Journal of Global Optimization **6**: 153–161, 1995.

The Linear Complementarity Problem as a Separable Bilinear Program*

O.L. MANGASARIAN
Computer Sciences Department, University of Wisconsin, 1210 West Dayton Street, Madison, WI 53706, U.S.A. (email: olvi@cs.wisc.edu)

(Received: 29 July 1994; accepted: 4 October 1994)

Abstract. The nonmonotone linear complementarity problem (LCP) is formulated as a bilinear program with separable constraints and an objective function that minimizes a natural error residual for the LCP. A linear-programming-based algorithm applied to the bilinear program terminates in a finite number of steps at a solution or stationary point of the problem. The bilinear algorithm solved 80 consecutive cases of the LCP formulation of the knapsack feasibility problem ranging in size between 10 and 3000, with almost constant average number of major iterations equal to four.

Key words: Nonmonotone linear complementarity, bilinear program, knapsack problem.

1. Introduction

It is well known that the linear complementarity problem [4, 16]

$$0 \leqslant x \perp Mx + q \geqslant 0 \tag{1}$$

for a given $n \times n$ real matrix M and a given $n \times 1$ vector q, can be written as the bilinear program

$$\min_{x,w}\{x'w | w = Mx + q, x \geqslant 0, w \geqslant 0\}. \tag{2}$$

For the case of a general M, considered here, the objective function of (2) is nonconvex and the constraints are inseparable in the variables x and w. Thus for a general M, the bilinear program (2) with inseparable constraints is not suitable for solving the LCP. An interesting and somewhat curious exception is the monotone case when M is positive semidefinite. For this case $x'w$ is a convex function in (x, w) on the feasible region of (2) [11, Proposition 1], and an effective bilinear algorithm can be prescribed [11, Section 4]. When M is positive definite, an exterior penalty function proposed [5] for (2) on $\{(x, w)|x \geqslant 0, w \geqslant 0\}$ is strongly convex in (x, w) for sufficiently large values of the penalty parameter. This also leads to an alternative computational approach for this case.

Recently there have been successful attempts for solving bilinear programs with separable constraints. In [1], separable bilinear programs were formulated for the

* This material is based on research supported by Air Force Office of Scientific Research Grant F49620-94-1-0036 and National Science Foundation Grants CCR-9322479 and CDA-9024618.

NP-complete bilinear separability problem, where 140 consecutive instances of the problem were solved. In [19] another NP-complete robotic problem is cast and solved as a separable bilinear program. Other bilinear techniques are given in [7, Chapter 9] and references cited therein. In this work we cast the NP-complete [2] general LCP as the following separable bilinear program.

$$\min_{x,r,s}\{(r' + s'M)x + q's \mid Mx + q \geqslant 0, x \geqslant 0, r + s = e, r, s \geqslant 0\}. \tag{3}$$

We note that the objective function is bilinear in x and (r, s) and that the constraints are separable in the variables x and (r, s). Hence, simple finitely terminating methods of bilinear programming, such as those of [1], are applicable for obtaining either an exact solution of (3) or a stationary point. Of course, we do not expect this bilinear formulation to be capable of processing every LCP, but it is interesting to note that it can indeed solve nonmonotone LCP's. For example, this bilinear scheme solved 80 consecutive nonmonotone LCP's associated with the knapsack feasibility problem [3, 2] without failure and with as many as 3000 variables.

We briefly outline the paper now. In Section 2 we establish equivalence of two formulations, (3) and (5), to the general LCP (1). We give a simple bilinear algorithm, Algorithm 2.3, based on (5) and establish its finite termination to a solution of the LCP or to a stationary point in Theorem 2.4. In Remark 2.6 we note that the objective function of the bilinear program (3), when minimized with respect to the (r, s) variables, yields the objective function of (5), which turns out to be a classical natural concave error residual of the LCP [18, 8, 12]. In Section 3 we report on numerical testing of the Bilinear Algorithm 2.3. We tested it on the NP-complete nonmonotone LCP formulation of knapsack feasibility problems of size 10 to 3000. For each problem size, ten instances were solved. A total of eighty consecutive problems were solved without failure. Average major iteration number did not grow with problem size, but stayed near a constant mean of 4 iterations. Total solution time appears to grow polynomially in problem size rather than exponentially, as indicated by a leveling off of the concave-looking log-time versus size plot depicted in Figure 2. Section 4 ends the paper with some conclusions and open questions.

A word about our notation. The feasible region of the LCP (1) is the set $\{x \mid Mx + q \geqslant 0, x \geqslant 0\}$. The scalar product of two vectors x and y in the n-dimensional real space will be denoted by $x'y$ in conformity with MATLAB [14] notation. For a linear program $\min_{x\in X} c'x$ with a vertex solution, the notation

$$\textit{arg vertex partial} \min_{x\in X} c'x$$

will denote any vertex of the feasible region X, typically obtained by any desired number of steps of the simplex method. For $x \in R^n$, the norm $\|x\|$ will denote the 2-norm, $(x'x)^{1/2}$ while $\|x\|_1$ will denote the 1-norm, $\sum_{i=1}^{n} |x_i|$. The notation $\min\{x, y\}$ applied to vectors x and y in R^n will denote a vector with components that are minima of corresponding components of x and y. For $x \in R^n, (x_+)_i =$

$\max\{0, x_i\}, i = 1, \ldots, n$. For an $m \times n$ matrix A, A_i will denote the ith row of A. The step function, step(x), applied to $x \in R^n$ is defined as a binary vector in R^n of zeros and ones, with ones corresponding to positive components of x. The identity matrix in a real space of arbitrary dimension will be denoted by I, while a column vector of ones of arbitrary dimension will be denoted by e.

2. Separable Bilinear Formulation of the LCP

We begin by establishing the validity of the bilinear program (3) as an equivalent formulation of the general LCP.

PROPOSITION 2.1. *The point x in R^n solves the LCP (1) if and only if x and some r, s in R^n solve the bilinear program (3) with a zero minimum.*

Proof. Necessity. Let x solve the LCP (1). Define r and s as follows:

$$(1 - s_i) = r_i = \begin{cases} 1 & \textit{for} \quad M_i x + q_i > 0 \\ 1 & \textit{for} \quad x_i = 0, M_i x + q_i = 0 \\ 0 & \textit{for} \quad x_i > 0 \end{cases}$$

It immediately follows, that x, r, s satisfy the constraints of (3) and render zero its objective which is nonnegative on the feasible region.

Sufficiency. Let x, r, x solve (3) with a zero minimum. Then $0 = r'x + s'(Mx + q)$. Since $r + s = e > 0$, it follows that $x'(Mx + q) = 0$, and x solves the LCP (1). □

We note that Proposition 2.1 also follows from [10, Theorem 1]. We further note that, for a fixed x satisfying the constraints of (3), the objective of (3) is minimized over $r, s \geqslant 0, r + s = e$, by taking

$$s = s(x) = \mathit{step}((I - M)x - q), \quad r = r(x) = e - s(x). \tag{4}$$

This leads to the following alternative characterization of an LCP solution as a consequence of Proposition 2.1.

PROPOSITION 2.2. *The point x in R^n solves the LCP (1) if and only if*

$$\min_x \{(e + (M' - I)s(x))'x + q's(x) | Mx + q \geqslant 0, x \geqslant 0\} = 0, \tag{5}$$

where $s(x)$ is defined in (4).

We note immediately that, in the objective function of (5), the minimization over r and s prescribed in (3) has been carried out and hence the bilinear algorithm [1, Algorithm 2.1] simplifies to the following, where the minimization over r and s has been carried out as indicated by (4).

ALGORITHM 2.3. ***Bilinear Algorithm.*** *Start with any feasible x^0 for (5). Determine x^{i+1} from x^i such that*

$$x^{i+1} \in \textit{arg vertex partial} \min_x \{(e + (M' - I)s(x^i))'x \mid Mx + q \geqslant 0, x \geqslant 0\} \tag{6}$$

where $s(x)$ is defined in (4) and such that

$$(e + (M' - I)s(x^i))'(x^{i+1} - x^i) < 0.$$

Finite termination of the Bilinear Algorithm 2.3 follows directly from [1, Theorem 2.1] as shown below.

THEOREM 2.4. *For a feasible LCP (1), the sequence $\{x^i\}$ of the Bilinear Algorithm 2.3 is well defined and terminates at a solution of the LCP or at a stationary point $\bar{x}$ satisfying the following necessary optimality conditions of (5) for $\bar{x}$ and some $\bar{y} \in R^n$:*

$$\begin{aligned} &M\bar{x} + q \geqslant 0, \bar{x} \geqslant 0, -M'\bar{y} + (M' - I)s(\bar{x}) + e \geqslant 0, \bar{y} \geqslant 0 \\ &\bar{y}'(M\bar{x} + q) + \bar{x}'(-M'\bar{y} + (M' - I)s(\bar{x}) + e) = 0, \end{aligned} \tag{7}$$

where $s(\bar{x})$ is defined in (4). The stationary point $\bar{x}$ solves the LCP (1) if and only if $\bar{y} = s(\bar{x})$.

Proof. The first part of the theorem is a direct consequence of [1, Theorem 2.1] where the minimization over (r, s) in (3) has been carried out resulting in the objective of (5) that does not depend on (r, s). To obtain the last part of the theorem, note that the conditions (7) are satisfied by $\bar{y} = s(\bar{x})$, when $\bar{x}$ is feasible for the LCP (7) and that

$$s(\bar{x})'(M\bar{x} + q) + \bar{x}'(-s(\bar{x}) + e) = 0.$$

Since $s(\bar{x}) + (-s(\bar{x}) + e) = e > 0$, it follows that $\bar{x}'(M\bar{x} + q) = 0$, and $\bar{x}$ solves the LCP (1). □

REMARK 2.5. The objective function of (5) is a concave function of x. This follows from that fact that it is obtained by minimizing the objective $(r' + s'M)x + q's$ of (3) over $r, s \geqslant 0, r + s = e$, and the latter objective is linear in r and s [9, Theorem 1]. This, of course, makes problem (5) a difficult concave minimization problem.

REMARK 2.6. By using the definitions (4) of $s(x)$ and $r(x)$ in the objective function of (5), the latter problem can be rewritten as

$$\min_x \left\{ \sum_{M_i x + q_i \geqslant x_i} x_i + \sum_{M_i x + q_i < x_i} M_i x + q_i \mid Mx + q \geqslant 0, x \geqslant 0 \right\}. \tag{8}$$

This problem can be rewritten as

$$\min_x\{\|\min\{x, Mx+q\}\|_1 | Mx+q \geqslant 0, x \geqslant 0\}. \tag{9}$$

The objective function of (9) is a classical natural residual of the linear complementarity problem [18, 8, 12] and constitutes a local error bound for any LCP, and a global error bound for strongly monotone M. It is also a global error bound for $M \in R_0$, the class of matrices M such that 0 is the only solution to $M_x \geqslant 0, x \geqslant 0, x'Mx = 0$. Hence, the objective of (8) or (9) bounds a constant multiple of the distance to the solution set of the LCP for these cases. We note again that the objective function of (9) is concave in x because it is the sum of n terms, each of which is the minimum of two linear functions.

We give now some computational results.

3. Computational Results

Numerical testing of the Bilinear Algorithm 2.3 was carried out with MATLAB [14] on a DECstation 5000/125 using MINOS 5.4 [15]. Since there is no guarantee that the algorithm will terminate at a global solution, random restarts (major iterations) in the (r, s) space were carried out. Each restart consisted of taking $r(x^0)$ as a random vertex of the unit cube $\{r|0 \leqslant r \leqslant e\}$ and $s(x^0) = e - r(x^0)$. For a sparse nonmonotone LCP test problem, we chose the LCP formulation of the knapsack feasibility problem, also known as the subset sum problem [13]. The problem consists of finding an n-dimensional binary vector z such that

$$a'z = b, \tag{10}$$

where a is a given $n \times 1$ vector of positive integers, and b is a positive integer. The equivalent LCP [3] is given by

$$M = \begin{bmatrix} -I & 0 & 0 \\ e' & -n & 0 \\ -e' & 0 & -n \end{bmatrix}, \qquad q = \begin{bmatrix} a \\ -b \\ b \end{bmatrix}. \tag{11}$$

We note that this formulation is slightly different from that of [2] wherein $-n$ of (11) is replaced by -1. The formulation (11), unlike that of [2] where M is indefinite, ensures that M is negative definite. However the two formulations are essentially the same and are related by a simple change of variable of the components x_{n+1} and x_{n+2}. An interesting symmetric LCP formulation of the knapsack feasibility problem is given in [6]. The knapsack solution z is obtained from the LCP solution $x \in R^{n+2}$ by the relation

$$z_i = \frac{x_i}{a_i}, \quad i = 1, \ldots, n. \tag{12}$$

The matrix M of (11) is negative definite, as can be easily deduced by the Gersgorin Circle Theorem [17, Theorem 3.2.1] applied to $(M + M')/2$ and the fact that $(M + M')/2$ is nonsingular.

TABLE I. Summary of results for 80 problems

	n	# restarts	# lp's (last restart)	time sec. per problem
min	10	1	1	0.1
mean		4.4	1.1	0.5
max		15	2	1.7
min	100	1	1	0.4
mean		5.7	1.2	3.4
max		15	2	7.4
min	500	1	1	6.7
mean		3.8	1.6	40.4
max		8	3	69.0
min	1000	1	1	12.3
mean		3.2	1.3	146.0
max		7	2	378.0
min	1500	1	1	22.0
mean		2.4	1.1	247.7
max		6	2	953.4
min	2000	1	1	19.2
mean		4.4	1.3	706.6
max		15	2	2971.2
min	2500	1	1	80.7
mean		2.8	1.5	711.8
max		8	3	1564.3
min	3000	1	1	145.5
mean		3.9	1.7	1252.6
max		10	2	3161.8

Eighty consecutive knapsack feasibility problems (10) were solved by the Bilinear Algorithm 2.3 with $10 \leqq n \leqq 3000$. The vector a had random integer components uniformly distributed in [0, 10]. A random binary vector x in R^n was generated and b was determined from $b = a'x$. Ten problems for each of $n = 10$, 100, 500, 1000, 1500, 2000, 2500 and 3000 were solved. The results are summarized in Table I. For each n, the minimum, mean and maximum, over 10 problems, of the following quantities are given: the number of restarts (major iterations), number of linear programs solved after the last restart and total time. Maximum number of restarts permitted was 30 (actual maximum number of restarts needed was 15) and maximum number of linear programs per restart permitted was 8 (actual maximum number of linear programs needed in each last restart was 3). It is interesting to note that the number of restarts does not grow with problem size,

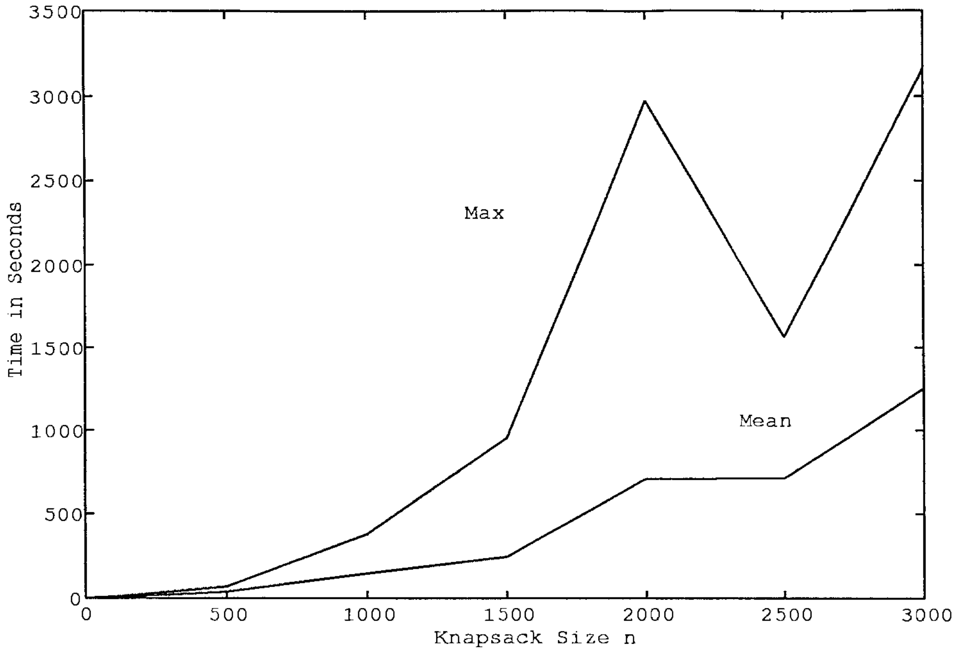

Fig. 1. Mean and maximum times versus knapsack size n.

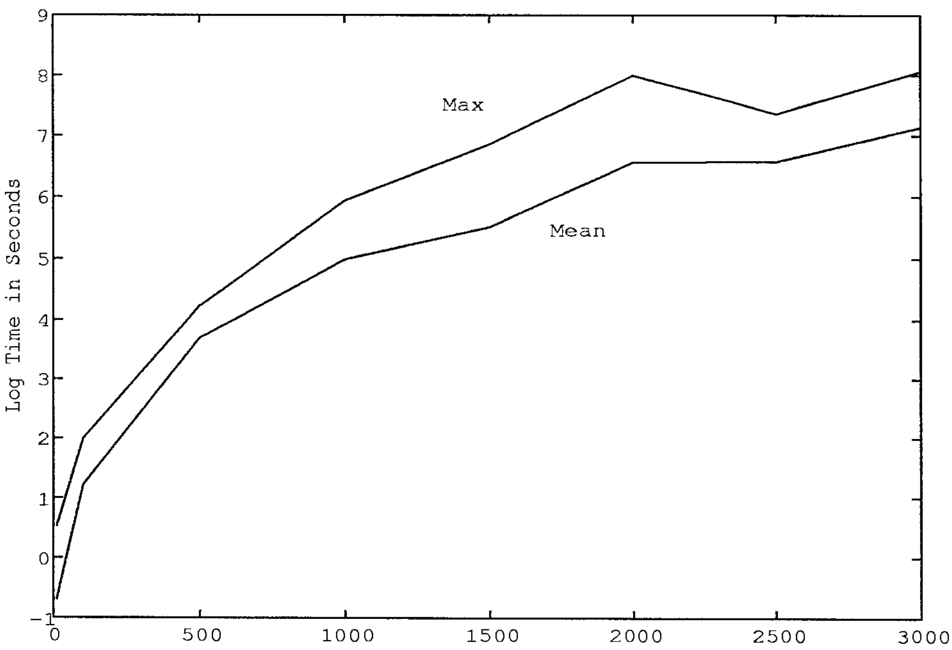

Fig. 2. Mean and maximum log times versus knapsack size n.

and in fact it decreases sometimes. For example, the maximum of the mean number of restarts is 5.7 for n=100 while the same mean is 3.9 for n=3000. The number of linear programs solved per last restart is fairly small, never more than 3 and average less than 2. Finally, the time to solve each problem is given both in Table I and Figures 1 and 2. Figure 1 plots time versus problem size n while Figure 2 plots logarithm of time to the base e versus problem size n. Both figures, and especially Figure 2, indicate a leveling off of the time versus problem size. The leveling off of the log plot may be indicative of polynomial time growth with problem size.

4. Conclusion

We have presented a separable bilinear programming approach for solving the nonmonotone LCP. Computational results on a class of nonmonotone LCP's are encouraging and warrant further study of the approach. Interesting open questions are: What classes of LCP's is the bilinear approach best suited for? What modifications are needed to further speed the bilinear approach and make it applicable for wider classes of LCP's? For what LCP's are the stationarity conditions (7) sufficient for solvability of the LCP?

Acknowledgement

I am grateful to my colleague Michael C. Ferris for making available his MINOS interface for MATLAB, and to my Ph.D. student Chunhui Chen for reading this paper and making valuable suggestions.

References

1. K. P. Bennett and O. L. Mangasarian (1993), Bilinear separation of two sets in n-space, *Computational Optimization & Applications* **2**: 207–227.
2. S.-J. Chung (1989), NP-completeness of the linear complementarity problem, *Journal of Optimization Theory and Applications* **60**: 393–399.
3. S.-J. Chung and K. G. Murty (1981), Polynomially bounded ellipsoid algorithms for convex quadratic programming, in O. L. Mangasarian, R. R. Meyer, and S. M. Robinson (eds), *Nonlinear Programming 4*, pages 439–485, New York, Academic Press.
4. R. W. Cotle, J.-S. Pang, and R. E. Stone (1992). *The Linear Complementarity Problem*, Academic Press, New York.
5. J. T. J. Van Eijndhoven (1986), Solving the linear complementarity problem in circuit simulation. *SIAM Journal on Control and Optimization*, **24**: 1050–1062.
6. R. Horst, P. Pardalos, and N. V. Thoai (1994), *Introduction to Global Optimization*. Kluwer Academic Publishers, Dordrecht, Netherlands.
7. R. Horst and H. Tuy (1993), *Global Optimization*, Springer-Verlag, Berlin. (Second, Revised Edition).
8. Z.-Q. Luo and P. Tseng (1992), Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem. *SIAM Journal on Optimization*, **2**: 43–54.
9. O. L. Mangasarian (1964), Nonlinear programming problems with stochastic objective functions. *Management Science*, **10**: 353–359.
10. O. L. Mangasarian (1979), Simplified characterization of linear complementarity problems solvable as linear programs. *Mathematics of Operations Research*, **4**: 268–273.

11. O. L. Mangasarian and J.-S. Pang (1993), The extended linear complementarity problem. *SIAM Journal on Matrix Analysis and Applications* **16(2)**, January 1995.
12. O. L. Mangasarian and J. Ren (1994), New improved error bounds for the linear complementarity problem, *Mathematical Programming* **66**, 1994, 241–255.
13. S. Martello and P. Toth (1987), Algorithms for knapsack problems, in S. Martello, G. Laporte, M. Minoux, and C. Ribeiro (eds.), *Surveys in Combinatorial Optimization*, Annals of Discrete Mathematics 31, North-Holland, Amsterdam.
14. MathWorks, inc. (1991), *PRO-MATLAB for UNIX Computers*, The MathWorks, Inc., South Natick, MA 01760.
15. B. A. Murtagh and M. A. Saunders (1983), MINOS 5.0 user's guide. Technical Report SOL 83.20, Stanford University, December 1983. MINOS 5.4 Release Notes, December 1992.
16. K. G. Murty (1988), *Linear Complementarity, Linear and Nonlinear Programming*, Helderman-Verlag, Berlin.
17. J. M. Ortega (1972), *Numerical Analysis, A Second Course*. Academic Press.
18. J.-S. Pang (1986), Inexact Newton methods for the nonlinear complementarity problem. *Mathematical Programming*, **36(1)**: 54–71.
19. J.-S. Pang, J. C. Trinkle, and G. Lo (1994), A complementarity approach to a quasistatic rigid body motion problem. Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, MD 21218.