Mathematical Programming 19 (1980) 213–219.
North-Holland Publishing Company

COMPUTATIONAL COMPLEXITY OF PARAMETRIC LINEAR PROGRAMMING*

Katta G. MURTY

The University of Michigan, Ann Arbor, MI, U.S.A.

Received 11 December 1978
Revised manuscript received 10 December 1979

We establish that in the worst case, the computational effort required for solving a parametric linear program is not bounded above by a polynomial in the size of the problem.

Key words: Parametric Linear Program, Computational Complexity, Worst Case Behavior, Exponential Growth.

1. Introduction

We consider the parametric linear program

$$\begin{aligned} f(\lambda) = \text{minimum value of } & \theta(x) = cx, \\ \text{subject to} \qquad & Ax = b + \lambda b^*, \quad x \geqq 0 \end{aligned} \tag{1}$$

where A is a matrix of order $m \times p$ and rank m, and λ is a real valued parameter. It may be required to solve this problem for all real values of λ, or for all values of λ in some specified interval of the real line.

$f(\lambda)$ is a piecewise linear convex function in λ. The simplex algorithm for this parametric linear program partitions the real line into intervals, each interval being the optimality interval of a feasible basis for (1), and the algorithm moves from one interval to the next by making a single dual simplex pivot step. In each interval the slope of $f(\lambda)$ is constant. In consecutive intervals obtained during the algorithm, the slope of $f(\lambda)$ remains the same if the algorithm moves from one of these intervals to the next without making any nondegenerate dual simplex pivot steps; otherwise the slopes of $f(\lambda)$ in these intervals are different. See Chapters 11, 12 in [1], Chapter 8 in [2] and Chapter 7 in [3].

Let $\phi(A, b, b^*, c)$ denote the total number of intervals on the real line, each of positive length, such that the slope of $f(\lambda)$ is a constant in each interval, and the slopes of $f(\lambda)$ in any two intervals are different. Since each of these intervals have to be separately obtained (as the slopes of $f(\lambda)$ in them are different) when (1) is solved, $\phi(A, b, b^*, c)$ provides a lower bound on the computational effort required for solving (1) for all real values of λ by any algorithm. Clearly, the

* This research is partially supported by Air Force Office of Scientific Research, Air Force Number AFOSR-78-3646.

parametric right-hand side simplex algorithm for solving (1), requires at least $\phi(A, b, b^*, c) - 1$ pivot steps before termination. So we can use $\phi(A, b, b^*, c)$ as a measure of the computational complexity of problem (1).

The interesting question is whether $\phi(A, b, b^*, c)$ remains bounded above by a polynomial in p, the size of the parametric linear program (1), irrespective of what the data in A, b, b^*, c may be. We investigate this question here. We construct a class of parametric linear programs. The nth problem in the class has $m = n$, $p = 2n$, for $n \geqq 2$, and we show that the value of $\phi(A, b, b^*, c)$ for it is 2^n. This conclusively establishes that in the worst case, the computational effort required for solving the parametric linear program (1), is not bounded above by any polynomial in the size of the problem.

2. The class of problems

In the nth problem in our class, the number of constraints is n, and the number of variables is $2n$. For the sake of convenience, we will denote the variables in the problem by the symbols $w_1, \dots, w_n$; $z_1, \dots, z_n$. I_n is the identity matrix of order n. Let $\tilde{M}(n)$ be the lower triangular matrix of order n defined by $\tilde{m}_{ii} = 1$ for all $i = 1$ to n, $\tilde{m}_{ij} = 0$ for all $j > i$, and $\tilde{m}_{ij} = 2$ for all $j < i$. This matrix was constructed in [4] to study the computational complexity of complementary pivot algorithms. Let

$$A_n = (I_n \vdots -\tilde{M}_n), \quad b(n) = (2^n, 2^{n-1}, \dots, 2),$$
$$b^*(n) = (-1, -1, \dots, -1) \in R^n,$$
$$c(n) = (0, \dots, 0, 4^{n-1}, 4^{n-2}, \dots, 4, 1) \in R^{2n}.$$

Then the nth problem in our class is

$$\begin{aligned} &\text{minimize} && \theta(w, z) = c(n) \begin{pmatrix} w \\ \cdots \\ z \end{pmatrix}, \\ &\text{subject to} && A_n \begin{pmatrix} w \\ \cdots \\ z \end{pmatrix} = b(n) + \lambda b^*(n), \\ & && w = (w_1, \dots, w_n) \geqq 0, \qquad z = (z_1, \dots, z_n) \geqq 0. \end{aligned} \tag{2}$$

We will call the pair of variables (w_j, z_j) as the jth *complementary pair of variables*, and each variable in this pair is the *complement* of the other. A *complementary vector of variables* for this problem is an ordered vector of variables $(y_1, \dots, y_n)$ where $y_j \in \{w_j, z_j\}$ for each $j = 1$ to n. Thus there are 2^n complementary vectors of variables, and it can be verified, using the results in [4], that every complementary vector of variables is a basic vector for (2). $A(n)$,

$b(n)$, $b^*(n)$, $c(n)$ is the data in the nth problem in our class. As discussed above, the computational complexity of the nth problem in our class is $\phi(A_n, b(n), b^*(n), c(n))$.

3. The sequence of complementary basic vectors

In our proofs, we will encounter the complementary basic vectors for the nth problem in our class, arranged in a specific order, as a sequence, and this order is the same as that obtained in the proof of theorem 1 in [4]. Here we describe how to generate this sequence. For $n = 2$, the specific order is

$$(w_1, w_2),\ (w_1, z_2),\ (z_1, z_2),\ (z_1, w_2). \tag{3}$$

Let $a(s) = 2^s - 1$, for any $s \geqq 2$. To get the order for the nth problem in the class, let $v_0, v_1, \dots, v_{a(n-1)}$ be the ordered sequence of complementary basic vectors for the $(n-1)$th problem when the variables in it are treated as $w_2, \dots, w_n$; $z_2, \dots, z_n$ (instead of $w_1, \dots, w_{n-1}$; $z_1, \dots, z_{n-1}$). Then the specific order of complementary basic vectors for the nth problem is

$$\begin{gathered} V_0 = (w_1, v_0), \quad (w_1, v_1), \dots, (w_1, v_{a(n-1)}), \\ (z_1, v_{a(n-1)}), \quad (z_1, v_{a(n-1)-1}), \dots, (z_1, v_0) = V_{a(n)}. \end{gathered}$$

4. A numerical example

Consider problem (2) with $n = 3$. To solve this, the parametric simplex algorithm begins with (w_1, w_2, w_3) as the unique optimum feasible basic vector corresponding to $\lambda = 0$. It can be verified that it partitions the parameter space into $2^3 = 8$ optimality intervals, and requires $2^3 - 1 = 7$ pivots in all. The relative cost coefficients of the nonbasic variables will be strictly positive in each of the canonical tableaus obtained, and this implies that in the interior of each optimality interval, the optimum basis obtained is unique. Each pivot step made in the algorithm is a nondegenerate dual simplex pivot step, and hence, after every pivot step, the slope of the optimum objective value strictly increases. So $\phi(A_3, b(3), b^*(3), c(3)) = 8 = 2^3$. Also it can be verified that the sequence of optimum basic vectors obtained, is exactly the complementary vectors of variables for this problem, in the specific order discussed above, which is the same as the order of basic vectors in table 2 of [4].

5. The main result

Theorem. *The following results hold when the nth parametric linear program in the class is solved, beginning with $(w_1, \dots, w_n)$ as the unique optimum basic vector corresponding to $\lambda = 0$, for $n \geqq 2$.*

(i) *The sequence of optimum basic vectors obtained in the parametric algorithm is exactly the complementary vectors of variables for this problem, in the specific order discussed above.*

(ii) *The optimality intervals obtained are* $[-\infty, 2]$, $[2,4]$, $[4,6], \ldots, [2^{n+1}-4, 2^{n+1}-2]$, $[2^{n+1}-2, \infty]$. *As the algorithm moves from one interval to the next in this sequence, the slope of the optimum objective value increases strictly.*

(iii) *The relative cost efficients of all the nonbasic variables are strictly positive in every tableau obtained during the algorithm.*

(iv) *In the interior of each optimality interval, the optimum feasible basis obtained is primal and dual nondegenerate and is the unique optimum feasible basis for the problem.*

(v) *The algorithm goes through* $2^n - 1$ *pivot steps before termination.*

Proof. The statement of the theorem can be verified to be true for $n = 2$. Proof is by induction on n. We now set up an induction hypothesis.

Induction hypothesis: The statement of the theorem holds for the $(n-1)$th parametric linear program in our class.

Under the induction hypothesis, we will now prove that the statement of the theorem must also hold for the nth problem in our class, (2). When the first row, and the columns of w_1, z_1 are eliminated from (2), it can be verified that it becomes the $(n-1)$th problem in our class, with the exception that the variables are called $w_2, \ldots, w_n$; $z_2, \ldots, z_n$ (instead $w_1, \ldots, w_{n-1}$, $z_1, \ldots, z_{n-1}$). Call this the *principal subproblem* of the nth problem in our class. Let $v_0, v_1, \ldots, v_{a(n-1)}$ be the ordered sequence of complementary basic vectors for this problem with these names for the variables.

In (2), any pivots performed in the columns of $w_2, \ldots, w_n$, $z_2, \ldots, z_n$ do not change Row 1. Also $2^n - \lambda$ remains strictly positive for all $\lambda < 2^n$. From this it is clear that when the nth problem in our class is solved beginning with $(w_1, \ldots, w_n)$ as the initial basic vector, w_1 remains in the basic vector until the value of λ reaches 2^n. Applying the induction hypothesis to the principal subproblem, we see that the entering and leaving variables in the pivot steps that occur in solving the nth problem in our class will be the same as those needed when the principal subproblem is solved, until λ reaches the value $2^n - 2$. By the induction hypothesis this requires $2^{n-1} - 1$ pivot steps, and at the end of these steps we reach the basic vector $(w_1, z_2, w_3, \ldots, w_n)$. By the induction hypothesis applied to the principal subproblem, and the arguments listed, above, the sequence of basic vectors obtained before reaching this basic vector is

$$V_0 = (w_1, v_0), \quad V_1 = (w_1, v_1), \ldots, V_{a(n-1)} = (w_1, v_{a(n-1)}).$$

and statement (iii), (iv) of the theorem hold for each of these basic vectors.

The canonical tableau for (2) with respect to the basic vector $(w_1, z_2, w_3, \ldots, w_n)$ can be obtained by performing one pivot step in (2) with the column of z_2 as the pivot column and row 2 as the pivot row. It can be verified that this basic vector is optimal in the interval $2^n - 2 \leqq \lambda \leqq 2^n$, it is primal and dual nondegenerate and is the unique optimum basic vector in the interior of this interval. To continue the parametric algorithm for $\lambda > 2^n$, we have to make a dual simplex pivot step in Row 1 of this canonical tableau and it can be verified that the column vector of z_1 will be the pivot column. This leads us to the canonical tableau, with respect to the basic vector $(z_1, z_2, w_3, \ldots, w_n)$, which is given in Table 1.

From Table 1 it can be verified that the basic vector $V_{2^{n-1}} = (z_1, V_{a(n-1)}) = (z_1, z_2, w_3, \ldots, w_n)$ is optimal in the interval $2^n \leqq \lambda \leqq 2^n + 2$. The relative cost coefficients associated with all the nonbasic variables are strictly positive in Table 1, and the basic vector $V_{2^{n-1}}$ is primal nondegenerate when λ is the interior of the interval $[2^n, 2^n + 2]$. The updated right-hand side constant in the first row in table 1 is $-2^n + \lambda$ and this remains strictly positive for all $\lambda > 2^n + 2$. So when we continue the solution of the nth problem in our class from Table 1 (for $\lambda > 2^n + 2$), we will never have to choose Row 1 as the pivot row again. Calling $\lambda - 2^n = \nu$, and eliminating Row 1 and the columns of w_1, z_1 from Table 1, we verify that it leads again to the $(n-1)$th problem in the class, with the exception that the variables are now called $z_2, w_3, \ldots, w_n$; $w_2, z_3, \ldots, z_n$, in that order, and the parameter is ν ($\lambda \geqq 2^n + 2$ corresponds to $\nu \geqq 2$). Call this the *principal subproblem of Table* 1. Using the induction hypothesis on this principal subproblem of Table 1, and the above facts, we conclude that when the solution of the nth problem in our class is continued from Table 1 for $\lambda > 2^n + 2$, the basic variable changes that occur are exactly the same as those that will occur when this principal subproblem of Table 1 is solved, and hence it will lead through the following basic vectors:

$$V_{2^{n-1}+1} = (z_1, \boldsymbol{v}_{2^{n-1}-2}), \quad V_{2^{n-1}+2} = (z_1, \boldsymbol{v}_{2^{n-1}-3}), \ldots, V_{2^n-1} = (z_1, \boldsymbol{v}_0).$$

By the induction hypothesis, the statements (iii), (iv) of the theorem continue to hold.

Table 1
Canonical tableau for (2) with $(z_1, z_2, w_3, \ldots, w_n)$ as the basic vector

Basic variables	w_1	w_2	w_3	…	w_n	z_1	z_2	z_3	…	z_n	$-\theta$	
z_1	-1	0	0		0	1	0	0		0	0	$-2^n + \lambda$
z_2	2	-1	0	…	0	0	1	0	…	0	0	$2^{n-1} - (\lambda - 2^n)$
w_3	2	-2	1	…	0	0	0	-1	…	0	0	$2^{n-2} - (\lambda - 2^n)$
⋮	⋮	⋮	⋮		⋮	⋮	⋮	⋮		⋮	⋮	⋮
w_n	2	-2	0	⋯	1	0	0	-2	⋯	-1	0	$2 - (\lambda - 2^n)$
$-\theta$	$2 \times 4^{n-2}$	4^{n-2}	0	…	0	0	0	4^{n-3}	⋯	1	1	

Also, (iii) implies that after each pivot step in this algorithm, the slope of the optimum objective value strictly increases.

These facts imply that under the induction hypothesis, the statements of the theorem hold for the nth problem in the class.

The theorem has already been verified from $n = 2$. Hence it is true for all $n \geqq 2$.

Corollary 1. $\phi(A_n, b(n), b^*(n), c(n)) = 2^n$.

Proof. By the above theorem, when the nth problem in our class is solved by the parametric right hand side simplex algorithm, it partitions the parameter space into 2^n optimality intervals. In each of these 2^n optimality intervals obtained for the nth problem, the optimum solution is unique; and in the interior of that interval the optimum basis obtained is primal nondegenerate, and is the unique optimum basis. The slope of the optimum objective value strictly increases as we move from one interval to the next one on its right. Hence $\phi(A_n, b(n), b^*(n), c(n))$ is the number of optimality intervals into which the parameter space is partitioned when the nth problem in our class is solved by the parametric right hand side simplex algorithm for all values of λ, and hence is equal to 2^n. Thus the complete answer the nth parametric LP in our class is itself exponentially long.

Corollary 2. *In the worst case, the computational effort needed to completely solve the parametric LP* (1), *is not bounded above by any polynomial in the number of variables in the problem, p.*

Proof. As discussed above, $\phi(A, b, b^*, c)$ is a measure of the computational complexity of (1). The nth problem in our class has $m = n$, $p = 2n$, and has data A_n, $b(n)$, $b^*(n)$, $c(n)$. By Corollary 1, $\phi(A_n, b(n), b^*(n), c(n)) = 2^n$, and this grows exponentially with p, for $n \geqq 2$, and hence is not bounded above by any polynomial in p. Thus the class of parametric LPs constructed above, clearly establishes this corollary.

Note. Since each of the basic vectors obtained when the nth problem in our class is solved, is both primal and dual nondegenerate in the interior of its optimality interval, by taking the class of dual problems of the problems constructed above, we conclude that the same results hold for parametric cost linear programs.

We have established the exponential growth phenomenon of the computational requirements, in parametric linear programs which are required to be solved for all real values of the parameter λ. Consider parametric linear programs which are required to be solved only for values of the parameter λ in some specified finite interval, say the interval [0, 1]. Even on these problems, the worst case computational requirements grow exponentially or faster with p. A class of parametric linear programs exhibiting this fact can be constructed from the class of parametric linear-programs discussed above, by proper scaling of the data.

References

[1] G.B. Dantzig, *Linear programming and extensions* (Princeton University Press, Princeton, NJ, 1963).

[2] S.I. Gass, *Linear programming: Methods and applications*, 4th Edition (McGraw-Hill, New York, 1975).

[3] K.G. Murty, *Linear and combinatorial programming* (Wiley, New York, 1976).

[4] K.G. Murty, "Computational complexity of complementary pivot methods", *Mathematical Programming Study* 7 (1978) 61–73.