

PARALLEL SUCCESSIVE OVERRELAXATION METHODS
FOR SYMMETRIC LINEAR COMPLEMENTARITY PROBLEMS
AND LINEAR PROGRAMS

by

O. L. Mangasarian & R. De Leone

Computer Sciences Technical Report #647

June 1986

2. Preliminary Background

In this section we give some background results needed to derive our parallel SOR algorithm for solving the symmetric linear complementarity problem and linear programs. We begin first with a special case of the serial SOR Algorithm 2.1 of Ref. 1.

Algorithm 2.1 (Serial SOR algorithm for (1)) Let $z^0 \geq 0$. For $i = 0, 1, 2, \dots$, let

$$z^{i+1} = \left(z^i - \omega E^i (Mz^i + q + K^i(z^{i+1} - z^i)) \right)_+ \quad (3)$$

where $\omega > 0$, and $\{E^i\}$ and $\{K^i\}$ are bounded sequences of matrices in $R^{n \times n}$, with each E^i being a positive diagonal satisfying

$$E^i > \alpha I \quad (4)$$

for some $\alpha > 0$ and such that for some $\gamma > 0$

$$y((\omega E^i)^{-1} + K^i - M/2)y \geq \gamma \|y\|_2^2, \quad \forall i, \forall y \in R^n \quad (5)$$

Remark 2.2 If we let

$$L + D + U = M \quad (6)$$

where L is the strictly lower triangular part of M , D is the diagonal of M and U is the strictly upper triangular part of M , then the iteration (3) is an explicit one if we set $K^i = L, U$ or 0 . More specifically for $K^i = L$ or U and $E^i = D^{-1}$ (assuming that D is positive) iteration (3) gives the projected SOR algorithm studied in Ref. 1 and condition (5) becomes the familiar SOR relaxation factor condition: $0 < \omega < 2$. When $K^i = 0$ we have the projected Jacobi method (Ref. 1). However K^i may be any matrix as long as (5) is satisfied, in which case the iteration (3) can be considered as solving a (hopefully simpler) linear complementarity problem.

An important useful modification of Algorithm 2.1 has been proposed by Subramanian (Ref. 6) in which instead of taking z^{i+1} of (3), any other point in the nonnegative orthant is taken with a value of f not exceeding $f(z^{i+1})$. This leads to the following algorithm.

Algorithm 2.3 (Modified serial SOR algorithm for (1)) Let $z^0 \geq 0$. For $i = 0, 1, 2, \dots$, let

$$s^i = \left(z^i - \omega E^i (Mz^i + q + K^i(s^i - z^i)) \right)_+ \quad (7)$$

where $\omega > 0$ and the sequences of matrices $\{E^i\}$ and $\{K^i\}$ satisfy all the requirements of Algorithm 2.1. Choose $z^{i+1} \geq 0$ such that $f(z^{i+1}) \leq f(s^i)$.

The important point to note about Algorithm 2.2 is that it allows a whole class of algorithms to be based on Algorithm 2.1. Typically, z^{i+1} of Algorithm 2.2 is obtained from s^i by some sort of line search.

The convergence of Algorithm 2.1 was established in Theorem 2.1 of Ref. 1, and the convergence of Algorithm 2.2 in Theorem 4.1 of Chapter 3 of Ref. 6. We combine these results into the following fundamental convergence result under no assumption on the matrix M other than symmetry.

Theorem 2.1 (Convergence of serial SOR algorithms for (1)) Let M be symmetric. Each accumulation point of the sequence $\{z^i\}$ of Algorithm 2.1 or 2.3 solves the linear complementarity problem (1).

Note that Theorem 2.1 does not guarantee the existence of an accumulation point for the sequence $\{z^i\}$. To do that we need additional assumptions (Ref. 1, Theorem 2.2) such as the following.

Theorem 2.2 (Strong convergence of serial SOR algorithms for (1)) Let M be symmetric and positive semidefinite, and let

$$Mz + q > 0 \quad \text{for some } z \in R^n \quad (8)$$

Then, the sequences $\{z^i\}$ of Algorithms 2.1 and 2.3 are bounded and have accumulation points. Each accumulation point of $\{z^i\}$ solves (1).

With this background material we are prepared to introduce our parallel SOR algorithm for solving the symmetric linear complementarity problem (1).

3. Parallel SOR for the Symmetric Linear Complementarity Problem

The key idea of our approach here is to consider K^i of Algorithm (2.1) as a substitution operator which replaces the old data z^i by the new data z^{i+1} . If for example $K^i := L$, where L is the strictly lower triangular part of the whole matrix M , then z_j^{i+1} replaces z_j^i during the computation of z_ℓ^{i+1} for all $\ell > j$. Now, consider instead the following procedure. Break M into k blocks of rows as follows:

$$M =: \begin{bmatrix} M_{I_1} \\ M_{I_2} \\ \vdots \\ M_{I_k} \end{bmatrix} \quad (9)$$

where the blocks M_{I_j} correspond to the variables z_{I_j} and $\{I_1, I_2, \dots, I_k\}$ is a consecutive partition of $\{1, 2, \dots, n\}$. Now partition M_{I_j} as follows

$$M_{I_j} =: [M_{I_j I_j} \quad M_{I_j \bar{I}_j}] \quad (10)$$

where \bar{I}_j is the complement of I_j in $\{1, 2, \dots, n\}$. Thus $M_{I_j I_j}$ is a principal square submatrix of M with elements M_{rs} , $r \in I_j$ and $s \in I_j$. We further partition $M_{I_j I_j}$ as follows

$$M_{I_j I_j} =: L_{I_j I_j} + D_{I_j I_j} + U_{I_j I_j} \quad (11)$$

where $L_{I_j I_j}$ is the strictly lower triangular part of $M_{I_j I_j}$, $D_{I_j I_j}$ its diagonal part and $U_{I_j I_j}$ its strictly upper triangular part. Thus for example if $k = 3$ we would have the following decomposition of M

$$M = \begin{bmatrix} M_{I_1} \\ M_{I_2} \\ M_{I_3} \end{bmatrix} = \begin{bmatrix} M_{I_1 I_1} & M_{I_1 I_2} & M_{I_1 I_3} \\ M_{I_2 I_1} & M_{I_2 I_2} & M_{I_2 I_3} \\ M_{I_3 I_1} & M_{I_3 I_2} & M_{I_3 I_3} \end{bmatrix} \quad (12)$$

Now let K^i of Algorithm 2.1 be defined by a block diagonal matrix as follows

$$K^i = K =: \begin{bmatrix} L_{I_1 I_1} & & & \\ & L_{I_2 I_2} & & \\ & & \ddots & \\ & & & L_{I_k I_k} \end{bmatrix} \quad (13)$$

where each $L_{I_j I_j}$ is a strictly lower triangular matrix defined in (11).

Algorithm 2.1 can now be performed for each row block I_j , $j = 1, \dots, k$, simultaneously, that is in parallel. Note that this is not a block Jacobi iteration. More specifically we have the following algorithm.

Algorithm 3.1 (Parallel SOR for (1)) Let $\{I_1, I_2, \dots, I_k\}$ be a consecutive partition of $\{1, 2, \dots, n\}$, let the diagonal D of M have positive elements and let $z^0 \geq 0$. For $i = 0, 1, 2, \dots$, let

$$z_{I_j}^{i+1} = \left(z_{I_j}^i - \omega D_{I_j I_j}^{-1} (M_{I_j} z^i + q_{I_j} + L_{I_j I_j} (z_{I_j}^{i+1} - z_{I_j}^i)) \right)_+ \quad (14)$$

$$j = 1, \dots, k$$

where

$$0 < \omega < \min_{1 \leq j \leq k} \min_{\ell \in I_j} \frac{2}{1 + \sum_{s \in \bar{I}_j} |M_{\ell s}| / D_{\ell \ell}} \quad (15)$$

Remark 3.2 Iteration (14) can be performed in parallel on k processors. The new value z^{i+1} must then be shared between the k processors.

Remark 3.3 If all $M_{I_j \bar{I}_j}$ are zero then $0 < \omega < 2$ for all j , which is the standard SOR relaxation factor range. This corresponds to k uncoupled linear complementarity problems. If all $M_{I_j \bar{I}_j}$ are small relative to D_{I_j} , which corresponds to a loosely coupled linear complementarity problem, the upper bound on ω given by (15) is close to 2.

We state now a parallel SOR version of Algorithm 2.3.

Algorithm 3.4 (Modified parallel SOR for (1)) Let the assumptions of Algorithm 3.1 hold. For $i = 0, 1, 2, \dots$, let

$$s_{I_j}^i = \left(z_{I_j}^i - \omega D_{I_j I_j}^{-1} (M_{I_j} z^i + q_{I_j} + L_{I_j I_j} (s_{I_j}^i - z_{I_j}^i)) \right)_+ \quad (16)$$

$$j = 1, \dots, k$$

where ω satisfies (15). Chose $z^{i+1} \geq 0$ such that $f(z^{i+1}) \leq f(s^i)$.

We can establish the convergence of Algorithms 3.1 and 3.4 by appealing to Theorem 2.4. We have then the following convergence result.

Theorem 3.5 (Convergence of parallel SOR algorithms for (1)) Let M be symmetric. Each accumulation point of the sequence $\{z^i\}$ of Algorithms 3.1 and 3.4 solves the linear complementarity problem (1).

Proof By Theorem 2.4 we only need to establish that condition (5) is satisfied by the choice (13) for K^i , $E^i := D^{-1}$ and ω satisfying assumption (15). We have

$$\begin{aligned}
& y((\omega E^i)^{-1} + K^i - M/2)y \\
&= \frac{1}{2}y(2\omega^{-1}D + 2K - M)y \\
&= \frac{1}{2}\sum_{r=1}^k y_{I_r}(2\omega^{-1}D_{I_r} + 2K_{I_r} - M_{I_r})y \\
&= \frac{1}{2}\sum_{r=1}^k y_{I_r} \left[(2\omega^{-1}D_{I_r I_r} + 2K_{I_r I_r} - M_{I_r I_r})y_{I_r} - \sum_{s \neq r} M_{I_r I_s} y_{I_s} \right] \\
&= \frac{1}{2}\sum_{r=1}^k y_{I_r} \left[(2\omega^{-1}D_{I_r I_r} + 2L_{I_r I_r} - (L_{I_r I_r} + D_{I_r I_r} + U_{I_r I_r}))y_{I_r} - \sum_{s \neq r} M_{I_r I_s} y_{I_s} \right] \\
&= \frac{1}{2}\sum_{r=1}^k y_{I_r} \left[(2\omega^{-1} - 1)D_{I_r I_r} y_{I_r} - \sum_{s \neq r} M_{I_r I_s} y_{I_s} \right] \\
&\quad \text{(Because } L = U^T) \\
&= \frac{1}{2}\sum_{r=1}^k y_{I_r} \left[(2\omega^{-1} - 1)D_{I_r I_r} \quad - M_{I_r \bar{I}_r} \right] \begin{bmatrix} y_{I_r} \\ y_{\bar{I}_r} \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} y \\ y \end{bmatrix} \begin{bmatrix} (2\omega^{-1} - 1)D_{I_1 I_1} & & -M_{I_1 \bar{I}_1} \\ & \ddots & \\ -M_{I_k \bar{I}_k} & & (2\omega^{-1} - 1)D_{I_k I_k} \end{bmatrix} \begin{bmatrix} y \\ y \end{bmatrix} \geq \gamma \|y\|_2^2
\end{aligned}$$

where last inequality holds for some $\gamma > 0$ because of the positive definiteness of the symmetric $n \times n$ matrix (preceding the inequality) which is induced by its row diagonal dominance (Ref. 5). The row diagonal dominance is precisely a consequence of assumption (15). ■

Having established that the parallel SOR Algorithms 3.1 and 3.4 can be considered as special cases of the general serial Algorithms 2.1 and 2.3 respectively, the following strong convergence result is a direct consequence of Theorem 2.5.

Theorem 3.6 (Strong convergence of parallel SOR algorithms for (1)) Let M be symmetric and positive semidefinite and let assumption (8) hold. Then the sequences $\{z^i\}$ of

Algorithms 3.1 and 3.4 are bounded and have accumulation points. Each accumulation point of $\{z^i\}$ solves (1).

Remark 3.7 Minor changes in the proof of Theorem 3.5 allows us to have a different ω_j for each $j = 1, \dots, k$ in (14). In particular all we need is that for $j = 1, \dots, k$, ω_j must satisfy

$$0 < \omega_j < \min_{\ell \in I_j} \frac{2}{1 + \sum_{s \in I_j} |M_{\ell s}| / D_{\ell\ell}} \quad (17)$$

This results in larger stepsizes for Algorithms 3.1 and 3.4.

We now turn our attention to the parallel solution of linear programs.

4. Parallel Solution of Linear Programs

The key idea here is to find the least 2-norm solution of a linear program by converting the problem to a positive semidefinite linear complementarity problem (Refs. 2-3) and to use the parallel SOR procedures proposed in the previous section. We will present a parallel implementation here of the linearly convergent iterative scheme proposed in Ref. 7.

We consider the linear program

$$\min_x cx \quad \text{subject to } Ax \geq b, x \geq 0 \quad (18)$$

where $c \in R^h$, $b \in R^m$ and $A \in R^{m \times h}$, and its dual

$$\max_u bu \quad \text{subject to } A^T u \leq c, u \geq 0 \quad (19)$$

It is known (Refs. 2-3) that \bar{x} is the unique least 2-norm solution to (18) if and only if \bar{x} is the unique solution to the quadratic program

$$\min_x cx + \frac{\varepsilon}{2} xx \quad \text{subject to } Ax \geq b, x \geq 0 \quad (20)$$

for all $\varepsilon \in (0, \bar{\varepsilon}]$ for some $\bar{\varepsilon} > 0$. The dual to the quadratic program (20) is

$$\max_{x,u,v} -\frac{\varepsilon}{2} xx + bu \quad \text{subject to } v = \varepsilon x - A^T u + c, (u,v) \geq 0 \quad (21)$$

To solve (20) for a fixed positive ε we shall use the parallel SOR procedures of Section 3 applied to its dual (21) with the variable x eliminated through the dual constraint

$$x = (A^T u + v - c)/\varepsilon \quad (22)$$

and thus obtaining the dual problem

$$\min_{(u,v) \geq 0} \theta(u,v) := \min_{(u,v) \geq 0} \frac{1}{2} \|A^T u + v - c\|_2^2 - \varepsilon bu \quad (23)$$

which is precisely of the form (2) with a positive semidefinite matrix and hence is equivalent to the symmetric linear complementarity problem (1) on R^{m+h} with $M := \nabla^2 \theta(u,v)$ and $q := \nabla \theta(0,0)$. We shall now describe a linearly convergent sequential parallel SOR

procedure for solving (23) based on Ref. 7 and the results of Section 3. We first need a definition.

Definition 4.1 (Approximate solutions to (23) and (20)) For a fixed positive ε any point in R_+^{m+h} is an approximate solution to the dual quadratic program (23) and is designated by $(u(\varepsilon), v(\varepsilon))$. The corresponding $x(\varepsilon)$ in R^h defined by (22) with $(u, v) = (u(\varepsilon), v(\varepsilon))$ is an approximate solution to the quadratic program (20). The residual $r(\varepsilon)$ associated with $(u(\varepsilon), v(\varepsilon), x(\varepsilon))$ is defined by

$$r(\varepsilon) := [\|x(\varepsilon)v(\varepsilon) + u(\varepsilon)(Ax(\varepsilon) - b)\| + \|(b - Ax(\varepsilon))_+\|_\infty + \|(-x(\varepsilon))_+\|_\infty]^{1/2} \quad (24)$$

Note that for an $\varepsilon > 0$ and an approximate solution $(u(\varepsilon), v(\varepsilon))$ to (23) and a corresponding approximate solution $x(\varepsilon)$ to (20), $r(\varepsilon) = 0$ if and only if $(u(\varepsilon), v(\varepsilon))$ is an exact solution of (23) and $x(\varepsilon)$ is the unique exact solution of (20).

We are prepared now to state and prove a linearly convergent parallel SOR procedure for computing the least 2-norm solution of the linear program (18).

Theorem 4.2 (Linearly convergent parallel SOR for linear programs) Assume that the linear program (18) is solvable and that $b \neq 0$. Let $\{\varepsilon_0, \varepsilon_1, \dots\}$ be a decreasing sequence of positive numbers such that

$$\varepsilon_{i+1} = \mu \varepsilon_i \quad \text{for some } \mu \in (0, 1) \quad (25)$$

and let $\{u(\varepsilon_i), v(\varepsilon_i), x(\varepsilon_i)\}$ be a corresponding sequence of approximate solutions to (23) and (20) satisfying Definition 4.1 and obtained by either of the parallel SOR Algorithms 3.1 or 3.4 applied to (23) and such that their residuals as defined by (24) satisfy

$$r(\varepsilon_{i+1}) \leq \nu r(\varepsilon_i) \quad (26)$$

for some $\nu \geq 0$ such that

$$\nu < \mu^{1/2} \quad (27)$$

Then the sequence $\{x(\varepsilon_i)\}$ converges to \bar{x} , the least 2-norm solution of the linear program (18) at the linear root rate

$$\|x(\varepsilon_i) - \bar{x}\|_2 \leq \delta(\nu/\mu^{1/2})^i \quad \text{for } i \geq \bar{i} \quad (28)$$

for some constant δ and some integer \bar{i} .

Proof See Theorem 3.7 of Ref. 7. ■

5. Conclusion

We have presented a framework for the parallel solution of symmetric linear complementarity problems and linear programs. The proposed SOR algorithm is best suited for a tightly-coupled shared-memory multiprocessor such as the one to be acquired by the Computer Sciences Department at Madison. However we plan to test the proposed algorithm on the existing loosely-coupled 20-processor token-ring-connected Crystal machine (Ref. 8) of the Computer Sciences Department in order to develop efficient computational implementations of our algorithm. Because we have been able to solve sparse linear programs of size 20,000 variables and 5,000 constraints by the serial version of our SOR procedure in 78 minutes on a VAX 11/780 (Ref. 2), we are hopeful of solving substantially larger problems by our parallel approach.

References

1. MANGASARIAN, O. L., *Solution of Symmetric Linear Complementarity Problems by Iterative Methods*, Journal of Optimization Theory and Applications, Vol. 22, pp. 465-485, 1977.
2. MANGASARIAN, O. L., *Normal Solutions of Linear Programs*, Mathematical Programming Study, Vol. 22, pp. 206-216, 1984.
3. MANGASARIAN, O. L., *Sparsity-Preserving SOR Algorithms for Separable Quadratic and Linear Programming*, Computers and Operations Research, Vol. 11, pp. 105-112, 1984.
4. MANGASARIAN, O. L. and DE LEONE, R., *A Parallel Successive Overrelaxation (SOR) Algorithm for Linear Programming*, 12th International Symposium on Mathematical Programming, Massachusetts Institute of Technology, Cambridge, Massachusetts, August 5-9, 1985.
5. ORTEGA, J. W., *Numerical Analysis A Second Course*, Academic Press, New York, New York, 1972.
6. SUBRAMANIAN, P. K., *Iterative Methods of Solution for Complementarity Problems*, University of Wisconsin-Madison, Ph.D. Thesis, 1985.
7. MANGASARIAN, O. L. and DE LEONE, R., *Error Bounds for Strongly Convex Programs and (Super)Linearly Convergent Iterative Schemes for the Least 2-Norm Solution of Linear Programs*, University of Wisconsin-Madison, Computer Sciences Department Report No. 631, 1986.
8. DEWITT, D., FINKEL, R. and SOLOMON, M., *The CRYSTAL Multicomputer: Design and Implementation Experience*, University of Wisconsin-Madison, Computer Sciences Department Report No. 553, 1984. (To appear in IEEE Transactions on Software Engineering.)

