On simple combinatorial optimization problems

A.J. Hoffman

Department of Mathematical Sciences, IBM Research Division, T.J. Watson Research Center, P.O. Box 218, Yorktown Height, NY 10598, USA

Received 3 January 1992
Revised 4 March 1992

Abstract


We characterize (0, 1) linear programming matrices for which a greedy algorithm and its dual solve certain covering and packing problems. Special cases are shortest path and minimum spanning tree algorithms.

1. Introduction

Two of the best known, conceptually simple and computationally easy combinatorial optimization problems are: to find the shortest path from a node s to a node t in a directed graph with nonnegative edge lengths; and to find a minimum spanning tree in a graph (more generally, a minimum rooted spanning arborescence in a directed graph). We announce a general theorem which includes as special cases the well-known algorithms for solving these problems. The theorem will also include as special cases the algorithm for finding a maximum flow in a series parallel graph [4], an optimum coloring of an interval graph, and all the algorithms for the problems described in the opening sections of [3]. For a survey of related material, see [2].

2. Sequentially greedy matrices

Let $A$ be a (0, 1) matrix with $m$ rows and $n$ columns for which each column has at least one 1. We consider, for a given $b \geq 0$, the problem

$$\max \sum x_j : x > 0, \quad Ax < b$$

Correspondence to: A.J. Hoffman, Department of Mathematical Sciences, IBM Research Division, T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, USA

0012-365X/92/$05.00 \textcopyright 1992 — Elsevier Science Publishers B.V. All rights reserved
and its dual

$$\min \sum y_i b_i ; y \geq 0, \quad y' A \geq 1.$$  \hspace{1cm} (2.2)

The sequentially greedy (SG) algorithm for solving (2.1) can be informally summarized as follows. Let $b_{i(1)} = \min \{ b_k : a_{k1} = 1 \}$. Set $\tilde{x}_i = b_{i(1)}$. Subtract $\tilde{x}_i$ from all $b_k$ such that $a_{k1} = 1$. Delete from $A$ row $i(1)$ and all columns $j$ such that $a_{i1}, j = 1$. Proceed inductively.

This process will produce a set of chosen columns $\hat{C} = \{ j(1) = 1, j(2), \ldots, j(k) \}$ and chosen rows $\hat{R} = \{ i(1), i(2), \ldots, i(k) \}$ such that the submatrix $A(\hat{R}, \hat{C})$ formed by them is (essentially) triangular; and such that the vector $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_n)$ given by

$$\tilde{x}_j = \begin{cases} \tilde{x}_{j(t)} & \text{if } j = j(t), t = 1, \ldots, k, \\ 0 & \text{otherwise} \end{cases}$$

is feasible for (2.1). We shall characterize those $A$ such that, for all $b \geq 0$, SG produces $\tilde{x}$ which is optimum.

The dual sequential greedy algorithm (DSG) is obtained by solving $\tilde{z} A(\hat{R}, \hat{C}) = \tilde{1}$, and setting

$$\tilde{y}_i = \begin{cases} \tilde{z}_i & \text{if } i \text{ is chosen,} \\ 0 & \text{otherwise.} \end{cases}$$

We shall characterize those $A$ such that, for all $b \geq 0$, DSG produces $\tilde{y} = (\tilde{y}_1, \ldots, \tilde{y}_n)$ which is feasible (hence optimum for (2.2), by linear programming duality). The main theorem is the following.

**Theorem.** The following conditions on $A$ are equivalent:

(2.3) for all $b \geq 0$, SG solves (2.1);

(2.4) for all $b > 0$, DSG solves (2.2);

(2.5) if $A$ contains a submatrix

$$\begin{bmatrix} j_1 & j_2 & j_3 \\ i_1 & 1 & 1 & 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} j_1 & j_2 & j_3 \\ i_1 & 1 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} j_1 & j_2 & j_3 \\ i_2 & 1 & 0 & 1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} j_1 & j_2 & j_3 \\ i_2 & 1 & 1 & 0 \end{bmatrix}$$

then at least one of the following holds:

(2.5a) for some $j$, $a_{i_1,j} = a_{i_2,j} = 0$, and for all $k$, $a_{kj} \leq a_{kj_1} + a_{kj_2}$;

(2.5b) for some $j < j_1$, and for all $k$, $a_{kj} \leq a_{kj_1} + a_{kj_2} + a_{kj_3}$.  

Here is an example. $A$ has 6 rows and 8 columns and satisfies (2.5),

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
4 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
5 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
6 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

The algorithm first chooses $z_1 = 5$, because 5 is the smallest value of $b_k$ among those $k$ such that $a_{k1} = 1$. So $j(1) = 1$, $i(1) = 4$ (because $b_4 = 5$). Columns 3 and 7 are deleted because $a_{43}$ and $a_{47}$ are 1, so $x_3$ and $x_7$ can only be 0. The value of $b_1$ and $b_5$ are reduced by 5. We continue inductively. The marked entries denote \{(i(1), j(1)), (i(2), j(2)), \ldots, (i(5), j(5))\}. The submatrix formed by these rows and columns is

\[
\begin{array}{cccc}
1 & 2 & 4 & 5 \\
4 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
3 & 0 & 0 & 1 \\
5 & 1 & 1 & 1 \\
2 & 0 & 1 & 0 \\
\end{array}
\]

The solution to primal and dual problems are

\[
\begin{align*}
\bar{x} &= (5, 2, 0, 4, 1, 0, 0, 5), & \bar{y} &= (0, 1, 1, 1, 0).
\end{align*}
\]

3. Applications

(3.1) Given a directed graph $G$ with distinguished nodes $s \neq t$.

Let $A$ be the following $(0, 1)$ matrix. The rows correspond to edges of $G$, columns to subsets $S \subseteq v(G)$, $s \in S$, $t \notin S$, with

\[
a_{eS} = \begin{cases} 
1 & \text{if edge } e \text{ ‘leaves’ } s, \\
0 & \text{otherwise.}
\end{cases}
\]

Then, if $S$ are numbered by increasing cardinality of $|S|$, $A$ satisfies (2.5). SG is max cut packing (cf. [6, p. 592]), and DSG is Dijkstra's algorithm.

(3.2) Given a directed graph $G$ with distinguished node $r$. Let rows of $A$ corresponding to edges of $G$, columns to subsets $S \subseteq V(G)$, $r \notin S$. Set

\[
a_{eS} = \begin{cases} 
1 & \text{if edge } e \text{ ‘enters’ } s, \\
0 & \text{otherwise.}
\end{cases}
\]
If the columns of \( A \) are ordered by increasing size of \( |S| \), then \( A \) satisfies (2.5), and DSG is the algorithm described in [5].

(3.3) Let \( A = [I^p] \). Then \( A \) satisfies (2.5) if and only if \( B \) contains neither of the \( 2 \times 3 \) matrices mentioned there. Hence, our theorem includes the problems mentioned in [3].

(3.4) It is well known that sequential greediness for any sequence of \( s - t \) paths solves the max flow problem for series-parallel graphs. Consider the incidence matrix \( A \) of edges versus paths of such a graph. It is easy to see that (2.5a) applies.

(3.5) Following Ford and Fulkerson in [1], one can find an optimum coloring of an interval graph \( G \) by the following procedure. Say interval \( I_i \) precedes interval \( I_j \) if the right-hand endpoint of \( I_i \) is to the left of the left-hand endpoint of \( I_j \). We can color \( G \) optimally by finding the smallest number of chains covering this partially ordered set, which is equivalent to finding a max match in the bipartite graph on \( I_1, \ldots, I_m \) and \( I'_1, \ldots, I'_m \) where \( I_i \) and \( I'_j \) are joined by an edge if \( I_i \) precedes \( I'_j \).

Observe that if

\[
M = (m_{ij}) = \begin{cases} 1 & \text{if } I_i \text{ precedes } I_j, \\ 0 & \text{otherwise,} \end{cases}
\]

and the numbering of rows and columns is consonant with the partial ordering, then \( M \) does not contain

\[
\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}
\]

(3.4a)
as a submatrix.

But the non-existence of (3.4a) as a submatrix of \( M \) implies that the linear program

\[
\begin{align*}
\max & \sum x_{ij}, \\
x_{ij} & \text{defined only if } m_{ij} = 1, \ x_{ij} \geq 0, \\
\sum_j x_{ij} & \leq 1 \quad \text{for all } i, \\
\sum_i x_{ij} & \leq 1 \quad \text{for all } j,
\end{align*}
\]
is solved by ‘Northwest Corner’ greediness, because of (2.5a).

Acknowledgement

We are very grateful to P. Krishnarao, Stanford University, for his help in clarifying this material.
On simple combinatorial optimization problems

References

1988).