A COMPETITIVE (DUAL) SIMPLEX METHOD FOR THE ASSIGNMENT PROBLEM

M.L. BALINSKI

CNRS, Laboratoire d'Econométrie de l'Ecole Polytechnique, Paris, and Department of Applied Mathematics and Statistics, SUNY, Stony Brook, NY, USA

Received 13 August 1984
Revised manuscript received 24 May 1985

Dedicated with affection to George B. Dantzig on the occasion of his seventieth birthday.

"Where there is abundance of mystery and confusion in every direction, the truth seldom remains hidden for long. It's a matter of having plenty of angles to go at it from. Only the utterly simple crimes - the simplex crimes, you may say - have the trick of remaining baffling." - Sir John (from Michael Innes, The Open House (A Sir John Appleby Mystery), Penguin Books, 1974).

A dual simplex method for the assignment problem leaves open to choice the activity \((i,j)\) of row \(i\) and column \(j\) that is to be dropped in pivoting so long as \(x_{0} < 0\). A choice \((i,j)\) over columns \(j\) having at least 3 basic activities that minimizes \(x_{j}\) is shown to converge in at most \(2^{1}/2\) pivots, and at most \(O(n^{3})\) time, and it is argued that on average the number of pivots is at most \(n \log n\).

Key words: Assignment Problem, Dual Method, Signature, Linear Programming, Simplex Method, Pivoting, Average Behavior.

1. Preliminaries

An assignment problem is defined by any \(n\) by \(n\) matrix \(c = (c_{ij})\) of real numbers: find a permutation \(\sigma\) of the column indices that minimizes \(\sum_{i} c_{i\sigma(i)}\). As a linear program the problem is to

\[
\text{minimize } c \cdot x = \sum_{i,j} c_{ij}x_{ij}
\]

for \(x \in P_{n} = \{x_{ij}: \sum_{i} x_{ij} = 1, \sum_{j} x_{ij} = 1, x_{ij} \geq 0, i \in I, j \in J\}\)

where \(I\) represents the set of row indices and \(J\) the set of column indices, \(|I| = |J| = n\).

The combinatorial assignment problem may be formulated and solved as a linear program because the extreme points of \(P_{n}\), the primal polyhedron, are integer valued and so correspond to permutations \(\sigma\) via the definition: \(x_{ij} = 1\) implies \(\sigma(1) = j\).

The arguments used in this paper are cast in terms of the following well known model. Let the set \(I\) of \(n\) nodes represent the rows of the matrix \(c\) or \(x\) (in the figures they are drawn as circles), the set of \(n\) nodes \(J\) the columns (in the figures.
they are drawn as squares), and let $T$ be any spanning tree of edges $(i, j)$, $i \in I$, $j \in J$ (see Fig. 1). $T$ contains exactly $2n - 1$ edges. Let $\tilde{P}_n$ be $P_n$ without the inequality constraints $x_{ij} \geq 0$. A spanning tree $T$ uniquely determines a basic solution $x$ of $\tilde{P}_n$, which I will call $x(T)$, defined by \{\(x \in \tilde{P}: x_{ij} = 0 \text{ for } (i, j) \notin T\}. It is integer valued and easily computed recursively (the values of $x(T)$ for $T$ of Fig. 1 are attached to the respective edges of $T$). A basic solution is feasible if $x_{ij} \geq 0$ for all $i, j$.

Fig. 1. Tree $T$, values $x(T)$.

The dual to the assignment problem is to

\[
\text{maximize} \quad \sum_i u_i + \sum_j v_j \\
\text{for} \quad (u, v) \in D_n = \{u_i, v_j: u_i + v_j \leq c_{ij}, u_i = 0, i \in I, j \in J\}. \tag{2}
\]

A spanning tree $T$ of our model uniquely determines a basic solution $u, v$ of $D_n$, called $u(T), v(T)$, defined by \{\(u, v\): $u_i = 0$, $u_i + v_j = c_{ij}$ for $(i, j) \in T\}. It is also easily computed recursively. The basic solution is feasible if $w_{ij} = c_{ij} - u_i - v_j \geq 0$ for all $i, j$. Since each $u, v$ gives rise to a unique $w = (w_{ij})$ we may also refer to $w(T)$, the matrix of $w_{ij}$'s arising from $T$. If $c$ is all integer then for basic solutions $u, v$ and $w$ are also all integer.

Given any $T$, $x(T)$ and $w(T)$ are orthogonal or satisfy the 'complementary slackness' property

\[
x_{ij}(T) \cdot w_{ij}(T) = 0 \quad \text{for all } i, j. \tag{3}
\]

**Lemma 1.** If $x(T)$ and $w(T)$ are feasible then $x(T)$ solves (1) and $w(T)$ solves (2).

This is the 'easy' part of the duality theorem.

A spanning tree $T$ is, in linear programming terms, simply a basis. Given a spanning tree $T$ a pivot step consists in deleting an edge $(k, l)$ from $T$ and adjoining an edge $(g, h)$ from not $T$ to form a new spanning tree $T'$, and in 'updating' the values of the variables, that is, finding the uniquely corresponding values $x' = x(T')$ and $w' = w(T')$. This-exchange or pivot is again easily computed.

Consider first the $x$ values. $T \cup (g, h)$ contains a unique cycle that includes $(g, h)$ and $(k, l)$. Edge $(k, l)$ is 'minus', and for every pair of adjacent edges in the cycle, one is 'minus', the other 'plus'. Note that $g \in T^l$ and $h \in T^k$. Letting $x_{kl} = \varepsilon$, the rule is: $x'_{ij} = x_{ij} - \varepsilon$, $(i, j)$ 'minus'; $x'_{ij} = x_{ij} + \varepsilon$, $(i, j)$ 'plus'; $x'_{ij} = x_{ij}$ otherwise (see Fig. 2).