Sonderdruck aus / Reprinted from:

METHODS OF OPERATIONS RESEARCH

49

Hain Verlag bei
athenäum

A NOTE ON THE GENERALITY OF THE SELF-DUAL ALGORITHM WITH VARIOUS STARTING POINTS*

NIMROD MEGIDDO, Palo Alto, California, U.S.A.

We show how the selection of various starting points for the self-dual simplex method for linear programming can reproduce seemingly different algorithms, sometimes referred to as "variable dimension" or "constraint-by-constraint" methods and their hybrids.

1.

Consider the linear programming problem in the form:

$$\begin{aligned} \text{Maximize} \quad & c^T x \\ \text{subject to} \quad & Ax \leq b \\ & x \geq 0 \quad , \end{aligned}$$

(where $x, c \in R^n$, $A \in R^{m \times n}$ and $b \in R^m$). The *self-dual simplex algorithm* is attributed to Dantzig [D]. Lemke [L] generalized it and in its full generality it is sometimes referred to as "Lemke's algorithm". It can be explained as follows. Suppose that $c^0 \in R^n$ is any *positive* vector and also let $b^0 \in R^m$ be any positive vector. For any nonnegative λ, let $P(\lambda)$ denote the following linear programming problem:

$$\begin{aligned} \text{Maximize} \quad & (c - \lambda c^0)^T x \\ \text{subject to} \quad & Ax \leq b + \lambda b^0 \\ & x \geq 0 \quad . \end{aligned}$$

For λ sufficiently large, the zero-vector is a basic optimal solution. Starting from such a sufficiently large λ, we can continuously decrease the value of this parameter, always maintaining a basic optimal solution of the problem $P(\lambda)$. Under certain nondegeneracy conditions, the path-following procedure will find a basic optimal solution of the original problem $P(0)$, if and only if every problem along the way has an optimal solution.

The self-dual algorithm has recently become a focus of attention in the context of probabilistic analysis. This began with the work of Borgwardt [Bo1, Bo2] and Smale [Sm1, Sm2]. Adler and Megiddo [AM1, AM2], Todd [T] have proven that with special parameterizing vectors (and relative to certain classes of probability distributions) this algorithm runs on the average in no more than $O(\min(m, n))^2)$ steps (see also [AKS2]).

In this note we are interested in special types of parameterizing vectors b^0 and c^0. In the following discussion these vectors will themselves be functions of a single parameter, ϵ, and we will be interested only in sufficiently small values of ϵ. Assume that the vectors $b^0(\epsilon)$ and $c^0(\epsilon)$ are such that each of their components is equal to a certain power of ϵ. For example, in one case considered later we have $b_i^0(\epsilon) = \epsilon^i$ and $c_j^0(\epsilon) = \epsilon^{m+j}$. Under suitable non-degeneracy conditions, for every

This work was supported in part by the National Science Foundation under Grants MCS-8300984, ECS-8121741 and ECS-8218181.

ϵ there is a well-defined sequence of bases, namely, the sequence of bases generated by the self-dual algorithm while employing the parameterizing vectors $b^0(\epsilon)$ and $c^0(\epsilon)$. Considering the progress of the self-dual algorithm as a function of ϵ, we discover that the critical values of ϵ, that is, values at which the sequence of bases changes, are all roots of finitely-many polynomials in ϵ. Thus, the number of intervals of ϵ, over which the sequence of bases does not change, is finite. Hence, there is some positive ϵ^*, such that for every ϵ, $0 < \epsilon < \epsilon^*$, the algorithm produces the same sequence of bases while employing the parameterizing vectors $b^0(\epsilon)$ and $c^0(\epsilon)$. Whenever we refer to ϵ as "sufficiently small" we simply mean that ϵ belongs to the particular interval whose left-hand open end is the zero.

We now argue that by suitably choosing the powers of ϵ, which constitute the components of $b^0(\epsilon)$ and $c^0(\epsilon)$, the sequence of bases produced by the self-dual algorithm (when ϵ is sufficiently small) can be made indentical to those produced by seemingly different algorithms, sometimes referred to as "variable dimension" or "constraint-by-constraint" methods. The ideas of a row at a time or a column at a time have been exploited since the 1960's. See for example [G] where degeneracies are handled with such an approach. Other related works are Bland's least-index rules [Bl], [v], [Bo2], [Sa] and [AKS1].

It is interesting to mention that the change of starting point can be replaced by scaling rows and columns, that is, multiplying them by suitable powers of ϵ. Thus, Dantzig's original self-dual algorithm can simulate many algorithms provided we first scale the problem in a suitable way.

2.

Consider first the choice $c_j^0(\epsilon) = \epsilon^j$ $(j = 1, \cdots, n)$ and $b_i^0(\epsilon) = \epsilon^{n+i}$ $(i = 1, \cdots, m)$. To gain some more insight into the problem, consider the following sequence of values of λ:

$$\lambda_k = \frac{1}{\epsilon^{k+0.5}} \quad (k = 0, \cdots, m+n) \quad .$$

Suppose $n \leq k \leq m+n$. Obviously, for every j $(j = 1, \cdots, n)$, the objective-function of the problem $P(\lambda_k)$ is determined by the vector whose components are $c_j - \epsilon^{j-k-0.5}$ $(j = 1, \cdots, n)$. The direction of this vector (that is, the unit-vector in this direction) tends to the direction of $(1, 0, \cdots, 0)^T \in R^n$ as ϵ tends to zero. More accurately, this direction is asymptotically equal to the direction of $c^0(\epsilon)$ when ϵ tends to zero. Analogously, consider the values of the right-hand side vector $b + \lambda b^0$ when $\lambda = \lambda_k$ and $n \leq k \leq m+n$. These are equal to $b_i + \epsilon^{n+i-k-0.5}$. Suppose $i \leq k - n$. For ϵ sufficiently small, such a constraint is inactive since its right-hand side tends to infinity. On the other hand, if $i \geq k - n + 1$ then the right-hand side tends to b_i as ϵ tends to zero. It follows that the basis, which is maintained by the algorithm at the point where $\lambda = \lambda_k$, is in fact an optimal one for the following problem:

$$\begin{aligned} \text{Maximize} \quad & (-c^0)^T x \\ \text{subject to} \quad & A_i x \leq b_i \quad (i = k - n + 1, \cdots, m) \\ & x \geq 0 \quad , \end{aligned}$$

where A_i is the i-th row of A. Moreover, we can now interpret what the algorithm actually does when λ varies between λ_k and λ_{k-1}. First, it is obvious that in this segment the algorithm starts with an optimal solution relative to the problem just stated and pivots until it reaches an optimal solution for the same problem with $k - 1$ replacing k. Thus, the self-dual algorithm starts with

only the nonnegativity requirements present and then adds the constraints one-by-one, in order of decreasing i. A more careful look at the particular way, by which the algorithm brings the next constraint, reveals the following. Let λ satisfy $\lambda_{k-1} \leq \lambda \leq \lambda_k$. For any such λ, the algorithm finds an optimal basis for the following problem:

$$\begin{aligned} \text{Maximize} \quad & (-c^0)^T x \\ \text{subject to} \quad & A_{k-n}x \leq b_{k-n} + \lambda\epsilon^k \\ & A_i x \leq b_i \quad (i = k-n+1, \cdots, m) \\ & x \geq 0 \quad . \end{aligned}$$

When λ varies between λ_k and λ_{k-1}, the value of $\lambda\epsilon^k$ varies over an interval which tends to $(0, \infty)$ as ϵ tends to zero. Of course, if the $(k-n)$-th constraint is satisfied when $\lambda = \lambda_k$ then the algorithm does not pivot at all in this segment. Otherwise, it starts the present segment with $(k-n)$-th constraint totally relaxed and proceeds by bringing it in continuously in the sense that the quantity $A_{k-n}x - b_{k-n}$ decreases continuously to zero (unless the algorithm discovers that the problem is not feasible). This description is equivalent to saying that the algorithm behaves like a parametric objective-function simplex algorithm solving the following problem:

$$\begin{aligned} \text{Maximize} \quad & t(b_{n-k} - A_{n-k}x) - (c^0)^T x \\ \text{subject to} \quad & A_i x \leq b_i \quad (i = k-n+1, \cdots, m) \\ & x \geq 0 \quad , \end{aligned}$$

where t varies from 0 to ∞.

If the algorithm does not stop before λ reaches the value λ_n then it means that all the constraints have been brought in successfully, so that the problem is feasible. Moreover, at this point the algorithm has a basic optimal solution relative to the objective-function determined by $-c^0$, subject to all the constraints of the original problem. It is obvious that from this point and on the algorithm behaves like a parametric objective-function simplex algorithm, solving the following problem:

$$\begin{aligned} \text{Maximize} \quad & (tc - c^0)^T x \\ \text{subject to} \quad & Ax \leq b \\ & x \geq 0 \quad , \end{aligned}$$

where t varies from 0 to ∞. It follows that the particular choice of powers of ϵ (with ϵ sufficiently small) makes the self-dual algorithm work precisely like the "constraint-by-constraint" method in the second work of Adler, Karp and Shamir [AKS2]. Some other variants of constraint-by-constraint (see [AKS1]) can also be simulated by the self-dual method with appropriate choices of the starting point.

3.

Another approach, which can be interpreted as dual to the constraint-by-constraint method, is that of "variable-dimension". Here we look at the projection of the problem on the subspace of the last j variables, letting j grow from 1 to n. An equivalent way of looking at this situation is that we force the variables $x_1, \cdots, x_{n-j}$ to be equal to zero, which can be implemented by parametrizing with an objective-function that assigns a very large negative profit to these variables. Suppose we assign the powers of ϵ in a way converse to the previous one, namely, $c_j^0(\epsilon) = \epsilon^{m+j}$

$(j = 1, \cdots, n)$ and $b_i^0(\epsilon) = \epsilon^i$ $(i = 1, \cdots, m)$. Like in the previous case, when λ varies between $\epsilon^{-(k+0.5)}$ and $\epsilon^{-(k-0.5)}$ (where $m \leq k \leq m+n$), the algorithm solves the following problem:

$$\begin{aligned} &\text{Maximize} && c^T x \\ &\text{subject to} && Ax \leq b^0(\epsilon) \\ & && x_1, \cdots, x_{k-m} = 0, \quad x_{k-m+1}, \cdots, x_n \geq 0 \quad . \end{aligned}$$

Having increased the dimension to n, the algorithm then works like a parametric right-hand side simplex algorithm, deforming the vector $b^0(\epsilon)$ into the original b.

4.

It is interesting to note that the algorithm, with a special initialization procedure, described by Borgwardt [Bo1, Bo2], can also be related to the self-dual algorithm as we explain in this section. This algorithm solves problems of the form:

$$\begin{aligned} &\text{Maximize} && c^T x \\ &\text{subject to} && Ax \leq e \end{aligned}$$

(where $x, c \in R^n$, $A \in R^{m \times n}$ and $e = (1, \cdots, 1)^T \in R^m$). These problems have the advantage that the zero-vector satisfies all the constraints and, trivially, they are also feasible relative to the projection of the problem on any space of a subset of variables. To start the argument, let us first consider a related problem where in addition to the original constraints the variables must also be nonnegative. Now, consider the self-dual algorithm with parameterizing vectors $c^0(\epsilon)$, with $c_j^0(\epsilon) = \epsilon^{n+1-j}$ and $b^0(\epsilon)$, with $b_i^0(\epsilon) = \epsilon^{n+1}$. Like in the preceding algorithm, the dimension is increasing (in the reverse order of coordinates, in order to fit Borgwardt's original description), obtaining optimal solutions relative to projections of the problem, until an optimal solution for the original problem is found. In Borgwardt's formulation there are no nonnegativity constraints. In order to fit that version we may use the standard trick of representing an unconstrained variable x_j as the difference between two nonnegative variables $x_j = x_j' - x_j''$, associating the same power of ϵ with both of them, so that they are introduced to the problem at the same stage. The difference bewteen this problem and the one in the previous section is that, due to the positivity of the given right-hand side vector, feasibilty is already maintained at the start.

5.

In general, if the self-dual algorithm is started at a vector whose components are pairwise distinct powers of ϵ (with ϵ sufficiently small) then it behaves like a hybrid of "constraint-by-constraint" and "variable dimension" methods. It starts with all the variables set to zero and all the constraints (except for the nonnegativity requirements) relaxed. It then brings in the constraints one-by-one and adds the variables one-by-one; the specific order of these $m+n$ steps is determined by the powers of ϵ. Thus, the algorithm "solves" subproblems obtained from the original one by omitting some constraints and setting some of the variables to zero. We note that some of these subproblems may be unbounded, in which cases the algorithm terminates the current stage with an appropriate ray rather than an optimal solution. Note that a constraint is brought in by moving the hyperplane parallel to itself. The movement starts from a very far location depending on the value of ϵ. In the case when a new constraint changes the subproblem from unbounded to bounded then, at

the start of the movement, the algorithm holds an optimal solution to a problem obtained from the unbounded one by adding the hyperplane at its initial position. The algorithm pivots from a terminal basis of one subproblem to a terminal basis of the next subproblem in the sequence, behaving during this segment either like a parametric objective-function simplex algorithm (when it brings in a new constraint) or like a parametric right-hand side simplex algorithm (when it increases the dimension). We note again that the original self-dual algorithm (starting at $(1, \cdots, 1)^T$) can simulate the implementation of Lemke's algorithm with any positive starting point.

We remark that it follows from the analysis in [AM1] that under a certain class of probability distributions (see [AM2]) the expected number of pivot steps performed by the self-dual algorithm, corresponding to *any* such starting point (consisting of distinct powers of ϵ), is $O((\min(m, n))^2)$.

References

[AKS1] I. Adler, R. M. Karp and R. Shamir, "A family of simplex variants solving an $m \times d$ linear program in expected number of pivot steps depending on d only", December 1983.

[AKS2] I. Adler, R. M. Karp and R. Shamir, "A simplex variant solving an $m \times d$ linear program in $O(\min(m^2, n^2))$ expected number of pivot steps", December 1983.

[AM1] I. Adler and N. Megiddo, "A simplex-type algorithm solves linear programs of order $m \times n$ in only $O((\min(m, n))^2)$ steps on the average", preliminary report, November 1983.

[AM2] I. Adler and N. Megiddo, "A simplex algorithm whose average number of steps is bounded between two quadratic functions of the smaller dimension", Proceedings of the 16th Annual ACM Symposioum on Theory of Computing (1984), ACM, New York, 1984, pp. 312-323.

[Bl] R. G. Bland, "New finite pivoting rules for the simplex method", *Mathematics of Operations Research* **2** (1977), 103-107.

[Bo1] K.-H. Borgwardt, "Some distribution-independent results about the asymptotic order of the average number of pivot steps of the simplex method", *Mathematics of Operations Research* **7** (1982), 441-462.

[Bo2] K.-H. Borgwardt, "The average number of steps required by the simplex method is polynomial", *Zeitschrift fur Operations Research* **26** (1982) 157-177.

[D] G. B. Dantzig, *Linear programming and extensions*, Princeton University Press, Princeton, New Jersey, 1963.

[G] G. W. Graves, "A complete constructive algorithm for the general mixed linear programming problem", *Naval Research Logistics Quarterly* **12** (1965) 1-34.

[L] C. E. Lemke, "Bimatrix equilibrium points and mathematical programming", *Management Science* **11** (1965) 681-689.

[Sa] R. Saigal, "A homotopy for solving large, sparse and structured fixed point problems", *Mathematics of Operations Research* **8** (1983) 557-578.

[Sm1] S. Smale, "On the average number of steps of the simplex method of linear programming", *Mathematical Programming* **27** (1983) 241-262.

[Sm2] S. Smale, "The problem of the average speed of the simplex method", in *Mathematical Programming: The State of the Art*, A. Bachem, M. Grotschel and B. Korte, eds, Springer-Verlag, 1983, pp. 530-539.

[T] M. J. Todd, "Polynomial expected behaviour of a pivoting algorithm for linear complementarity and linear programming problems", Technical Report No. 595, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York, November 1983.

[v] L. van der Heyden, "A variable dimension algorithm for the linear complementarity problem", *Mathematical Programming* **19** (1980) 328-346.

DR. NIMROD MEGIDDO, IBM Research, 5600 Cottle Road, San José, CA 94193, U.S.A.