T_C at a geometric rate, with rate of convergence bounded by $(\beta_C)^{1/L}$.

Proof :

The idea of the proof is quite simple, although the notation may become a little unwieldy. Define

I^t $(t=1,2,\ldots)$ = Index of the processor performing the t-th computation.

Ω^t $(t=1,2,\ldots)$ = S_{I^t}

x_j^t $(j\in M; t=1,2,\ldots)$ = P_r's value of the jth coordinate immediately following the t-th computation, where j belongs to S_r.

α_j^t $(j\notin \Omega^t; t=1,2,\ldots)$ = The number such that, when processor I^t performs the t-th computation (thus generating $x_j^t, j\in\Omega^t$), the x_j $(j\notin\Omega^t)$ value used is generated by the α_j^t-th computation. In other words,

$$(\ldots x_j^t \ldots)_{j\in\Omega^t} = T_{\Omega^t}(\ldots x_j^{\alpha_j^t} \ldots)$$

and

$$t - L \le \alpha_j^t < t \quad , \quad \forall\, j\notin\Omega^t .$$

Using Proposition 6 we obtain (recall that $x^* = T_C(x^*)$) that

$$| \, x_j^t - x_j^* \, | \quad \le \quad \beta_C \max_{j\notin\Omega^t} | \, x_j^{\alpha_j^t} - x_j^* \, | \quad , \quad \forall\, j\in\Omega^t \, . \quad (28)$$

Since (using the definition of α_j^t and Ω^t)

$$j\in\Omega^{\alpha_j^t} \quad \forall\, j\in M \quad , \quad t=1,2,\ldots$$

and $\alpha_j^t = \alpha_k^t$, t=1,2,... , for all j and k belonging to the same element of $\mathcal{C}$,

we can apply (28) recursively to the righthand side of (28) to obtain :

$$
\begin{aligned}
| x_j^t - x_j^* | &\le \beta_{\mathcal{C}} \max_{j \notin \Omega^t} | x_j^{\alpha_j^t} - x_j^* | \\
&\le (\beta_{\mathcal{C}})^2 \max_{k \notin \Omega^{\alpha_j^t},\ j \notin \Omega^t} | x_k^{\alpha_k^{\alpha_j^t}} - x_k^* | \\
&\ \ \vdots \\
&\le (\beta_{\mathcal{C}})^Y \max_{n \in M} | x_n^0 - x_n^* |
\end{aligned}
$$

where Y is some positive integer, and x_n^0 denotes P_r's initial estimate of x_n^* for all $n \in S_r$. Then using the fact that

$$
\begin{aligned}
t - \alpha_j^t &\le L \qquad \forall\ j \notin \Omega^t , \\
\alpha_j^t - \alpha_k^{\alpha_j^t} &\le L \qquad \forall\ k \notin \Omega^{\alpha_j^t},\ \forall\ j \notin \Omega^t \\
\alpha_k^{\alpha_j^t} - \alpha_l^{\alpha_k^{\alpha_j^t}} &\le L \qquad \forall\ l \notin \Omega^{\alpha_k^{\alpha_j^t}},\ \forall\ k \notin \Omega^{\alpha_j^t},\ \forall\ j \notin \Omega^t \\
&\ \ \vdots
\end{aligned}
$$

we obtain (upon summing the above set of inequalities)

$$t \le Y \cdot L .$$

It follows that

$$(\beta_{\mathcal{C}})^Y \le (\beta_{\mathcal{C}})^{t/L}$$

and therefore

$$|x_j^t - x_j^*| \quad \leq \quad (\beta_{\mathcal{C}}^{1/L})^t \max_{j \in M} |x_j^0 - x_j^*| \quad , \ \forall \ j \in \Omega^t \ .$$

Q.E.D.

The scalar L is a measure of the level of synchronization in the system : the worse the synchronization, the larger the L. An example of near-perfect synchronization is when the processors compute in a cyclical order (round robin) under zero communication delay. For the special case where $\mathcal{C} = \{\{1\},\{2\},\ldots,\{m\}\}$ and the order of computation being $1,2,\ldots m$, we can verify that

$$\alpha_j^t \quad = \quad \begin{cases} t-(i-j) & \text{if } i > j \\ t-(m+i-j) & \text{if } i < j \end{cases} \quad .$$

We then see that $t - \alpha_j^t \leq m-1$ for all j, $t=0,1,\ldots,$ and therefore $L = m-1$. Proposition 9 can be extended to the case where the $\mathcal{C}^t$'s are not all equal by replacing $\beta_{\mathcal{C}}$ with μ where μ is given by (27). Note that the proof of Proposition 9 relies only on the contractivity of $T_{\mathcal{C}}$ and therefore Proposition 9 holds for any contractive (with respective to the sup norm) mapping. For some recent results on distributed computation of fixed points see [4].

A Numerical Example

We illustrate the relaxation method with a very simple example. We consider solving the following problem using the relaxation method :

$$\begin{array}{ll} \text{Maximize} & a^T x \\ & \\ \text{Subject to} & C^1 x \leq d^1 \\ & C^2 x \leq d^2 \end{array}$$

where $a \geq 0$, and

$$C^1 = \begin{bmatrix} 1 & -1/2 \\ -1/2 & 1 \end{bmatrix},\ d^1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad ; \quad C^2 = \begin{bmatrix} 1 & -1/4 \\ -3/4 & 1 \end{bmatrix},\ d^2 = \begin{bmatrix} 1/4 \\ 1 \end{bmatrix}.$$

For the above problem, we obtain that

$$\beta = 3/4 \quad ; \quad x^* = \begin{bmatrix} 4/7 \\ 9/7 \end{bmatrix}.$$

The only nontrivial partitioning of M is {{1},{2}} which yields

$$T_1(x) = \min\{ x_2/2, 1/4+x_2/4 \} \quad ; \quad T_2(x) = \min\{ 1+x_1/2, 1+3x_1/4 \}.$$

Since $m=2$ for the above example, the only possible sequence of computations is when P_1 and P_2 alternate in computing. If we denote x_i^t to be the value of ith coordinate held by P_i after the t-th computation, and x^t to be the vector whose ith entry is x_i^t (x^0 is the initial estimate of x^*), then for $x^0 = (2, 0)$ and with P_1 initiating the computations, we obtain the following sequence of iterates as shown in the figure below :

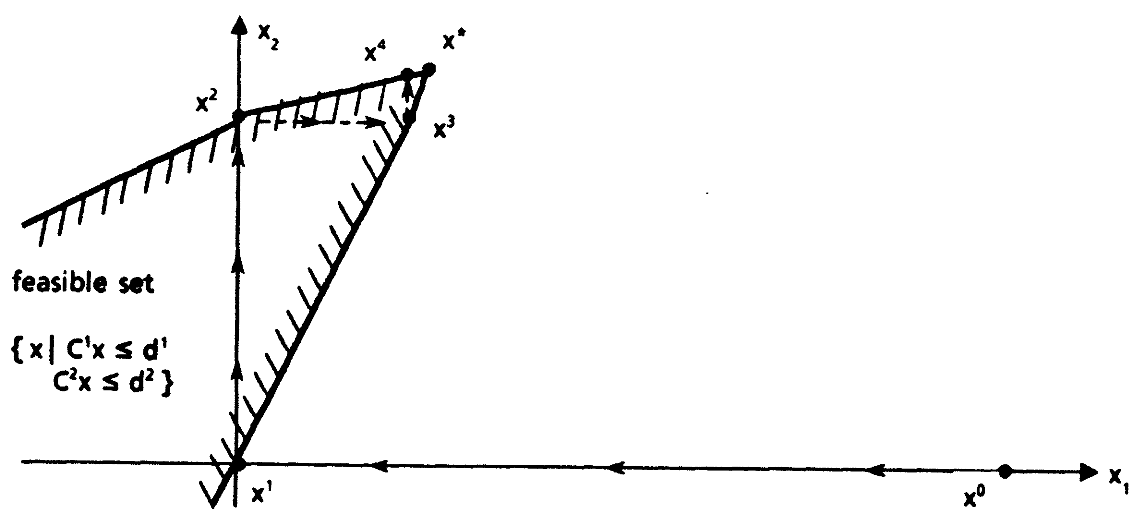

Figure 1 $\{x^t\}$ converging to x^*, the fixed point of T_C.

6. Conclusion

The method proposed in this paper is simple both in concept and in implementation. Yet despite this simplicity it possesses very strong convergence properties. Such strong properties are due in great part to the special structure of the problems themselves. It is possible that other classes of problems exist for which results similar to those obtained here hold and, in particular, it would be of practical as well as theoretical interest to generalize the rate of convergence result on the asynchronous, distributed implementation of the proposed method. This interest stems from the growing role which distributed computation plays in the area of optimization.

Acknowledgement : I wish to thank Professor Dimitri Bertsekas for his support and guidance in this work.