RESEARCH PROJECT: Numerical Optimization

Principal Investigator: Francesco ZIRILLI
Contractor: Università di Roma "La Sapienza"
Contract Number: DAJA45-86-C-0028
Fifth Periodic Report
Jan. '88 - July '88

The research reported in this document has been made possible through the support and sponsorship of the U.S. Government through its European Research Office of the U.S. Army.

This report is intended only for the internal management use of the contractor and the U.S. Government.
**Numerical Optimization**

This is the Fifth Periodic Report on Contract DAJA 45-86-C-0028; contains:

1. Statement of scientific work done during the reporting period...
## INDEX

1. Statement of scientific work done during the reporting period.......................... pag. 1

2. Research plans for the immediate future........... " 5

3. Administrative actions........................................ " 6


5. Appendix 2.................................................. " 20
§1. STATEMENT OF SCIENTIFIC WORK DURING THE REPORTING PERIOD.

Let $\mathbb{R}^N$ be the $N$ dimensional real euclidean space, $x = (x_1, x_2, \ldots, x_N)^T \in \mathbb{R}^N$ a generic vector, the superscript $T$ means transpose, $\langle \cdot, \cdot \rangle$ is the euclidean scalar product and $\| \cdot \|$ the euclidean norm.

We have considered the linear programming problem; that is:

Problem 1 (Linear programming) Given $b, c \in \mathbb{R}^N$ and $A$ an $m \times N$ matrix ($m < N$), solve the following minimization problem:

$$\begin{align*}
\text{minimize} & \quad \langle c, x \rangle \\
\text{for} \quad x & \quad \text{such that} \\
& x \geq 0 \\
& Ax - b \geq 0
\end{align*}$$

where $x \geq 0$ means that each component of $x$ is greater or equal to zero and similarly $Ax - b \geq 0$.

The Linear programming problem is transformed into a Linear Complementarity Problem using the Lagrange multipliers as follows:

Let

$$
\begin{align*}
Z &= \begin{pmatrix} x \\ \lambda \end{pmatrix} \\
\bar{Z} &= \begin{pmatrix} -c \\ b \end{pmatrix} \\
\bar{A} &= \begin{pmatrix} 0 & -A^T \\ A & 0 \end{pmatrix}
\end{align*}
$$

Problem 1 becomes
Problem 2 (Linear Complementarity Problem). Find $z$ such that:

\[ z \geq 0, \quad \tilde{A}z - \tilde{b} \geq 0 \]

\[ \langle z, \tilde{A}z - \tilde{b} \rangle = 0 \]

Finally the Linear Complementarity Problem is transformed into a global optimization problem via the transformation given at page 1 of the Fourth Periodic Report. In this way the linear programming problem is reduced to the problem of finding the minimizers of a function $F(z) \geq 0$ such that $F = 0$ at the minimizers.

If the original linear programming problem has a unique solution the corresponding minimization problem has a unique global minimizer, but unfortunately the function $F$ may have a rather complicated set of local minimizers including "cylindrical valleys". A direct application of a local minimization techniques such as GRACON (see Fourth Periodic Report) is unsuccessful. So that several modified algorithms have been considered the most promising ones are:

(i) A perturbation method. Let $\varepsilon \in \mathbb{R}, \varepsilon > 0$, the matrix $\tilde{A}$ is substituted with $\tilde{A}(\varepsilon) = \tilde{A} + \varepsilon I$ where $I$ is the identity matrix. The idea is to find the solution $z(\varepsilon)$ of the problem correspondent to the matrix $\tilde{A}(\varepsilon)$ and to let $\varepsilon$ go to zero. If $\varepsilon$ goes to zero too quickly the conjugate gradient method may remain trapped in a local minimizer, if $\varepsilon$ goes to zero too slowly the computational cost becomes
excessive. In Table 1 the results obtained with this method are indicated with DOD4 DOD9.

(ii) Avoiding the cylindrical valleys. When the conjugate gradient method remains trapped in a cylindrical valley the minimization procedure is stopped, the direction of the valley is computed and a large step is taken in the direction of the valley in the "descending" direction, finally the conjugate gradient minimization algorithm is started again. In Table 1 the results obtained with this method are indicated with DOD4 DOD9.

The test problems considered in Table 1 have been constructed following the suggestions of De Leone and Mangasarian (see Appendix 2). The numerical experience has been obtained on a VAX 8530 with VMS 4,5 Operating System.
TABLE 1

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>NFEV</td>
<td>T</td>
<td>M</td>
<td>NFEV</td>
<td>T</td>
</tr>
<tr>
<td>50</td>
<td>694</td>
<td>2''97</td>
<td>50</td>
<td>315</td>
<td>2''24</td>
</tr>
<tr>
<td>100</td>
<td>1502</td>
<td>7''.87</td>
<td>100</td>
<td>716</td>
<td>4''.21</td>
</tr>
<tr>
<td>500</td>
<td>4176</td>
<td>1''.35''.08</td>
<td>500</td>
<td>2154</td>
<td>43''.44</td>
</tr>
<tr>
<td>1000</td>
<td>9620</td>
<td>7''.4''.74</td>
<td>1000</td>
<td>1333</td>
<td>55''.39</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5000</td>
<td>12548</td>
<td>46''.26''.80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10000</td>
<td>14241</td>
<td>1''.54''.16''.51</td>
</tr>
</tbody>
</table>

Legenda

M dimension of the complementarity problem \( z \in \mathbb{R}^M \)

NFEV number of function and gradient evaluation of the function that must be minimized

T running time on the VAX 8530 (Example 1''.54''.18''.51 = 1 hour 54 minutes, 18 seconds, 51 seconds/100.

Tolerance required: Final function value \( \leq 10^{-20} \)

Finally some geophysical application of linear programming has been considered (see Appendix 1) since in our opinion the random generated test problems are not completely satisfactory.
§2. RESEARCH PLANS FOR THE IMMEDIATE FUTURE.

In the immediate future we plan to pursue the following objectives:

(i) study Karmarkar and Renegar methods for linear programming in the context of continuation methods;
(ii) study the nonlinear complementarity problem.
§3. ADMINISTRATIVE ACTIONS.

None.
§4. Appendix 1: L. Misici, F. Zirilli: "The inverse gravi-
metry problem: An application to the northern San Fran-
cisco craton granite" submitted to J. of the Geological
Society London.
THE INVERSE GRAVIMETRY PROBLEM: AN APPLICATION
TO THE NORTHERN SAN FRANCISCO CRATON GRANITE*

Luciano Misiu
Dipartimento di Matematica e Fisica
Università di Camerino
62052 Camerino (MC) Italy

Francesco Zirilli
Dipartimento di Matematica "G. Castelnuovo"
Università di Roma "La Sapienza"
00185 Roma Italy

Abstract
From the knowledge of the anomalies of the gravitational field we reconstruct the mass density
distribution in a large region of the state of Bahia (Brazil). This inverse gravimetry problem has
been translated in a linear programming problem and solved using the simplex method. Both two
and three dimensional models have been considered.

1. Introduction
In recent years a great deal of attention has been attracted by the study of several inverse
problems in science and technology. In particular in science we mention the inverse problem of
quantum mechanics, that is the reconstruction of a potential from its scattering data [1] and the
famous problem "can you hear the shape of a drum", that is the reconstruction of the shape
of a region in Euclidean space from the knowledge of spectral properties of suitable differential
operators [2]. In technology we mention the radar technology that is the use of electromagnetic
waves to detect not visible objects, the sonar and ecography technology that is the use of acoustics
waves to detect properties of regions that are not directly accessible because they are underwater
or inside the human body.

* The research reported in this document has been made possible through the support and sponsorship of the
U.S. Government through the European Research Office of the U.S. Army under contract n. DAJA 45-86-C-0028
at the University of Roma "La Sapienza" and of the Ministero Pubblica Istruzione under contract 60% 1087 at the
University of Camerino.
There are many inverse problems in geophysics both of fundamental interest such as the reconstruction of the Earth structure from knowledge of elastic, gravimetric and geomagnetic data or of more applied nature such as the use of similar data in geophysical prospecting to localize gas or oil.

In this paper we consider an inverse problem in gravimetry. Let \( x, y, z \) be cartesian coordinate in the three dimensional space, given a mass density \( \rho(x, y, z) \) in a certain region \( \Omega \), the direct problem of gravimetry consists in finding the gravitational potential \( V(x, y, z) \) generated by \( \rho \), that is in solving the Poisson’s equation:

\[
\Delta V = -4\pi \rho \quad \text{in } \Omega \tag{1.1}
\]

where \( \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \) is the laplacian, \( f = 6.67 \times 10^{-8} \text{cm}^3 \cdot \text{g/sec}^2 \) is the gravitational constant and suitable boundary conditions on the boundary of \( \Omega \), \( \partial \Omega \), are specified. Vice versa given the gravitational potential \( V(x, y, z) \) in a certain region \( \Omega \), the inverse gravimetry problem consists in finding the mass density \( \rho(x, y, z) \) that generates \( V \), that is in solving the inverse Poisson’s equation:

\[
\Delta^{-1} V = -\frac{1}{4\pi f} \quad \text{in } \Omega \tag{1.2}
\]

where \( \Delta^{-1} \) is the inverse laplacian with the appropriate boundary conditions on \( \partial \Omega \).

Almost all inverse problems and in particular the inverse gravimetry problem are ill posed, that is, to an arbitrarily small perturbation of the data \( V \) can correspond an arbitrarily large perturbation of the solution \( \rho \).

The great difficulty of solving the inverse gravimetry problem in practical situations is due to its ill posedness since the values of the gravitational field, that is the data, are obtained from experiments and so are affected by errors. In order to avoid this difficulty and restore a well behaved dependence of the solution \( \rho \) from the data \( V \) it is very useful to introduce some a priori constraints that \( \rho \) should satisfy.

In [3], [4] [5] using these ideas the inverse gravimetry problem has been reduced to a linear programming problem in a way that we will see later. In this paper using this linear programming formulation of the inverse gravimetry problem we study the northern San Francisco craton granite in Brazil on the basis of measurements of the residual Bouguer anomaly of the gravitational field taken from [6] and [7], see Fig.1. From these data we reconstruct a two dimensional section of the mass density along the BB’ segment of Fig.1. This section is about 270Km long and 20Km deep, see Fig.3. Moreover we reconstruct a three dimensional section of the mass density in the region bounded by the dotted rectangle of Fig.1. This section on the surface of the Earth is a rectangle of about 240Km x 189Km and is about 20Km deep, see Fig.4.

We have found that in the center of this three dimensional region there is a body of granite of mass density \( \rho = 2.57 \text{g/cm}^3 \) while the average mass density of this region is \( \rho = 2.67 \text{g/cm}^3 \). This body in the BB’ segment is surfacing for about 30Km and is about 67Km long on its bottom which is about 16Km deep, see Fig.3. These results are confirmed by the three dimensional reconstruction of \( \rho \) (Fig.4) and they are in good agreement with the results obtained by Ussami and Bott in [6]. The granite body found by Ussami and Bott in [6] is a little bit smaller than ours.
In section 2 we describe our mathematical model and we reduce the inverse gravimetry problem to a linear programming problem. In section 3 we present the data that we have used and the results obtained in the northern San Francisco craton granite (Brazil) using the mathematical model described in section 2.

2. The Mathematical model

Here we will follow the work of Safon, Vasseur and Cuer [4], it is well known that, using the Green's function $G(z, y, z')$ [81], the equation (1.2) can be reformulated as an integral equation as follows

$$ \int_{\Omega} G(x, y, z, x', y', z') \rho(x', y', z') dx' dy' dz' = V(x, y, z) \quad (2.1) $$

Let us cut the domain $\Omega$ in a great number $N$ of cubic domains $\omega_n$, the function $\rho(x, y, z)$ will be approximated on each cube $\omega_n$ by a constant $\rho_n$ and the Green's function $G$ will be approximated by a matrix $G = (g_{ij})$; $i = 1, 2, \ldots, M$; $j = 1, 2, \ldots, N$. Finally let $V_i$, $i = 1, 2, \ldots, M$ ($M \leq N$) be the measurements of the gravitational field $V(x, y, z)$ in $x_i, y_i, z_i$ that belongs to the cubic domain $\omega_i, i = 1, 2, \ldots, M$. So that equation (2.1) can be discretized as follows:

$$ \sum_{j=1}^{N} g_{ij} \rho_j = V_i , \quad i = 1, 2, \ldots, M \quad (2.2) $$

where $g_{ij} = \int_{\omega_n} G(x, y, z, x', y', z') dx' dy' dz'$. In general is natural to assume that the data $V_i$ are available only on cubic domains $\omega_n$ that have one side on the accessible surface of the region $\Omega$, so that $M < N$, see Fig 4 and the linear system (2.2) is underdetermined. Moreover since the data $V_i$, $i = 1, 2, \ldots, M$ are known only with a certain error $\epsilon_i > 0$, $i = 1, 2, \ldots, M$ we substitute the system of linear equations (2.2) with the system of linear inequalities

$$ V_i - \epsilon_i \leq \sum_{j=1}^{N} g_{ij} \rho_j \leq V_i + \epsilon_i , \quad i = 1, 2, \ldots, M \quad (2.3) $$

The mass density $\rho$ is greater or equal to zero, so that we may impose

$$ \rho_j \geq 0 , \quad j = 1, 2, \ldots, N \quad (2.4) $$

or more realistically in a geophysical problem

$$ \rho_{j}^{\min} \leq \rho_j \leq \rho_{j}^{\max} , \quad j = 1, 2, \ldots, N \quad (2.5) $$

where $\rho_{j}^{\min}$ and $\rho_{j}^{\max}$, $j = 1, 2, \ldots, N$ are the minimum and the maximum values for $\rho_j$ assigned on the basis of physical intuition. Both the linear system (2.2) or the more realistic system of inequalities (2.3), (2.5) if compatible have in general an infinite number of solutions.

In order to restore uniqueness we introduce a functional to be optimized on the set of points satisfying (2.3), (2.5), that is

$$ J = \sum_{k=1}^{L} \rho_k \text{vol}(\omega_k) \quad (2.6) $$
where \( \text{vol}(\omega_{k_i}) \) is the volume of the cube \( \omega_{k_i} \), so that \( J \) is the total mass of the body \( \Omega \) or of some part of it.

The linear programming problem that we consider is the following one: minimize the functional \( J \) given by (2.6) subject to the constraints (2.3), (2.5).

In section 3 we will also use a two-dimensional model, in this case we assume that the region \( \Omega \) has an infinite extension in the horizontal \( x \) coordinate and that \( \rho \) is a function of \( x \), the vertical coordinate, and \( y \) only. Reasoning as in the three-dimensional case with an appropriate Green's function \( G \) and data \( V_i, \ i = 1, 2, \cdots, M \) it is easy to obtain a linear programming problem analogous to the one obtained for the three-dimensional model given by (2.3), (2.5), (2.6).

3. Results and conclusions

Here we will consider two special applications of the method described in section 2. That is a two-dimensional vertical section and a three-dimensional slice of a large region in the state of Bahia in Brazil. In our computations the gravitational anomaly is measured in \( \text{mgal/s} \) and the distances in \( \text{Km} \). The gravitational anomalies, that is the residual Bouguer anomalies, are reported in Table 1 for the three-dimensional problem and in Fig. 2 for the two-dimensional problem. Moreover we assume that the average mass density in the regions considered is \( \rho_0 = 2.67\text{g/cm}^3 \). The data of table 1 and Fig. 2 have been regularized in order to avoid numerical instability due to the ill posedness of the problem. In particular the data of Fig. 2, that are taken along the BB' segment of Fig. 1, are highly irregular outside the central region of Fig. 2 where the maximum of the gravitational anomaly is attained so that, as shown in Fig. 2, the data have been regularized outside this region. For the three-dimensional case the numbers in bold face of Table 1 are real gravitational anomalies, these are the data taken in the rectangle \( AA'C'C \) of Fig. 1, while the numbers not in bold face in Table 1, that is the data of the region between the dotted rectangle and \( AA'C'C \) of Fig. 1, have been regularized solving some suitable direct problem. In order to choose the functional \( J \) given by (2.6) we have used an empiric formula [9] that estimates the maximum deepness \( \Delta z \) of a body that generates a certain gravitational anomaly \( V_i, \ i = 1, 2, \cdots, M \) on the surface, that is

\[
\Delta z = 0.65 \frac{\max |V_i|}{\max |\nabla V_i|} \quad (3.1)
\]

where \( \nabla V_i \) is a numerical approximation to the gradient of \( V_i \) expressed in \( \text{mgal/s/Km} \). With our data we obtain \( \Delta z \approx 16\text{Km} \). So that on the basis of this estimate for \( \Delta z \) and of the conclusion of [6] that the body is surfacing we have chosen

\[
J = \sum_{\{y\mid 2.5\text{Km} < y < 16\text{Km}\}} \rho_j \text{vol}\omega_j \quad (3.2)
\]

In particular in the two-dimensional case along the BB' segment of Fig. 1 we have considered 37 data points taken on the surface at a distance of 7.5\text{Km} one from the other. Moreover we assume that the body we are looking for is localized in the region \( z < 20\text{Km} \). So that the rectangle we are working on, as shown in Fig. 3, is \( (7.5\text{Km} \times 37 = 277.5\text{Km}) \times 20\text{Km} \) and has been divided in small rectangles \( \omega_j, \ j = 1, 2, \cdots, 370 \) where \( \omega_j \) is \( 7.5\text{Km} \times 2\text{Km} \) (see Fig. 3) and \( j \) runs from left...
The gravimetric problem that has been solved for the two dimensional case is the following one: find the values of \( \rho_j \), \( j = 1, 2, \cdots, 370 \) that minimize

\[
J = \sum_{j=1}^{370} \rho_j |v_0| \omega_j
\]  

subject to

\[
V_i - \xi_i \leq \sum_{j=1}^{370} \rho_j \leq V_i + \xi_i ; \quad i = 1, 2, \cdots, 370
\]

\[
2.57g/cm^2 \leq \rho_i \leq 2.69g/cm^3 ; \quad 1 \leq j \leq 37
\]

\[
2.57g/cm^2 \leq \rho_i \leq 2.675g/cm^3 ; \quad 38 \leq j \leq 74
\]

\[
2.57g/cm^2 \leq \rho_i \leq 2.675g/cm^3 ; \quad 75 \leq j \leq 370
\]

where \( V_i = \delta_i + \gamma_i \) with \( \delta_i = 2.67g/cm^3 \sum_{j=1}^{370} \rho_j \) and \( \gamma_i \) the anomalies shown in Fig.2 while \( \xi_i = 0.05\gamma_i \). This linear programming problem (3.3),(3.4) has been solved using the FORTRAN package [10] and the solution found is shown in Fig.3.

In the three dimensional problem the dotted rectangle of Fig.1 is of dimension 240Km in the BB' direction (y direction) times 189Km in the AC direction (x direction). This dotted rectangle has been divided in \( 16 \times 7 = 112 \) small rectangles of dimension 15Km in the BB' direction times 27Km in the AC direction, the gravitational anomaly data \( \gamma_i \) are taken in the central points of these small rectangles \((x_i, y_i, 0)\), \( i = 1, 2, \cdots, 7, \quad j = 1, 2, \cdots, 16 \).

| \( y_1 \) | \( y_2 \) | \( y_3 \) | \( y_4 \) | \( y_5 \) | \( y_6 \) | \( y_7 \) | \( y_8 \) | \( y_9 \) | \( y_{10} \) | \( y_{11} \) | \( y_{12} \) | \( y_{13} \) | \( y_{14} \) | \( y_{15} \) | \( y_{16} \) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| -0.18 | -0.25 | -0.35 | -0.40 | -0.82 | -1.41 | -2.69 | -4.57 | -5.34 | -6.57 | -2.69 | -1.41 | -0.82 | 0.32 | 0.35 | 0.30 |
| -0.73 | -0.32 | -0.48 | -0.77 | -1.37 | -9.60 | -17.66 | -22.66 | -48.76 | -57.66 | -37.66 | -23.04 | -1.37 | -0.77 | 0.48 | 0.32 |
| -2.57 | -0.35 | -0.50 | -0.93 | -1.79 | -15.24 | -18.24 | -28.98 | -40.13 | -53.18 | -35.14 | -14.40 | -3.74 | -1.73 | -0.93 | 0.56 | 0.38 |
| -2.77 | -0.41 | -0.61 | -1.01 | -1.85 | -26.86 | -40.32 | -57.60 | -57.60 | -22.08 | -3.74 | -1.73 | -0.93 | 0.56 | 0.38 |
| -2.85 | -0.34 | -0.50 | -1.01 | -1.29 | -12.10 | -23.04 | -42.09 | -52.80 | -39.36 | -22.08 | -3.74 | -1.73 | -0.93 | 0.56 | 0.38 |
| -2.02 | -0.32 | -0.48 | -0.77 | -1.37 | -3.69 | -4.76 | -26.50 | -48.00 | -46.08 | -38.04 | -2.94 | -1.37 | -0.77 | 0.48 | 0.32 |
| 0.15 | 0.23 | 0.34 | 0.55 | 0.82 | -1.41 | -2.69 | -4.57 | -5.34 | -6.57 | -2.69 | -1.41 | -0.82 | 0.32 | 0.35 | 0.29 |

The parallelepiped considered is the one of base the dotted rectangle of Fig.1 and deepness 20Km. This parallelepiped has been divided in \( 7 \times 16 \times 5 = 560 \) small parallelepipeds \( \omega_j \), \( j = 1, 2, \cdots, 560 \) of dimensions 27Km \times 15Km \times 4Km in the z, y, z directions respectively (see Fig.4) and \( j \) runs as shown in Fig.4.

The gravimetric problem that has been solved with the FORTRAN package [10] is the following one: find the values of \( \rho_j \), \( j = 1, 2, \cdots, 560 \) that minimize \( J \) given by (3.2) subject to:
\[ V_i - \epsilon_i \leq \sum_{j=1}^{560} g_{ij} \rho_j \leq V_i + \epsilon_i \quad i = 1, 2, \ldots, 112 \]

\[
\begin{align*}
2.57\text{g/cm}^3 & \leq \rho_j \leq 2.679\text{g/cm}^3 & & 1 \leq j \leq 112 \\
2.57\text{g/cm}^3 & \leq \rho_j \leq 2.672\text{g/cm}^3 & & 113 \leq j \leq 224 \\
2.57\text{g/cm}^3 & \leq \rho_j \leq 2.67\text{g/cm}^3 & & 225 \leq j \leq 560
\end{align*}
\]

where \( \epsilon_i \) and \( V_i = \delta_i + \gamma_i \) are defined as in the two-dimensional case. The results obtained are shown in Fig. 4.

As shown in Fig. 3 for the two-dimensional problem we have found a granite body about 16 km deep with density 2.57 g/cm\(^3\), that is 0.1 g/cm\(^3\) smaller than the surrounding medium, this body is about 30 km long on the surface and 67 km long on the bottom. The results of the three-dimensional study, shown in Fig. 4, confirm the results of Fig. 3. This results are in qualitative agreement with the ones obtained by Ussami and Bot in [6].
REFERENCES


FIGURE CAPTIONS

**Fig. 1** A large region in the state of Bahia (Brazil) is considered between 40-44 degrees of longitude and 10-15 degrees of latitude. The iso-Bouguer anomaly lines are shown. The BB' direction is the one considered in the two-dimensional problem. The dotted rectangle is the region considered in the three-dimensional problem. In the three-dimensional problem measured data have been used in the AA'C'C rectangle, regularized data have been used in the remaining region.

**Fig. 2** The measured residual Bouguer anomaly in the BB' segment and the regularized one is shown.

**Fig. 3** The mass density distribution obtained from the data of Fig.2 solving the linear programming problem (3.3),(3.4) in a vertical section, 20Km deep, along the BB' direction is shown.

**Fig. 4** The mass density distribution obtained from the data of Table 1 solving the linear programming problem (3.2),(3.5) is shown.
Effective residual Bouguer anomaly

Regularized residual anomaly

fig. 2
fig. 3
\[ \Delta x = 27 \text{Km} \]

fig. 4
§5. Appendix 2:
SERIAL AND PARALLEL SOLUTION OF LARGE SCALE LINEAR PROGRAMS BY AUGMENTED LAGRANGIAN SUCCESSIVE OVERRELAXATION,

by

R. De Leone and O. L. Mangasarian

Computer Sciences Technical Report #701

June 1987
Serial and Parallel Solution of Large Scale Linear Programs by Augmented Lagrangian Successive Overrelaxation

R. De Leone & O. L. Mangasarian
Computer Sciences Department
University of Wisconsin
Madison, Wisconsin 53706

Technical Report #701
June 1987

Abstract. Serial and parallel successive overrelaxation (SOR) methods are proposed for the solution of the augmented Lagrangian formulation of the dual of a linear program. With the proposed serial version of the method we have solved linear programs with as many as 125,000 constraints and 500,000 variables in less than 72 hours on a MicroVax II. A parallel implementation of the method was carried out on a Sequent Balance 21000 multiprocessor with speedup efficiency of over 65% for problem sizes of up to 10,000 constraints, 40,000 variables and 1,400,000 nonzero matrix elements.

Key Words: Linear programming, SOR, augmented Lagrangian, parallel algorithms

Abbreviated Title: SOR Solution of Linear Programs

1) This material is based on research supported by National Science Foundation Grants DCR-8420963 and DCR-8521228 and Air Force Office of Scientific Research Grants AFOSR-86-0172 and AFOSR-86-0255.
1. Introduction

In [8, 9, 10, 12] successive overrelation methods are proposed for solving the dual of the problem of finding the least 2-norm solution of a linear program. This leads to an exterior penalty formulation of the dual of the original linear program with the interesting property that the penalty parameter need not approach zero in order to obtain an exact solution of the primal linear program [1, 14]. Thus the penalty parameter need only be less than a certain threshold value in order to obtain an exact solution to the primal linear program. However, the penalty parameter must approach zero in order to obtain a solution to the dual problem. Although this approach has been used effectively in conjunction with successive overrelaxation methods both on serial [10] and parallel machines [11, 15], we propose here the use of an augmented Lagrangian on the dual problem instead of an exterior penalty function in order to alleviate the twin difficulties of determining the threshold value of the penalty parameter required for an exact primal solution, and the need for the penalty parameter to approach zero in order to obtain a dual solution. The first proposal for using an augmented Lagrangian formulation for solving linear programs was made in [22]. In [18] Polyak and Tretiyakov made the remarkable discovery that after a finite number of steps of the augmented Lagrangian algorithm, an exact solution to the primal and dual linear programs is obtained. In [3] Golshstein proposed a projected Gauss-Seidel method in conjunction with an augmented Lagrangian formulation and gave computational results for linear programs with sizes up to 352 variables and 166 constraints. No convergence proofs of the projected Gauss-Seidel method were given in [3], nor of the closely related iterative method of Syrov and Churkredze in [18]. We propose here the use of a projected successive overrelaxation method in conjunction with an augmented Lagrangian formulation. The convergence of the projected SOR method established in [7] is general enough to cover both a serial and a parallel implementation of the method. Since SOR methods are inherently serial in nature, their parallelization is not a routine matter. In [11, 13] two related methods were proposed for the parallelization of SOR methods. The more recent method [13] utilizes an unreduced relaxation factor interval of (0, 2) which we shall employ here with an augmented Lagrangian algorithm for the dual linear program.

The paper is organized as follows. In Section 2 we give the necessary theoretical
background and convergence results for the proposed augmented Lagrangian method applied to the dual linear program. In Section 3 we describe a serial SOR implementation of the method and establish its convergence. In Section 4 we describe our parallel SOR implementation and in Section 5 we present computational results for both the serial and parallel methods.

We briefly describe our notation now. For a vector $z$ in the $n$-dimensional real space $\mathbb{R}^n$, $x_+$ will denote the vector in $\mathbb{R}^n$ with components $(x_+)_i = \max \{x_i, 0\}$, $i = 1, \ldots, n$. The scalar product of two vectors $z$ and $y$ in $\mathbb{R}^n$ will be simply denoted by $zy$. For $1 \leq p \leq \infty$, the $p$-norm $\left(\sum_{i=1}^{n}|z_i|^p\right)^{1/p}$ of a vector in $\mathbb{R}^n$ will be denoted by $\|z\|_p$. For the 2-norm the subscript 2 will be dropped. $\mathbb{R}_+^n$ will denote the nonnegative orthant or the set of points in $\mathbb{R}^n$ with nonnegative components, while $\mathbb{R}_+^{m \times n}$ will denote the set of all $m \times n$ real matrices. For $A \in \mathbb{R}^{m \times n}$, $A^T$ will denote the transpose, $A_i$ will denote the $i$th row, $A_{ij}$ the element in row $i$ and column $j$, and for $I \subseteq \{1, \ldots, n\}$, $J \subseteq \{1, \ldots, n\}$, $A_I$ will denote the submatrix of $A$ with rows $A_i$, $i \in I$, while $A_{IJ}$ will denote the submatrix of $A$ with elements $A_{ij}$, $i \in I$, $j \in J$. Similarly for $z \in \mathbb{R}^n$ and $I \subseteq \{1, \ldots, n\}$, $z_I$ will denote $z_i$, $i \in I$. The set $\{I_1, I_2, \ldots, I_K\}$ is said to be a consecutive partition of $\{1, \ldots, n\}$ if it is a partition of $\{1, \ldots, n\}$ such that $i < j$ for $i \in I_{\ell}$, $j \in I_{\ell+1}$ and $\ell = 1, \ldots, k - 1$. Here and throughout the symbols $:=$ and $\equiv$ denote definition of the term on the left and right sides of each symbol respectively.
2. Theoretical Background

We consider the linear program

\[(2.1) \quad \min cx \text{ subject to } Ax \geq b, x \geq 0\]

where \(c \in \mathbb{R}^n\), \(b \in \mathbb{R}^m\) and \(A \in \mathbb{R}^{m \times n}\) and its dual

\[(2.2) \quad \max_{(u,v) \geq 0} bu \text{ subject to } v = -A^T u + c \geq 0\]

For simplicity we exclude trivial constraints with \(A_i = 0\). In [8, 9] the exterior penalty problem associated with the dual problem (2.2)

\[(2.3) \quad \max_{(u,v) \geq 0} ebu - \frac{1}{2} \|A^T u + v - c\|^2\]

was solved by an SOR procedure for a sufficiently small value of the penalty parameter \(\varepsilon\) to obtain \((u(\varepsilon), v(\varepsilon))\). The unique least 2-norm solution \(\hat{x}\) of the linear program (2.1) was obtained by using the equation

\[(2.4) \quad \hat{x}(\varepsilon) = \frac{1}{\varepsilon} (A^T u(\varepsilon) + v(\varepsilon) - c)\]

which relates an optimal solution \((u(\varepsilon), v(\varepsilon))\) of the dual penalty problem (2.3) and the unique solution \(x(\varepsilon)\) of the corresponding quadratic primal problem [5]

\[(2.5) \quad \min cx + \frac{\varepsilon}{2} x^2 \text{ subject to } Ax \geq b, x \geq 0\]

In particular it follows [14] that the least 2-norm solution \(\hat{x}\) of the linear program (2.1) is related to \(x(\varepsilon)\) of (2.4) by

\[(2.6) \quad \hat{x} = x(\varepsilon) \text{ for } \varepsilon \in (0, \varepsilon] \text{ for some } \varepsilon > 0\]

Thus the penalty parameter \(\varepsilon\) of the dual penalty problem (2.3) is the perturbation parameter of the perturbed primal problem (2.5). In order to avoid possible difficulties associated with determining the threshold value \(\varepsilon\) we consider instead of the exterior penalty problem (2.3) the augmented Lagrangian associated with the dual linear program (2.2)

\[(2.7) \quad L(u, v, x, \gamma) := bu - \frac{1}{2\gamma} \|A^T u + v - c\|^2 - x(A^T u + v - c)\]
It is a standard result [6, 2, 19] that for any \( \gamma > 0 \), a primal-dual solution \((\hat{x}, \hat{u}, \hat{v})\) of (2.1)-(2.2) is equivalent to a stationary point of the following saddle point problem of (2.7):

\[
(2.8) \quad L(u, v, \hat{x}, \gamma) \leq L(\hat{u}, \hat{v}, \hat{x}, \gamma) \leq L(u, v, x, \gamma)
\]

The standard augmented Lagrangian algorithm [2, 19] consists of a maximization step in the \((u, v)\) space \(\mathbb{R}_+^m \times \mathbb{R}_+^n\) followed by an unconstrained gradient descent step in the \(x\) space \(\mathbb{R}^n\). In particular we have the following.

2.1 Augmented Lagrangian Algorithm

Start with any \( z^0 \in \mathbb{R}^n \). Having \( z^t \) determine \( z^{t+1} \) as follows

\[
(2.9) \quad \left\{
\begin{array}{l}
(a) \quad L(u', v', z^t, \gamma_t) = \max_{(u, v) \geq z^0} L(u, v, z^t, \gamma_t) \\
(b) \quad z^{t+1} = z^t - \frac{1}{\gamma_t} \nabla_x L(u', v', z^t, \gamma_t) = z^t + \frac{1}{\gamma_t} (A^T u' + v' - c)
\end{array}
\right.
\]

where \( \{\gamma_t\} \) is any bounded sequence of positive numbers.

For this iterative linear programming algorithm Polyak and Tretiyakov [18] have given the following important finite convergence theorem.

2.2 Augmented Lagrangian Algorithm Finite Termination Theorem [18]

For any bounded positive sequence \( \{\gamma_t\} \) and \( z^0 \in \mathbb{R}^n \), Algorithm 2.1 is finite, that is there exists an integer \( k \) such that \((z^k, u^k, v^k)\) solve the dual linear programs (2.1)-(2.2). Furthermore for each \( z^0 \) there exists a \( \tilde{\gamma} > 0 \) such that for \( 0 < \gamma^0 < \tilde{\gamma} \), the method will terminate in one step, that is \((z^1, u^1, v^1)\) will solve the dual linear programs (2.1)-(2.2).

Note that in the above theorem, two exact maximizations over the \((u, v)\) space \(\mathbb{R}_+^m \times \mathbb{R}_+^n\) are required in order to obtain \((z^{t+1}, u^{t+1}, v^{t+1})\) from \(z^t\).

We note that since by duality theory [5]

\[
(2.10) \quad \max_{(u, v) \geq z^0} L(u, v, x, \gamma) = \min_{z \geq 0} \left\{ cz + \frac{\gamma}{2} \|z - x\|^2 \left| A z \geq b, z \geq 0 \right. \right\} =: \phi(x)
\]

the Augmented Lagrangian Algorithm 2.1 is equivalent to the following gradient method applied to the proximal point function \( \phi(x) \)

\[
(2.11) \quad \begin{align*}
z^{t+1} &= z^t - \frac{1}{\gamma_t} \nabla \phi(z^t) = \text{Prox} \left( z^t \right) \\
\text{where Prox} (z^t) \ &\text{is the solution of} \\
\min \ &\left\{ cz + \frac{\gamma_t}{2} \|z - z^t\|^2 \left| A z \geq b, z \geq 0 \right. \right\}
\end{align*}
\]
Bertsekas [1] and Rockafellar [21] also obtain finite termination for (2.11) from proximal point theory considerations for the linear programming case.

With this background we are prepared now to state and prove the convergence of our serial and parallel SOR algorithms.
3. Serial Successive Overrelaxation Algorithm

The proposed serial algorithm consists of applying the projected SOR method of [7] to the maximization step (2.9a) of the Augmented Lagrangian Algorithm 2.1 for a decreasing sequence of positive numbers \{\gamma^t\}. It follows from the Finite Termination Theorem 2.2 and a theorem of Pang [16, Theorem 3.1], that for any \(z^t\) the projected SOR method will generate a sequence of points converging to an \(z^{t+1}\) that solves the primal linear program (2.1), provided that \(\gamma^t\) is sufficiently small. There are no easily implementable and theoretically justifiable ways of determining how to choose \(\gamma^t\) sufficiently small [20, 4], however we shall prescribe some computationally effective ways for doing that.

The serial projected SOR method has been proposed [7, 9] for solving the quadratic minimization problem

\[
\min \theta(z) := \min_{t \geq 0} \frac{1}{2} z^T M z + q z
\]

where \(M \in \mathbb{R}^{k \times k}\) is symmetric and positive semidefinite. This is precisely problem (2.9a) of Algorithm 2.1 if we make the identifications

\[
M := \frac{1}{\gamma^t} \begin{bmatrix} A & A^T \end{bmatrix}, \quad q := \begin{bmatrix} A(z^t - \frac{c}{\gamma^t}) - b \\ z^t - \frac{c}{\gamma^t} \end{bmatrix}, \quad z := \begin{bmatrix} u \\ v \end{bmatrix}
\]

The projected SOR algorithm consists of the following.

\[
z_j^{t+1} = \left( z_j^t - \omega \nabla^2 \theta(z^t) \right)^{-1} \nabla \theta(z^t, z^{t+1}, \ldots, z^{t+1}, z_j^{t+1})_{+}
\]

\[
\omega \in (0, 2), \quad j = 1, \ldots, k
\]

More specifically for \(\theta(z) = \frac{1}{2} z^T M z + q z\) we have the following.

3.1 Serial SOR Algorithm for \(\min_{z \geq 0} \frac{1}{2} z^T M z + q z\)

Choose \(z^0 \in \mathbb{R}_+^k, \omega \in (0, 2)\). Having \(z^t\) compute \(z^{t+1}\) as follows

\[
z_j^{t+1} = \left( z_j^t - \omega M_j^{-1} \left( \sum_{t \leq j} M_{j, j} z_j^{t+1} + \sum_{t=1}^k M_{j, t} z_t^{t+1} + q_j \right) \right)_+
\]

\[
j = 1, \ldots, k
\]
We are ready to state and establish the convergence of our augmented Lagrangian serial SOR algorithm.

3.2 Augmented Lagrangian Serial SOR Algorithm

Let \( \{ \gamma^t \} \downarrow \gamma \) for some \( \gamma > 0 \), let \( \{ \delta^t \} \downarrow 0 \) and let \( z^0 \in \mathbb{R}^n \). Having \( z^t \) determine \( z^{t+1} \) as follows:

(a) Apply the Serial SOR Algorithm 3.1 to solve (2.9a) with the identifications (3.2), and let \( (u'(t), v'(t)) \) be the \( t \) iterate of this SOR algorithm. Stop if for some \( t = t' \) the following inequality is satisfied.

\begin{equation}
\left| u'(t) \nabla_x L(u'(t), v'(t), z'^t, \gamma^t) \right| + \left| v'(t) \nabla_v L(u'(t), v'(t), z'^t, \gamma^t) \right| + \left\| (\nabla_x L(u'(t), v'(t), z'^t, \gamma^t), -\gamma^t) \right\| \leq \delta^t
\end{equation}

(b) Set \( z^{t+1} = z'(t') \) where

\begin{equation}
z'(t) := z^t + \frac{1}{\gamma^t} (A^T u'(t) + v'(t) - c)
\end{equation}

3.3 Augmented Lagrangian Serial SOR Convergence Theorem

Let \( \{ \gamma^t \} \downarrow \gamma > 0 \) be a sufficiently rapidly decreasing sequence of positive numbers and \( \gamma \) sufficiently small. Then either

(a) For some integer \( k \), the sequence \( \{ z^k(t) \} \) converges to an \( x^* \) that solves the linear program (2.1), or

(b) For each subsequence of \( \{ (z^t, u'(t'), v'(t')) \} \) converging to some \( (x, u, v) \), the corresponding subsequence \( \{ z^{t+1} = z'(t') \} \) converges to an \( x^* \) such that \( x^* \) solves the linear program (2.1). If \( x = x^* \), then \( (u, v) \) solves the dual linear program (2.2).

**Proof**

Either the inequality (3.5) of the algorithm is satisfied at each iteration \( i \) for some \( t' \) or not. Accordingly we have the alternatives (b) and (a) below respectively.

(a) For some iteration \( i = k \) the inequality (3.5) is never satisfied. Hence by Pang's Theorem 3.1 [16], since \( L(u, v, x^k, \gamma^k) \) is by duality theory bounded above for \( (u, v) \geq 0 \) by \( cz + \gamma^k \| x - x^* \|^2 \) for any \( x \geq 0 \) such that \( Ax \geq b \), it follows that the sequence

\begin{equation}
\{ z^k(t) \} := \left\{ z^k + \frac{A^T u^k(t) + v^k(t) - c}{\gamma^k} \right\}, \quad t = 0, 1, 2, \ldots
\end{equation}
where $t$ is the SOR iteration index, converges to a vector $\bar{z}^k$ defined by

\[
\bar{z}^k := z^k + \frac{A^T \hat{u}^k + \hat{\theta}^k - c}{\gamma^k}
\]

for some $(\hat{u}^k, \hat{\theta}^k)$ which solves $\max \limits_{(u,v) \geq 0} L(u,v, z^k, \gamma^k)$. Note however that $\{u^k(t), v^k(t)\}$ need not converge to $(\hat{u}^k, \hat{\theta}^k)$. If $\gamma^k$ is sufficiently small (and this is what is meant by requiring that $\{\gamma^i\}$ decreases sufficiently rapidly) it follows by Theorem 2.2 that $z^{k+1} = \bar{z}^k$, is a solution of the primal linear program (2.1). (Note that $(\hat{u}^k, \hat{\theta}^k)$ need not be a solution of the dual linear program (2.2). To obtain such a dual optimal $(\hat{u}^k, \hat{\theta}^k)$ we need to solve $\min \limits_{(u,v) \geq 0} L((u,v, z^k, \gamma^k)$ exactly. See Corollary 3.5 below.)

(b) If inequality (3.5) holds for each $i$ for some $\ell'$, then since $\{\delta^i\} \downarrow 0$ and $\{\gamma^i\} \downarrow \gamma > 0$, we have that for any subsequence of $\{(x^i, u'(t'), v'(t'))\}$ converging to some $(\bar{x}, \bar{u}, \bar{v})$, the point $(\bar{u}, \bar{v})$ solves the problem $\min \limits_{(u,v) \geq 0} L(u,v, \bar{x}, \bar{v})$ (because $\{\delta^i\} \downarrow 0$), and moreover the corresponding subsequence $\{z^{i+1}\}$ defined by (3.6) and part (b) of Algorithm 3.2 converges to an $\bar{z}$ defined by

\[
\bar{z} := \bar{x} + \frac{A^T \bar{u} + \bar{\theta} - c}{\gamma}
\]

If $\gamma$ is sufficiently small, then it follows by Theorem 2.2 that $\bar{z}$ solves the linear program (2.1). If in addition $\bar{z} = \bar{x}$, then $(\bar{u}, \bar{v})$ is feasible for the dual linear program (2.2) and is also optimal because $b\bar{u} = c\bar{z}$.

It is useful to point out that when the sequence $\{(u'(t), v'(t))\}$ of Algorithm 3.2 is bounded then it has an accumulation point and by Theorem 2.1 [7], each such accumulation point satisfies the optimality conditions for (2.9a). Consequently for each $i$ inequality (3.5) of Algorithm 3.2 is satisfied after a finite number of steps of part (a) of the algorithm. However by Theorem 2(ii) of [9] we have that if the linear program (2.1) satisfies the Slater constraint qualification, then the sequence $\{(u'(t), v'(t))\}$ is indeed bounded. Therefore we have the following.

3.4 Augmented Lagrangian Serial SOR Convergence Corollary

Let the linear program (2.1) satisfy a Slater constraint qualification, that is $A\bar{z} > b$ for some $\bar{z} \geq 0$. Then Theorem 3.3 holds with outcome (b).
Another useful observation follows from the fact [18, Lemma 1], [19] that minimizing the augmented Lagrangian $L(u, v, z, \gamma)$ with an optimal value of $r$ and any $\gamma > 0$ gives a solution to the dual linear program (2.2). Hence we have the following.

3.5 Dual LP SOR Solution Corollary

Under the assumptions of Theorem 3.3 a solution to the dual linear program (2.2) can be obtained for either outcome (a) or (b) of Theorem 3.3 by solving respectively

(a) $\min_{u, v \geq 0} L(u, v, z^k, \gamma^k)$

or

(b) $\min_{u, v \geq 0} L(u, v, z, \gamma)$

We note immediately that we have left open the procedure by which the sequence $\{\gamma^i\}$ is decreased. This is an inherent theoretical difficulty that arises when using inexact minimization in the subproblems of proximal point or augmented Lagrangian algorithms. Thus the approximate minimization criteria of [21, Criteria A, B, A', B'] are not implementable for our problem, while the assumptions of [2, Section 2.5] are not verifiable for our problem. Computationally we have overcome this difficulty by using the following scheme, often used for updating the penalty parameter in augmented Lagrangian algorithm

$$
\gamma^{i+1} = \begin{cases} 
\gamma^i & \text{if } \|z^{i+1} - z^i\| \leq \mu \|z^i - z^{i-1}\|, 0 < \mu < 1 \\
\nu \gamma^i & \text{otherwise}, 0 < \nu < 1
\end{cases}
$$

(3.10)

This scheme works effectively for the solution of very large sparse linear programs as our computational results indicate.
4. Parallel Successive Overrelaxation Algorithm

The key to our parallel SOR algorithm is the use of the parallel gradient projection SOR (GP-SOR) that we proposed in [13] for the solution of (3.1) and which we outline below. Partition the matrix $M$ of (3.1) into $r$ contiguous horizontal blocks as follows:

\begin{equation}
M = \begin{bmatrix}
M_{t_1} \\
M_{t_2} \\
\vdots \\
M_{t_r}
\end{bmatrix}
\end{equation}

where the blocks $M_{t_i}$ correspond to the variables $x_{t_i}$, and $\{I_1, I_2, \ldots, I_r\}$ is a consecutive partition of $\{1, 2, \ldots, k\}$. Now partition $M_{t_i}$ as follows

\begin{equation}
M_{t_i} = \begin{bmatrix} M_{t_i,t_i} & M_{t_i,I_j} \\
M_{I_i,t_i} & M_{I_i,I_j}
\end{bmatrix}
\end{equation}

where $I_j$ is the complement of $I_i$ in $\{1, 2, \ldots, k\}$. Thus $M_{I_i,t_i}$ is a principal square submatrix of $M$ with elements $M_{st}$, $s \in I_i$ and $t \in I_j$. We further partition $M_{t_i,t_i}$ into the sum of its strictly lower triangular part $L_{t_i,t_i}$, its diagonal part $D_{t_i,t_i}$ and its strictly upper triangle part $U_{t_i,t_i}$ as follows

\begin{equation}
M_{t_i,t_i} = L_{t_i,t_i} + D_{t_i,t_i} + U_{t_i,t_i}
\end{equation}

Now define a block diagonal matrix $K$ as follows

\begin{equation}
K = \begin{bmatrix}
L_{t_1,t_1} & L_{t_1,I_j} \\
L_{t_2,t_2} & L_{t_2,I_j} \\
\vdots & \vdots \\
L_{t_r,t_r} & L_{t_r,I_j}
\end{bmatrix}
\end{equation}

where each $L_{t_i,t_i}$ is the strictly lower triangular part of $M_{t_i,t_i}$. An SOR algorithm can now be performed for each row block $I_j$, $j = 1, \ldots, r$, simultaneously, that is in parallel. Note that the block diagonal matrix $K$ replaces the traditional strictly lower triangular matrix of the serial SOR. Specifically we have the following.

4.1 Parallel GP-SOR Algorithm for (3.1)

Let $\{I_1, I_2, \ldots, I_r\}$ be a consecutive partition of $\{1, 2, \ldots, k\}$, let $E$ be a positive diagonal matrix in $R^{k \times k}$ and let $x^0 \geq 0$. For $i = 0, 1, 2, \ldots$, do the following
Direction Generation

Define the direction

\[
\begin{align*}
    d'_i = p(z') - z'_i = \left( \begin{array}{c}
    p_i(z') - z'_i \\
    \vdots \\
    p_i(z') - z'_i 
    \end{array} \right), \\
    \end{align*}
\]

such that \( p(z') \) satisfies

\[
    p_i(z') = \left( z'_i - \omega E_{i,l_i} \left( M_{i,l_i} z^i + q_{i,l_i} + L_{i,l_i} \eta_i (p_{l_i}(z') - z'_i) \right) \right)_+, \quad j = 1, \ldots, r
\]

where \( \omega > 0 \) is chosen such that for some \( \nu > 0 \)

\[
    z_i \left( (\omega E_{i,l_i})^{-1} + L_{i,l_i} \right) z_i \geq \nu \| z_i \|^2, \quad \forall z_i, \quad j = 1, \ldots, r
\]

Stop if \( d'_i = 0 \), else continue.

Stepsize Generation

\[
    z^i+1 = z^i + \lambda d^i
\]

where

\[
    f(z^i + \lambda d^i) = \min_{\lambda} \{ f(z^i + \lambda d^i) | z^i + \lambda d^i \geq 0 \}
\]

4.2 Remark

The principal part of this algorithm consists of the direction generation part (4.3), which can be performed in parallel on \( r \) processors. Once this is done the stepsize generation (4.8) is performed and the new value \( z^i+1 \) is shared between the \( r \) processors.

The following convergence results were derived in [13] for Algorithm 4.1.

4.3 Theorem

(Convergence of the Parallel GP-SOR Algorithm) Let \( M \) be symmetric and positive semidefinite. Either the sequence \( \{z^i\} \) generated by the Parallel GP-SOR Algorithm 4.1 terminates at a solution of (3.1) or each of its accumulation points solves (3.1).

4.4 Corollary

(Parallel GP-SOR special cases) Condition 4.7 of Algorithm 4.1 holds under either of the following two assumptions:

\[
    0 < \omega < \min_{j=1, \ldots, r} \left( \min_{l \in J_i} \frac{2}{\sum_{l \in J_i \setminus \{j\}} |M_{i,l}|} \right).
\]
\[(4.10) \quad 0 < \omega < 2, \quad E = D^{-1} \quad \text{and} \quad M \text{ is positive semidefinite}\]

Our parallel augmented Lagrangian method consists of replacing the Serial SOR Algorithm 3.1 by the Parallel GP-SOR Algorithm 4.1 in Algorithm 3.2 with option (4.10) for the choice of $\omega$ and $E$. We formally state the algorithm below.

4.5 Augmented Lagrangian Parallel SOR Algorithm

Identical to Algorithm 3.2 except that the Serial SOR Algorithm 3.1 in Algorithm 3.2 is replaced by the Parallel GP-SOR Algorithm 4.1 with condition (4.10) above in force.
5. Computational Results

Our algorithms were tested on random linear programs which were generated as follows. First the matrix $A$ of the linear program (2.1) was generated. Each of its nonzero elements was a random number generated by a uniform distribution on the interval $[-100, 100]$. The number of nonzero elements in each row was in accordance to a prescribed density and the random position of each nonzero element was determined according to a uniform distribution on the column indices of the matrix. Next a primal-dual solution vector $(\hat{x}, \hat{u})$ was generated from a uniform distribution on the interval $[0, 10]$ with $50\%$ of the components being nonzero. Finally the vectors $b$ and $c$ were chosen so that $(\hat{x}, \hat{u})$ is optimal.

In Figure 1 we give a summary of our computational results for 6 problems with the number of constraints varying between 25,000 and 125,000 and the number of variables varying between 100,000 and 500,000. All tests were performed on a MicroVax II with 16 megabytes of memory and an expanded disk swap space. We are not aware of any other linear programming software that can handle problems of the size that we have attempted on a comparable machine. MINOS [15], a state-of-the-art pivotal linear programming package, cannot handle any of the problems listed in Figure 1 because they are too big for the machine memory using the MINOS configuration. The largest problem attempted on the MicroVax II with MINOS was a problem with 5000 variables of 20,000 constraints and a matrix density of 0.2% with about 200,000 nonzero elements. MINOS was used with the standard partial pricing and scaling options. After 3150 iterations and 49 hours 54 minutes of machine time, the point was infeasible and the objective function was in error by 59% of the exact minimum. By comparison our Algorithm 3.2 solved the same problem in 1 hour and 4 seconds with a primal-dual objective function accuracy of 7 figures, and relative accuracy of not less than $10^{-9}$ as defined in Figure 1.

The Parallel SOR Algorithm 4.5 was implemented on the Sequent Balance 21000, a multiprocessor that incorporates eight NS32032 processors running at 10MHz, each with a floating point unit, memory management and an 8-kbyte cache sharing a global memory via a 32-bit wide pipelined bus. The machine has 8-Mbytes of physical memory. The operating system DYNIX, is a version of Berkeley 4.2 bsd unix. The computational results are depicted in Figures 2 and 3.
Figure 2 shows the total computing time versus number of processors for four different densities: \( d = 1, 2, 7 \) and 10 percent for a linear program with 1000 constraints and 4000 variables. All problems were solved to a 7-figure accuracy of the primal-dual objective function and relative accuracy better than \( 10^{-7} \) as defined in Figure 1. We observe that the optimal number of processors, that is the one that solves the problem in minimum total time, increases with density as expected. This number is 3 for 1% and 2% densities, 6 for 7% density, and 7 or more for 10% density. This means that for denser problems, a larger number of processors is needed in order to arrive at the shortest solution time. This also means that for denser problems, the communication cost does not become a dominant and hence prohibitive factor until a larger number of processors are used.

In Figure 3 we show results for the case with 10,000 constraints and 40,000 variables, with density of 0.35% and about 1,400,000 nonzero elements. To our knowledge, this is one of the largest linear programs solved on this relatively modest sized multiprocessor. One of the reasons that we were able to solve larger problems on the MicroVax II, is that the latter had twice the total memory size of the Balance 21000 and furthermore, the MicroVax was essentially a single-user machine dedicated to the serial SOR algorithm. The optimal number of processors for the low-density case shown in Figure 3 is 4. Just as in the cases of Figure 2, the optimal number of processors should increase with problem density.

We conclude with some observations on the speedup efficiency of our Parallel SOR Algorithm 4.5. We define the speedup efficiency \( E(r) \) as the ratio of the actual to the theoretical speedup of the algorithm using \( r \) processors instead of 1 processor, thus

\[
E(r) := \frac{T(1)}{rT(r)}
\]

where \( T(r) \) is the total time for solving a given problem using \( r \) parallel processors. Figure 4 shows the speedup efficiencies for a typical case of a linear program with 1000 constraints, 4000 variables and 2% density. The reason why some efficiencies are over 100% was pointed out in [13]. The explanation is that our Parallel SOR Algorithm 4.5 changes with the number of processors used, because the matrix \( K \) defined by (4.4) changes with the number of blocks into which \( M \) is divided. Thus we are not comparing identical algorithms when we evaluate the ratio \( T(1)/rT(r) \) of (5.1). Nevertheless the expression is a valid measure of efficiency in the sense of comparing the theoretical reduced time \( T(1)/r \) to the observed time \( T(r) \) for an algorithm with a variable \( K \) that depends on the partition.
of $M$. If the matrix $K$ is held fixed for $r = 1$ and $r > 1$, then we obtain efficiencies for identical algorithms, and they would all be less than 100%. This was demonstrated in [13]. Nevertheless the present efficiencies of over 100% are indeed very encouraging and also give the additional and somewhat surprising result that a serial implementation of our $r$-block Parallel SOR Algorithm 4.5 on a single machine, will give for some $r$ a better computing time than the single block Serial SOR Algorithm 3.2. For the specific case of Figure 4, a serial implementation of the $r$-block parallel SOR with $r=2, 3, 4$ and 5 will be faster than the single block serial SOR.
<table>
<thead>
<tr>
<th>No.</th>
<th>Prob. constr. $m \times 10^{-3}$</th>
<th>No. var. $n \times 10^{-2}$</th>
<th>Nonzero elements per row $\times 10^{-1}$</th>
<th>Total No. Nonzero elements</th>
<th>Time hr:min</th>
<th>Fig. Accur.</th>
<th>Obj. Func. Accur.</th>
<th>Log$_{10}$ Rel. Accur.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25</td>
<td>100</td>
<td>20</td>
<td>5</td>
<td>309</td>
<td>2:31</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>120</td>
<td>25</td>
<td>7.5</td>
<td>392</td>
<td>4:34</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>160</td>
<td>20</td>
<td>8</td>
<td>363</td>
<td>4:56</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>200</td>
<td>16</td>
<td>8</td>
<td>525</td>
<td>7:47</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>400</td>
<td>12</td>
<td>12</td>
<td>967</td>
<td>22:35</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>725</td>
<td>500</td>
<td>9</td>
<td>11.25</td>
<td>3100</td>
<td>71:40</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

(a) Number of correct figures in primal objective

(b) Number of correct figures in dual objective

(c) $\frac{\|(-Az + b)\|_\infty}{\|b\|_\infty}$ (Relative Accuracy)

(d) $\frac{\|A^T(u - c)\|_\infty}{\|(-c)\|_\infty}$ (Relative Accuracy)

(e) $\frac{|cz - cz|}{|cz + cz|}$, $z$: exact, $z$: computed (Relative Accuracy)

(f) $\frac{|bu - b\delta|}{|bu + b\delta|}$, $u$: exact, $u$: computed (Relative Accuracy)

Fig. 1. MICROVAX II: Serial SOR Algorithm 3.2 test results.
Fig. 2. BALANCE 21000: Total time for Parallel SOR Algorithm 4.5 to solve linear program versus number of processors for various densities \( d \). (Average of 4 randomly generated cases with 1000 constraints and 4000 variables.)
Fig. 3. BALANCE 21000: Total time for Parallel SOR Algorithm 4.5 to solve linear program versus number of processors. ($d =$ density, $m =$ number of constraints, $n =$ number of variables.)
Density \( d = 2\% \), \( m = 1000 \), \( n = 4000 \)

<table>
<thead>
<tr>
<th>No. Processes ( r )</th>
<th>Time Sec. ( T(r) )</th>
<th>Speedup Efficiency ( E(r) = T(1)/rT(r) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3946</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>709</td>
<td>278%</td>
</tr>
<tr>
<td>3</td>
<td>638</td>
<td>206%</td>
</tr>
<tr>
<td>4</td>
<td>716</td>
<td>138%</td>
</tr>
<tr>
<td>5</td>
<td>711</td>
<td>111%</td>
</tr>
<tr>
<td>6</td>
<td>840</td>
<td>78%</td>
</tr>
<tr>
<td>7</td>
<td>850</td>
<td>66%</td>
</tr>
</tbody>
</table>

Fig. 4. BALANCE 21000: Speedup efficiency \( E(r) \) for the Parallel SOR Algorithm 4.3 for an LP with 1000 constraints and 4000 variables and 2% density.
References


