weighted matching problem is:

$$\text{maximize} \sum_{i,j} w_{ij}x_{ij}$$

subject to

$$\sum_j x_{ij} \leq 1,$$

$$\sum_i x_{ij} \leq 1,$$

$$x_{ij} \geq 0,$$

with the understanding that

$$x_{ij} = 1 \Rightarrow (i,j) \in X,$$

$$x_{ij} = 0 \Rightarrow (i,j) \notin X.$$

The dual linear programming problem is:

$$\text{minimize} \sum_i u_i + \sum_j v_j$$

subject to

$$u_i + v_j \geq w_{ij},$$

$$u_i \geq 0,$$

$$v_j \geq 0.$$

Orthogonality conditions which are necessary and sufficient for optimality of primal and dual solutions are:

$$x_{ij} > 0 \Rightarrow u_i + v_j = w_{ij}, \tag{8.1}$$

$$u_i > 0 \Rightarrow \sum_j x_{ij} = 1, \tag{8.2}$$

$$v_j > 0 \Rightarrow \sum_i x_{ij} = 1. \tag{8.3}$$

The Hungarian method maintains primal and dual feasibility at all times, and in addition maintains satisfaction of all orthogonality conditions, except conditions (8.2). The number of such unsatisfied conditions is decreased monotonically during the course of the computation.

The procedure is begun with the feasible matching $X = \varnothing$ and with the feasible dual solution $u_i = W$, where $W \geq \max \{w_{ij}\}$, and $v_j = 0$, for all i, j. These initial primal and dual solutions clearly satisfy all of the conditions (8.1) and (8.3), but not the conditions (8.2).

At the general step of the procedure, X is feasible, u_i and v_j are dual feasible, all conditions (8.1) and (8.3) are satisfied, but some of the conditions (8.2) are not. One then tries, by means of a labeling procedure, to find an

augmenting p
$u_i + v_j = w_{ij}$.
node i in S fo
new matching
satisfied, and
If augmentati
ables, by subt
to v_j to each

It is al
added to an a
choice of δ is
smallest value
at each step si
ables have bee
made. It follov
(8.2) are satisf

The au
$u_i + v_j = w_{ij}$;
alternating tre
one of two thi
matching is of
adding to the
that $u_i + v_j >$

Let us
to add to the a
dions (8.1) are
are incident t
tree. In the ma
situation, ther
tree. In the pr
as to achieve

Suppos
δ to v_j for each
the net value
other end out
increased by δ,
the current ma
decreased by

The eff
Figure 5.16. U
v_j. On each arc
ties are accour
to have one e

If the a

augmenting path within the subgraph containing only arcs (i,j) for which $u_i + v_j = w_{ij}$. In particular, an augmenting path is sought from an exposed node i in S for which (necessarily) $u_i > 0$. If such a path can be found, the new matching will be feasible, all conditions (8.1) and (8.3) continue to be satisfied, and one more of the conditions (8.2) will be satisfied than before. If augmentation is not possible, then a change of δ is made in the dual variables, by subtracting $\delta > 0$ from u_i for each labeled S-node i and adding δ to v_j to each labeled T-node j.

It is always possible to choose δ so that at least one new arc can be added to an alternating tree, while maintaining dual feasibility, unless the choice of δ is restricted by the size of u_i at some S-node. But u_i takes on its smallest value at the exposed S-nodes. The exposed nodes have been exposed at each step since the beginning of the algorithm, and hence their dual variables have been decremented each time a change in dual variables has been made. It follows that when u_i is reduced to zero at these nodes, the conditions (8.2) are satisfied, and both the primal and dual solutions are optimal.

The augmentation computation is such that only arcs (i,j) for which $u_i + v_j = w_{ij}$ are placed in the alternating trees. If the construction of the alternating trees concludes without an augmenting path being found, then one of two things has occurred. Either the trees are truly Hungarian and the matching is of the maximum cardinality, or else it is not possible to continue adding to the trees because all arcs (i,j) available for that purpose are such that $u_i + v_j > w_{ij}$.

Let us deal with the latter case first. Any arcs which we should like to add to the alternating trees are arcs not in the matching X. (Because condions (8.1) are satisfied, arcs in X are such that $u_i + v_j = w_{ij}$.) Such arcs are incident to an S-node in an alternating tree and a T-node not in any tree. In the max-min problem, we lowered the threshold in the comparable situation, thereby permitting at least one arc to be added to an alternating tree. In the present case, we manipulate the values of the dual variables so as to achieve the desired effect.

Suppose we subtract $\delta > 0$ from u_i for each S-node i in a tree and add δ to v_j for each T-node j in a tree. Such a change in the dual variables affects the net value of $u_i + v_j$ only for arcs which have one end in a tree and the other end out. If such an arc is incident to a T-node of the tree, $u_i + v_j$ is increased by δ, which is of no consequence (note that such an arc cannot be in the current matching). If the arc is incident to an S-node of a tree, $u_i + v_j$ is decreased by δ, possibly to w_{ij}, in which case it can be added to the tree.

The effect of the changes in the dual variables is summarized in Figure 5.16. Under each node in that figure is indicated the change in u_i or v_j. On each arc is indicated the net change in $u_i + v_j$ for that arc. All possibilities are accounted for. (Note that it is not possible for an arc in the matching to have one end in an alternating tree and the other end out.)

If the alternating trees are truly Hungarian, then the choice of δ is

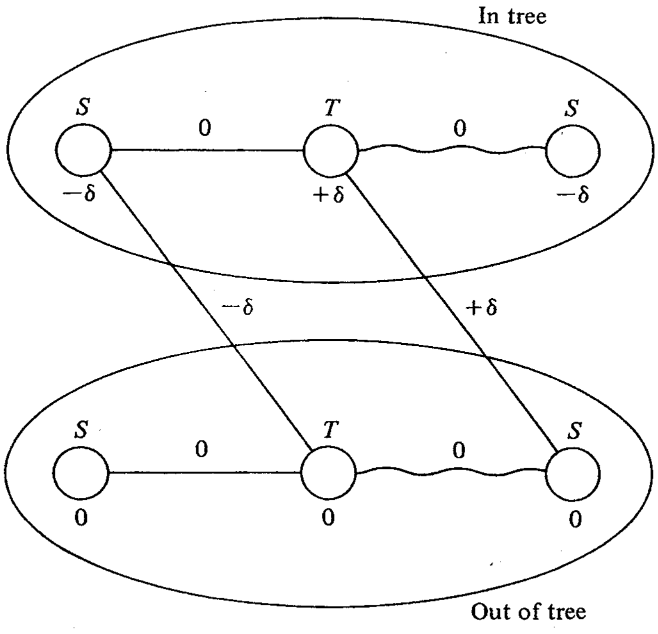

Figure 5.16 Effect of change in dual variables

indeed determined by the value of u_i at the exposed S-nodes. In this case, the values of the dual variables are changed, as indicated above, conditions (8.2) are satisfied, and both the primal and dual solutions are optimal.

The algorithm begins with the empty matching, X_0, and then produces matchings $X_1, X_2, \ldots, X_k$, containing $1, 2, \ldots, k$ arcs. Each of these matchings is of maximum weight, with respect to all other matchings of the same cardinality, as is shown below. (Incidentally, note that the maximum weight matching existing at the end of the computation does not necessarily have maximum cardinality.)

Suppose we were to demand a maximum weight matching, subject to the constraint that it contains no more than k arcs. Then we could add a single constraint to the primal linear programming problem:

$$\sum_{ij} x_{ij} \leq k.$$

This constraint is identified with a dual variable λ and, after appropriate modifications in the dual problem, the orthogonality conditions become

$$x_{ij} > 0 \Rightarrow u_i + v_j + \lambda = w_{ij},$$

$$u_i > 0 \Rightarrow \sum_j x_{ij} = 1,$$

$$v_j > 0 \Rightarrow \sum_i x_{ij} = 1,$$

$$\lambda > 0 \Rightarrow \sum_{i,j} x_{ij} = k.$$

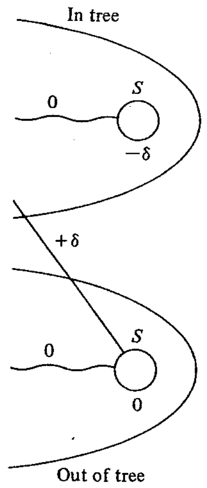

Figure 5.16 Effect of change in dual variables

lue of u_i at the exposed S-nodes. In this case, the
are changed, as indicated above, conditions
he primal and dual solutions are optimal.
with the empty matching, X_0, and then pro-
, X_k, containing $1, 2, \ldots, k$ arcs. Each of these
weight, with respect to all other matchings of
own below. (Incidentally, note that the maxi-
ting at the end of the computation does not
cardinality.)

demand a maximum weight matching, subject
ains no more than k arcs. Then we could add a
mal linear programming problem:

$$\sum_{ij} x_{ij} \leq k.$$

with a dual variable λ and, after appropriate
roblem, the orthogonality conditions become

$$0 \Rightarrow u_i + v_j + \lambda = w_{ij},$$

$$0 \Rightarrow \sum_j x_{ij} = 1,$$

$$0 \Rightarrow \sum_i x_{ij} = 1,$$

$$0 \Rightarrow \sum_{i,j} x_{ij} = k.$$

Let X_k be the matching of cardinality k obtained by the algorithm, and $\bar{u}_i$, $\bar{v}_j$ be the dual solution. Choose $\lambda = \min\{\bar{u}_i\}$. Then $X_k, \bar{u}_i - \lambda, \bar{v}_j, \lambda$ are feasible primal and dual solutions for the k-cardinality problem and satisfy the new orthogonality conditions indicated above. It follows that X_k is of maximum weight, with respect to all matchings containing k arcs.

As in the case of the threshold algorithm for max-min matching, a number π_j is associated with each node j in T. This number indicates the value of δ by which the dual variables must be changed, in order that j may be added to an alternating tree. The labeling procedure progressively decreases π_j until π_j is equal to the smallest value of $u_i + v_j - w_{ij}$, for arcs (i, j) with $i \in S$ labeled. A node j in T may receive a label if $\pi_j > 0$, but its label is scanned only if $\pi_j = 0$. In other words, j is "in tree" if and only if $\pi_j = 0$.

The algorithm is summarized below. We leave it as an exercise for the reader to verify that the number of computational steps required is $O(m^2n)$, the same as for cardinality matching and max-min matching.

BIPARTITE WEIGHTED MATCHING ALGORITHM

Step 0 (*Start*) The bipartite graph $G = (S, T, A)$ and a weight w_{ij} for each arc $(i,j) \in A$ are given. Set $X = \varnothing$. Set $u_i = \max\{w_{ij}\}$ for each node $i \in S$. Set $v_j = 0$ and $\pi_j = +\infty$ for each node $j \in T$. No nodes are labeled.

Step 1 (*Labeling*)

(1.0) Give the label "$\varnothing$" to each exposed node in S.

(1.1) If there are no unscanned labels, or if there are unscanned labels, but each unscanned label is on a node i in T for which $\pi_i > 0$, then go to Step 3.

(1.2) Find a node i with an unscanned label, where either $i \in S$ or else $i \in T$ and $\pi_i = 0$. If $i \in S$, go to Step 1.3; if $i \in T$, go to Step 1.4.

(1.3) Scan the label on node i ($i \in S$) as follows. For each arc $(i, j) \notin X$ incident to node i, if $u_i + v_j - w_{ij} < \pi_j$, then give node j the label "i" (replacing any existing label) and set $\pi_j = u_i + v_j - w_{ij}$. Return to Step 1.1.

(1.4) Scan the label on node i ($i \in T$) as follows. If node i is exposed, go to Step 2. Otherwise, identify the unique arc $(i, j) \in X$ incident to node i and give node j the label "i." Return to Step 1.1.

Step 2 (*Augmentation*) An augmenting path has been found, terminating at node i (identified in Step 1.4). The nodes preceding node i in the path are identified by "backtracing" from label to label. Augment X by adding to X

all arcs in the augmenting path that are not in X and removing from X those which are. Set $\pi_j = +\infty$, for each node j in T. Remove all labels from nodes. Return to Step 1.0.

Step 3 (*Change in Dual Variables*) Find

$$\begin{aligned} \delta_1 &= \min \{u_i | i \in S\}, \\ \delta_2 &= \min \{\pi_j | \pi_j > 0, j \in T\}, \\ \delta &= \min \{\delta_1, \delta_2\}. \end{aligned}$$

Subtract δ from u_i, for each labeled node $i \in S$. Add δ to v_j for each node $j \in T$ with $\pi_j = 0$. Subtract δ from π_j for each labeled node $j \in T$ with $\pi_j > 0$. If $\delta < \delta_1$ go to Step 1.1. Otherwise, X is a maximum weight matching and the u_i and v_j variables are an optimal dual solution. Halt. //

There is an alternative, "primal" approach to weighted matching. This is to perform successive augmentations of the matching X by means of a maximum weight augmenting path (where the weight of arc (i, j) is taken to be w_{ij} if $(i, j) \in X$ and $-w_{ij}$ if $(i, j) \notin X$). This approach is essentially the same as that used in the previous chapter to compute min-cost flows by successive min-cost augmentations. We refer to this as a "primal" method because it involves no dual variables or other considerations of duality.

It is easy to devise a procedure for determining maximum weight augmentations. In fact, a method essentially like that of Bellman and Ford can be implemented very nicely within the framework of a labeling procedure. The computation of a maximum weight augmenting path requires $O(m^2n)$ steps, when carried out in this way. Since $O(m)$ augmentations are called for, the overall complexity is $O(m^3n)$, compared with $O(m^2n)$ for the Hungarian method.

The efficiency of the primal method can be improved, by making use of node numbers, as described in the previous chapter. The number π_i^k indicates the weight of a maximum weight alternating path from an exposed S-node to node i, relative to matching X_k. These node numbers are used to modify the arc weights, so that all arc weights are negative when a maximum weight augmentation is sought, relative to matching X_{k+1}. (Negative arc weights are desired, since a maximum weight path is sought.) It follows that a Dijkstra-like procedure can be used to find an optimal augmenting path.

When the details have been worked out, it is discovered that the Dijkstra-like procedure looks very much like the Hungarian method. Specifically, the computation of δ_2 in Step 3 of the Hungarian method corresponds to the operation of finding, in the Dijkstra computation, that "tentative"

th that are not in X and removing from X
), for each node j in T. Remove all labels from

ables) Find

in $\{u_i | i \in S\}$,
in $\{\pi_j | \pi_j > 0, j \in T\}$,
in $\{\delta_1, \delta_2\}$.

eled node $i \in S$. Add δ to v_j for each node $j \in T$
π_j for each labeled node $j \in T$ with $\pi_j > 0$.
rwise, X is a maximum weight matching and
optimal dual solution. Halt. //

, "primal" approach to weighted matching.
ugmentations of the matching X by means of
g path (where the weight of arc (i, j) is taken
if $(i, j) \notin X$). This approach is essentially the
us chapter to compute min-cost flows by suc-
s. We refer to this as a "primal" method be-
riables or other considerations of duality.
rocedure for determining maximum weight
od essentially like that of Bellman and Ford
within the framework of a labeling procedure.
um weight augmenting path requires $O(m^2 n)$
s way. Since $O(m)$ augmentations are called
$O(m^3 n)$, compared with $O(m^2 n)$ for the Hun-

mal method can be improved, by making use
in the previous chapter. The number π_i^k indi-
m weight alternating path from an exposed
atching X_k. These node numbers are used to
all arc weights are negative when a maximum
t, relative to matching X_{k+1}. (Negative arc
ximum weight path is sought.) It follows that
be used to find an optimal augmenting path.
been worked out, it is discovered that the
very much like the Hungarian method. Speci-
Step 3 of the Hungarian method corresponds
the Dijkstra computation, that "tentative"

label which is next to be made permanent. Thus, the Hungarian method and the modified primal method are essentially similar.

We noted a similar situation in the previous section, with respect to the threshold method and the max-min augmenting path method for max-min matching. The reader is referred to that discussion.

PROBLEMS

8.1 Apply the Hungarian algorithm to the weighted bipartite graph shown in Figure 5.14 to find a maximum weight matching and an optimal dual solution.

8.2 Interpret each step of the Hungarian algorithm, as nearly as possible, as a step of the out-of-kilter method. Where do the two algorithms differ?

8.3 Generalize the algorithm to the case

$$\sum_j x_{ij} \le a_i,$$
$$\sum_i x_{ij} \le b_j.$$

8.4 (D. Gale) There are m potential house buyers and n potential house sellers, where $m \le n$. Buyer i evaluates house j and decides that its value to him is w_{ij} dollars. If seller j puts a price of v_j on his house, buyer i will be willing to buy only if $w_{ij} \ge v_j$. Moreover, if there is more than one house j for which $w_{ij} \ge v_j$, he will prefer to buy a house for which $w_{ij} - v_j$ is maximal. A set of prices is said to be "feasible" if it is such that for every buyer i there is at least one house j for which $w_{ij} \ge v_j$. Show that, with respect to all other feasible sets of prices, there is one set of prices which maximizes both the sum of the total profits to the buyers,

$$\sum (w_{ij} - v_j)$$

and total proceeds to the sellers, $\sum v_j$.

8.5 Devise a simple example of a matching problem in which a maximum weight matching does not have maximum cardinality. (All arc weights are to be strictly positive.) How should the Hungarian method be modified so as to produce a maximum cardinality matching which is of maximum weight (relative to all other such matchings)?

8.6 Write out, in detail, the steps of a weighted matching algorithm based on the approach of finding maximum-weight augmenting paths by a Dijkstra-like procedure. Make a detailed comparison with the Hungarian algorithm.

9

A Special Case: Gilmore-Gomory Matching

Consider two examples of weighted matching problems which have particularly simple solutions.