November 1992 (revised February 1993) LIDS-P-2147

Mathematical Equivalence of the Auction Algorithm for Assignment and the ϵ-Relaxation (Preflow-Push) Method for Min Cost Flow [1]

by

Dimitri P. Bertsekas[2]

Abstract

It is well known that the linear minimum cost flow network problem can be converted to an equivalent assignment problem. We show here that when the auction algorithm is applied to this equivalent problem with some special rules for choosing the initial object prices and the person submitting a bid at each iteration, one obtains the generic form of the ϵ-relaxation method. The reverse equivalence is already known, that is, if we view the assignment problem as a special case of a minimum cost flow problem and we apply the ϵ-relaxation method with some special rules for choosing the node to iterate on, we obtain the auction algorithm. Thus, the two methods are mathematically equivalent.

[1] Research supported by NSF under Grant No. CCR-9103804, and by the ARO under Grant DAAL03-92-G-0115.

[2] Laboratory for Information and Decision Systems, M.I.T, Cambridge, Mass. 02139.

1. INTRODUCTION

An extension of the assignment auction algorithm [Ber79] was given for the minimum cost flow problem by the author in [Ber86a] and [Ber86b]. This method, called ϵ-relaxation, also contains as a special case a one-phase version of the preflow-push algorithm for the max-flow problem, earlier developed by Goldberg and Tarjan [Gol85], [GoT86]. These methods, which are frequently called *auction* or *preflow-push* methods, have attracted much interest recently because of their excellent practical performance and worst-case complexity properties [AOT89], [AhO89], [BeE87], [BeE88], [Ber88], [ChM89], [Gol87], [GoT90], [MPS91]. An extensive account can be found in the textbooks [AMO89], [BeT89], [Ber91], and in the tutorial survey [Ber92].

The purpose of this paper is to show that the ϵ-relaxation method is not just a generalization of the original 1979 auction algorithm but is in fact mathematically equivalent with it, in the sense that each method can be derived starting from the other. We have shown elsewhere [BeE87], [BeE88], [BeT89] (p. 374) how to derive the auction algorithm starting from the ϵ-relaxation method, so in this paper we will focus in the reverse derivation. In particular, we apply the auction algorithm to an assignment problem, which is equivalent to the linear minimum cost flow problem. After we streamline the computations, we obtain the generic form of the ϵ-relaxation method for the original problem. As a corollary we obtain that the generic form of the preflow-push max-flow algorithm can be viewed as a special case of the auction algorithm.

2. THE AUCTION ALGORITHM FOR THE ASSIGNMENT PROBLEM

In the classical symmetric assignment problem there are n persons and n objects that we have to match on a one-to-one basis. There is a benefit a_{ij} for matching person i with object j and we want to assign persons to objects so as to maximize the total benefit. We are given a set $\mathcal{A}$ of pairs (i,j) that can be matched. For each person i, we denote by $A(i)$ the set of objects that can be matched with i

$$A(i) = \{j \mid (i,j) \in \mathcal{A}\}. \tag{1}$$

For simplicity we assume that there are at least two objects in each set $A(i)$.

By an *assignment* we mean a set S of person-object pairs (i,j) such that each person i and each object j is involved in at most one pair from S. If the number of pairs in S is n, so that every person is assigned to a distinct object, we say that S is *feasible*; otherwise S is said to be *infeasible*. If a feasible assignment exists, the problem is said to be feasible, and otherwise it is said to be infeasible. We seek an optimal assignment within the set of feasible assignments, that is, a set of person-object pairs $(1, j_1), \ldots, (n, j_n)$ from $\mathcal{A}$, such that the objects $j_1, \ldots, j_n$ are all distinct and the total benefit $\sum_{i=1}^{n} a_{ij_i}$ is maximum.

The auction algorithm uses a positive scalar $\epsilon > 0$, and maintains a price vector p consisting of a price p_j for each object j and an assignment S. We refer to $a_{ij} - p_j$ as the *value of object j for person i*. Throughout the algorithm, the pair (p, S) satisfies the condition

$$a_{ij_i} - p_{j_i} \geq \max_{j \in A(i)} \{a_{ij} - p_j\} - \epsilon, \qquad \forall\, (i, j_i) \in S, \tag{2}$$

known as *ϵ-complementary slackness* or (ϵ-CS for short). Initially one may use any price vector p and the empty assignment $S = \emptyset$; this pair trivially satisfies ϵ-CS. The algorithm terminates if S is feasible, that is, if all persons are assigned. Otherwise an unassigned person i is selected to bid in the auction as follows.

Typical Iteration of the Auction Algorithm

An unassigned person i finds an object j_i that maximizes over all $j \in A(i)$

$$a_{ij} - p_j, \tag{3}$$

and increases p_{j_i} to the level

$$a_{ij_i} - w_i + \epsilon, \tag{4}$$

where w_i is the second best object value

$$w_i = \max_{j \in A(i), j \neq j_i} \{a_{ij} - p_j\}. \tag{5}$$

The pair (i, j_i) is added to the assignment S and, if j_i was assigned to some person k at the beginning of the iteration, the pair (k, j_i) is deleted from S.

The main property of the method is that for a feasible problem, it terminates with a feasible assignment S that is optimal within $n\epsilon$; S is strictly optimal if $\epsilon < 1/n$ and the benefits a_{ij} are integer. As suggested in the original proposal of the method [Ber79], it is often beneficial to use *ϵ-scaling*, that is, apply the algorithm several times with decreasing values of ϵ, each time obtaining a favorable initial price vector for the next application.

There are several variants of the auction algorithm that differ from the preceding algorithm in small details. For example, several persons may bid simultaneously with objects being awarded to the highest bidders, the price increment may be slightly different than the one of Eq. (5), etc. The important ingredients for each iteration of the method are that:

(a) ϵ-CS is maintained.

(b) At least one unassigned person gets assigned to some object and the price of this object is increased by at least ϵ. Furthermore, the previously assigned person to each object that gets assigned at the iteration (if any) becomes unassigned.

(c) No price is decreased and every object that was assigned at the start of the iteration remains assigned at the end of the iteration.

Any variant of the auction algorithm that obeys these three rules can be shown to have the termination property of the basic method stated above.

3. A PROFIT-ORIENTED AUCTION ALGORITHM FOR THE ASSIGNMENT PROBLEM

There is considerable freedom in the way one uses the auction algorithm. In particular, one may choose arbitrarily the unassigned person to bid at each iteration. Furthermore, the initial price-assignment pair is arbitrary other than it must satisfy ϵ-CS. By specifying various restrictions on the choice of the bidding person and the initial price-assignment pair, one can obtain special cases of the algorithm. In this section we will describe one such special case, which has not been reported thus far and will prove useful for our purposes in this paper.

In this section we will introduce a version of the auction algorithm, which instead of a price vector, maintains a *profit* vector π, consisting of a profit π_i for each person i. The profit vector, however, implicitly determines the price vector. Profits are in effect dual variables, and play the same role for persons as prices play for objects. If in the course of the auction object j is assigned to person i, we can view the value $a_{ij} - p_j$ as the current level of profit for i. With this in mind, it is reasonable that a profit π_i for each assigned person i should specify the prices of the corresponding assigned objects j_i via the relation $p_{j_i} = a_{ij_i} - \pi_i$. However, to be able to express *all* the object prices in terms of the person profits, we need the notion of a *preassignment* $\overline{S}$, which is defined as a set of n pairs from $\mathcal{A}$ consisting of exactly one pair (i_j, j) for each object j. The person i_j is called the *preferred person of object* j, and its profit defines a price p_j via the relation

$$p_j = a_{i_j j} - \pi_{i_j}, \qquad i = 1, \ldots, n. \tag{6}$$

The price vector p thus defined is called the *price vector implied by* $(\pi, \overline{S})$. Note that there are two possibilities regarding a preassignment $\overline{S}$:

(a) Each person is the preferred person for exactly one object, in which case $\overline{S}$ is a feasible assignment.

(b) There is at least one person that is not preferred by any object, and at least one person that is preferred by more than one object. In this case, $\overline{S}$ contains as subsets a collection $G(\overline{S})$ of assignments all of which are infeasible. We call $G(\overline{S})$ the *assignment subset* of the preassignment $\overline{S}$.

Definition 1: We say that a profit-preassignment pair $(\pi, \overline{S})$ satisfies ϵ-CS if the price vector implied by $(\pi, \overline{S})$ together with each assignment in the assignment subset $G(\overline{S})$ satisfies ϵ-CS.

Using the definition (2) of ϵ-CS, it is seen that a profit-preassignment pair $(\pi, \overline{S})$ satisfies ϵ-CS if and only if the condition

$$\pi_i + p_j \geq a_{ij} - \epsilon, \qquad \forall\ i \text{ preferred by at least one object, and } j \in A(i) \tag{7}$$

where p_j is the implied price of j. Thus in order for the pair $(\pi, \overline{S})$ to satisfy ϵ-CS, it is sufficient that

$$\pi_i + p_j \geq a_{ij} - \epsilon, \qquad \forall\ (i,j) \in \mathcal{A}. \tag{8}$$

Note also that given any profit vector π and $\epsilon > 0$, we can obtain a preassignment $\overline{S}$ satisfying ϵ-CS together with π by defining

$$p_j = \max_{\{i|(i,j)\in\mathcal{A}\}} \{a_{ij} - \pi_i\}, \qquad \forall\ j = 1, \ldots, n, \tag{9}$$

and by letting $\overline{S}$ consist of exactly one pair (i_j, j) attaining the maximum in the above equation for each j. The sufficient condition (8) for ϵ-CS will then be satisfied by $(\pi, \overline{S})$.

The following auction algorithm starts with and maintains a profit-preassignment pair $(\pi, \overline{S})$ that satisfies ϵ-CS. If the preassignment $\overline{S}$ is an assignment (necessarily feasible), the algorithm terminates, and $\overline{S}$ is within $n\epsilon$ of being optimal. Otherwise, a person i that is not preferred by any object under $\overline{S}$ is selected to bid under the rules of the auction algorithm based on the price vector p implied by $(\pi, \overline{S})$. The auction iteration, however, can be expressed in terms of the profit vector π using the definition (6) of the implied price vector. The preassignment can then be rearranged so that the new profit-preassignment pair satisfies ϵ-CS. The complete iteration is as follows (compare with the iteration of the preceding section):

Typical Profit-Oriented Auction Iteration

A person i that is not preferred by any object under $\overline{S}$ finds an object j_i that maximizes over all $j \in A(i)$

$$a_{ij} - a_{k_j j} + \pi_{k_j}, \tag{10}$$

where k_j is the person preferred by j under $\overline{S}$, and sets π_i to the level

$$w_i - \epsilon, \tag{11}$$

where w_i is the second best object value in terms of the implied prices p_j,

$$w_i = \max_{j\in A(i), j\neq j_i} \{a_{ij} - p_j\} = \max_{j\in A(i), j\neq j_i} \{a_{ij} - a_{k_j j} + \pi_{k_j}\}. \tag{12}$$

The pair (i, j_i) is added to the preassignment $\overline{S}$ and the pair (i_{k_i}, j) that belonged to $\overline{S}$ at the beginning of the iteration is deleted from $\overline{S}$.

It can be seen that the final assignment and the sequence of implied price vectors generated by the above auction algorithm can also be generated by the auction algorithm of the preceding section with a special choice of initial price vector and sequence of persons submiting a bid at each iteration. Thus the algorithm of this section is a special case of the algorithm of the preceding section.

4. AUCTION ALGORITHMS FOR PROBLEMS WITH SIMILAR PERSONS

We now consider a special type of assignment problem that involves groups of persons that are indistiguinshable in the sense that they can be assigned to the same objects and with the same corresponding benefits.

Definition 2: We say that two persons i and i' are *similar*, if

$$A(i) = A(i'), \qquad \text{and} \qquad a_{ij} = a_{i'j} \quad \forall\, j \in A(i). \tag{13}$$

For each person i, the set of all persons similar to i is called the *similarity class of* i.

If there are similar persons, the auction algorithm can get bogged down into a long sequence of bids (known as a "price war"), whereby a number of similar persons compete for a smaller number of objects by making small incremental price changes. An example is given in Fig. 1. As described in [BeC89] (see also [Ber91] and [Ber92]), if one is aware of the presence of similar persons, one can "compress" a price war within a similarity class into a single iteration. It is important to note that the corresponding algorithm is still a special case of the auction algorithm of Section 2; the computations are merely streamlined by combining many bids into a "collective" bid by the persons of a similarity class.

The method to resolve a price war within a similarity class is to let the auction algorithm run its course, then look at the final results and see how they can be essentially reproduced with less calculation. In particular, suppose that we have a price-assignment pair (p, S) satisfying ϵ-CS, and that a similarity class M has m persons, only $q < m$ of which are assigned under S. Suppose that we restrict the auction algorithm to run within M, that is, we require the bidding person to be from M, until all persons in M are assigned. We call this the *M-restricted auction*.

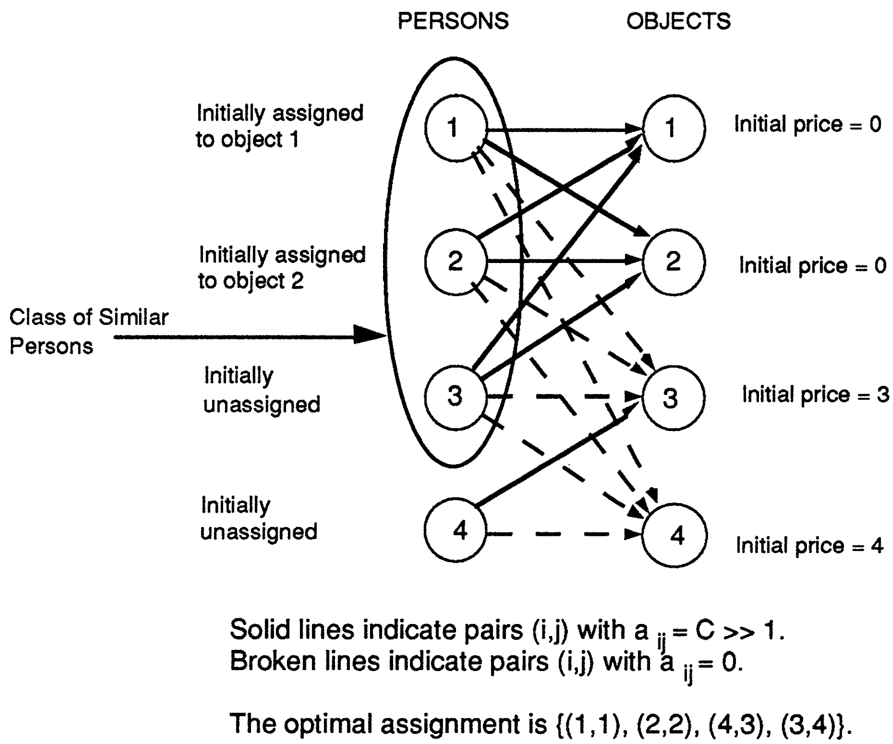

Figure 1: An example of an assignment problem with similar persons. Here the persons 1, 2, and 3 form a similarity class. This structure induces a price war in the auction algorithm. The persons 1, 2, and 3 will keep on bidding up the prices of objects 1 and 2 until the prices p_1 and p_2 reach or exceed $C + 3$. The price increments will be at most 2ϵ.

The final results of an M-restricted auction are quite predictable. In particular, the set

$$A_{new} = \text{The } m \text{ objects that are assigned to persons in } M \text{ at the end of the } M\text{-restricted auction}$$

consists of the set

$$A_{old} = \text{The } q \text{ objects that were assigned to persons in } M \text{ at the beginning of the } M\text{-restricted auction}$$

plus $m - q$ objects not in A_{old} that offered the best value $a_{ij} - p_j$ for the persons $i \in M$ under the price vector p at the start of the M-restricted auction. For a more precise description, let us label the set of objects not in A_{old} in order of decreasing value, that is,

$$\{j \mid j \notin A_{old}\} = \{j_1, \ldots, j_{m-q}, j_{m-q+1}, \ldots, j_{n-q}\}, \tag{14}$$

where for all persons $i \in M$,

$$a_{ij_r} - p_{j_r} \geq a_{ij_{r+1}} - p_{j_{r+1}}, \qquad r = 1, \ldots, n - q - 1. \tag{15}$$

Then

$$A_{new} = A_{old} \cup \{j_1, \ldots, j_{m-q}\}. \tag{16}$$

The price changes of the objects as a result of the M-restricted auction can also be predicted to a great extent. In particular, the prices of the objects not in A_{new} will not change, while the ultimate prices of the objects $j \in A_{new}$ will be such that the corresponding values $a_{ij} - p_j$ for the persons $i \in M$ will all be within ϵ of each other and no less than the value $a_{ij_{m-q+1}} - p_{j_{m-q+1}}$ of the next best object j_{m-q+1} minus ϵ. At this point, to simplify the calculations, we can just raise the prices of the objects $j \in A_{new}$ so that their final values $a_{ij} - p_j$ for persons $i \in M$ are exactly equal to the value $a_{ij_{m-q+1}} - p_{j_{m-q+1}}$ of the next best object j_{m-q+1} minus ϵ, that is, we set

$$p_j := a_{ij} - \left(a_{ij_{m-q+1}} - p_{j_{m-q+1}}\right) + \epsilon, \qquad \forall\, j \in A_{new}, \tag{17}$$

where i is any person in M. It can be seen that this maintains the ϵ-CS property of the resulting price-assignment pair, and that the desirable termination properties of the algorithm are maintained.

Consider the operation that starts with a price-assignment pair (p, S) satisfying ϵ-CS and a similarity class M that has m persons, only q of which are assigned under S, and produces through an M-restricted auction a price-assignment pair specified by Eqs. (14)-(17). We call this operation an *M-auction iteration*. Note that when the similarity class M consists of a single person, an M-auction iteration produces the same results as the simpler auction iteration given earlier. Thus the algorithm that consists of a sequence of M-auction iterations generalizes the auction algorithm iteration given earlier, and deals effectively with the presence of similarity classes. Table 1 illustrates this algorithm.

At Start of Iteration #	**Object Prices**	**Assigned Pairs**	**Bidder Class M**	**Preferred Object(s)**
1	0,0,3,4	(1,1),(2,2)	$\{1,2,3\}$	1,2,3
2	$C+4+\epsilon, C+4+\epsilon, 4+\epsilon, 4$	(1,1),(2,2),(3,3)	$\{4\}$	3
3	$C+4+\epsilon, C+4+\epsilon, C+4+\epsilon, 4$	(1,1),(2,2),(4,3)	$\{1,2,3\}$	1,2,4
Final	$2C+4+2\epsilon, 2C+4+2\epsilon, C+4+\epsilon, C+4+2\epsilon$	(1,1),(2,2),(4,3),(3,4)		

Table 1: Illustration of the algorithm based on M-auction iterations for the problem of Fig. 1.

It is possible to derive also a profit-oriented version of algorithm that uses M-auction iterations.

This algorithm maintains a profit-preassignment pair $(\pi, \overline{S})$ satisfying ϵ-CS. We impose two more requirements:

(a) The profits of all persons in each similarity class must be equal.

(b) Persons in a similarity class should participate fairly in $\overline{S}$, that is, no person of a similarity class is preferred by more than one object while another person in the same similarity class is not preferred by any object.

A profit-preassignment pair $(\pi, \overline{S})$ satisfying the above two conditions is said to be *fair*.

At each iteration of the algorithm we have a fair profit-preassignment pair $(\pi, \overline{S})$ satisfying ϵ-CS. If $\overline{S}$ is an assignment (necessarily feasible) the algorithm terminates. Otherwise the following iteration is executed.

Typical Profit-Oriented M-Auction Iteration

Let M be a similarity class that has m persons, only $q < m$ of which are preferred by some object under $\overline{S}$. Denote by A_{old} the set of q objects j such that $(i,j) \in \overline{S}$ for some person $i \in M$. Label the objects not in A_{old} in order of decreasing value, that is, as

$$\{j \mid j \notin A_{old}\} = \{j_1, \ldots, j_{m-q}, j_{m-q+1}, \ldots, j_{n-q}\},$$

where for all persons $i \in M$,

$$a_{ij_r} - (a_{k_{j_r} j_r} - \pi_{k_{j_r}}) \geq a_{ij_{r+1}} - (a_{k_{j_{r+1}} j_{r+1}} - \pi_{k_{j_{r+1}}}), \qquad r = 1, \ldots, n-q-1, \tag{18}$$

and for each object j, k_j is the person preferred by j. The iteration sets the profit π_i of each person $i \in M$ to the level

$$w_i - \epsilon, \tag{19}$$

where w_i is the value of the next best object j_{m-q+1} in terms of the implied prices p_j,

$$w_i = a_{ij_{m-q+1}} - p_{j_{m-q+1}} = a_{ij_{m-q+1}} - a_{k_{j_{m-q+1}} j_{m-q+1}} + \pi_{k_{j_{m-q+1}}}, \tag{20}$$

and modifies the preassignment $\overline{S}$ as follows: It deletes from $\overline{S}$ the m pairs associated with the objects in the set

$$A_{new} = A_{old} \cup \{j_1, \ldots, j_{m-q}\}$$

and then adds to $\overline{S}$ m pairs (i, j_i) that assign each person $i \in M$ to a distinct object j_i of A_{new}.

It can be seen that the profit-preassignment pairs $(\pi, \overline{S})$ generated by this algorithm are fair and satisfy ϵ-CS. As an example consider using the algorithm to solve the problem of Fig. 1 with initial profit vector $\pi = (0, 0, 0, -1)$ and preassignment $\overline{S} = \{(1,1), (2,2), (4,3), (4,4)\}$. The initial implied price vector is $p = (C, C, C+1, 1)$. The bidder class at the first iteration will be $M = \{1, 2, 3\}$ and will bid for the set of three best objects $A_{new} = \{1, 2, 4\}$. The profits of the bidder class will be lowered to the implied value $-(C+1)$ of the next best object (object 3) minus ϵ. Thus the new profit vector will be $\pi = \big(-(C+1+\epsilon), -(C+1+\epsilon), -(C+1+\epsilon), -1\big)$, the new preassignment will be $\overline{S} = \big\{(1,1), (2,2), (4,3), (3,4)\big\}$. This preassignment is a (feasible) assignment, so the algorithm will terminate.

For the remainder of the paper, we will assume that the persons have been partitioned in disjoint similarity classes and all references to the auction algorithm refer to the generalized version of the profit-oriented auction algorithm that consists of M-auction iterations as described above. We will now show that this algorithm yields as a special case the ϵ-relaxation method.

5. THE MINIMUM COST FLOW PROBLEM AND ITS EQUIVALENT ASSIGNMENT PROBLEM

We consider the minimum cost flow problem with upper and lower bounds on the arc flows. For simplicity of notation, we will take the lower arc flow bounds to be zero. Nonzero lower arc flow bounds can be set to zero without loss of generality by subtracting the lower bound vector from the arc flow vector. We are given a directed graph with set of nodes $\mathcal{N}$ and set of arcs $\mathcal{A}$. Each arc (i,j) carries a flow x_{ij}. We denote by x the flow vector $\{x_{ij} \mid (i,j) \in \mathcal{A}\}$. We consider the problem

$$\text{minimize} \quad \sum_{(i,j)\in\mathcal{A}} a_{ij} x_{ij} \tag{MCF}$$

subject to

$$\sum_{\{j \mid (i,j)\in\mathcal{A}\}} x_{ij} - \sum_{\{j \mid (j,i)\in\mathcal{A}\}} x_{ji} = s_i, \qquad \forall\, i \in \mathcal{N},$$

$$0 \le x_{ij} \le c_{ij}, \qquad \forall\, (i,j) \in \mathcal{A},$$

where a_{ij}, c_{ij}, and s_i are given integers.

The problem can be converted into an equivalent transportation problem by replacing each arc (i,j) by a node labeled (i,j), and two incoming arcs $\big(i,(i,j)\big)$ and $\big(j,(i,j)\big)$ to that node as shown in Fig. 2. The flows of these arcs are denoted $y_{i(i,j)}$ and $z_{j(i,j)}$, and correspond to the arc flow x_{ij} via the transformation

$$y_{i(i,j)} = x_{ij}, \qquad z_{j(i,j)} = c_{ij} - x_{ij}.$$

The equivalent transportation problem is

$$\begin{aligned}
\text{maximize} \quad & \sum_{i \in \mathcal{N}} \sum_{\{j \mid (i,j) \in \mathcal{A}\}} a_{ij} z_{j(i,j)} \\
& \text{subject to} \\
& \sum_{\{j \mid (i,j) \in \mathcal{A}\}} y_{i(i,j)} + \sum_{\{j \mid (j,i) \in \mathcal{A}\}} z_{i(j,i)} = s_i + \sum_{\{j \mid (j,i) \in \mathcal{A}\}} c_{ji}, \qquad \forall\, i \in \mathcal{N}, \\
& y_{i(i,j)} + z_{j(i,j)} = c_{ij}, \qquad \forall\, (i,j) \in \mathcal{A}, \\
& 0 \le y_{i(i,j)}, \qquad 0 \le z_{j(i,j)}, \qquad \forall\, (i,j) \in \mathcal{A},
\end{aligned}$$

(see Fig. 3).

It is in turn possible to transform this transportation problem into an assignment problem with similar persons by means of the following two devices (see Fig. 4):

(a) Create $s_i + \sum_{\{j \mid (j,i) \in \mathcal{A}\}} c_{ji}$ similar persons in place of each node/source i of the transportation problem.

(b) Create c_{ij} duplicate objects for each arc/sink (i,j) of the transportation problem. The benefit for assigning any one of the duplicate objects corresponding to arc (i,j) is zero for a person in the similarity class corresponding to node i, and is a_{ij} for a person in the similarity class corresponding to node j (cf. Fig. 3).

We will use this equivalence to transcribe the auction algorithm of the previous section into the minimum cost flow context. We first derive the appropriate form of ϵ-CS.

Let us consider the equivalent assignment problem and denote by π the corresponding profit vector. For a given $\epsilon > 0$, consider a fair profit-preassignment pair $(\pi, \overline{S})$ for this problem. Then $\overline{S}$ uniquely defines a flow vector (y, z) for the equivalent transportation problem. In particular, $y_{i(i,j)}$ [or $z_{j(i,j)}$] is the number of pairs of $\overline{S}$ associated with a person of the similarity class corresponding to node i (or node j, respectively), and with a duplicate object corresponding to arc (i,j). Because there is a unique pair in $\overline{S}$ for each object of the equivalent assignment problem, we have

$$y_{i(i,j)} + z_{j(i,j)} = c_{ij}$$

for each arc (i,j), so $\overline{S}$ defines the flow x_{ij} of each arc (i,j) via the relations

$$x_{ij} = y_{i(i,j)} = c_{ij} - z_{j(i,j)}.$$

The profits of all persons corresponding to a node i are equal [since the profit-preassignment pair $(\pi, \overline{S})$ is fair] and will be denoted by π_i. Regarding the implied prices of the objects corresponding to an arc (i,j), we note that the objects paired, according to $\overline{S}$, with persons in the similarity class of i [which correspond to the flow $y_{i(i,j)}$] have implied price $-\pi_i$, while the objects paired with persons

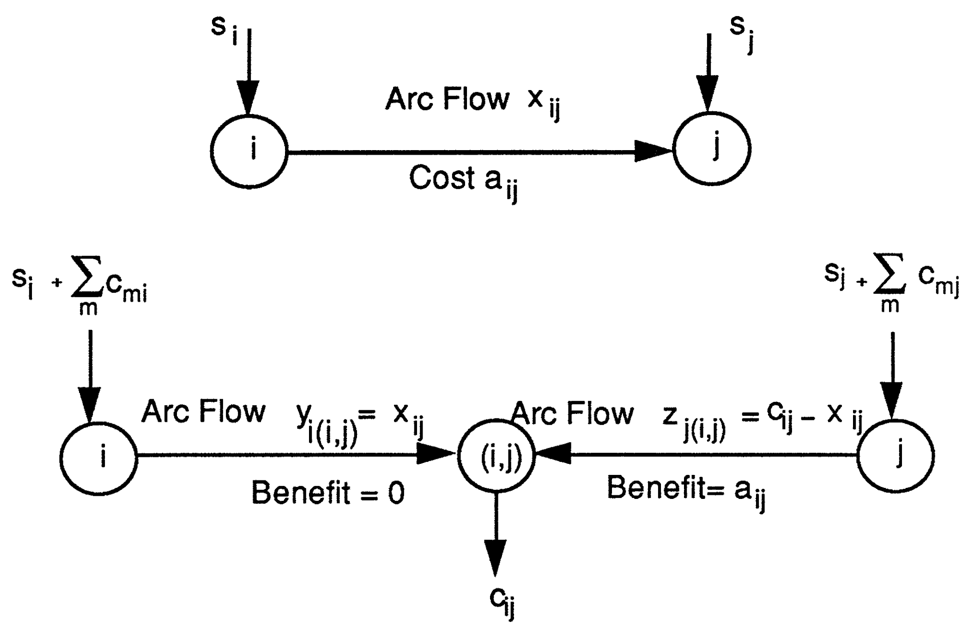

Figure 2: Transformation of a min cost flow problem into a transportation problem. Each arc (i,j) is replaced by a node labeled (i,j) and two incoming arcs $\big(i,(i,j)\big)$ and $\big(j,(i,j)\big)$ to that node.

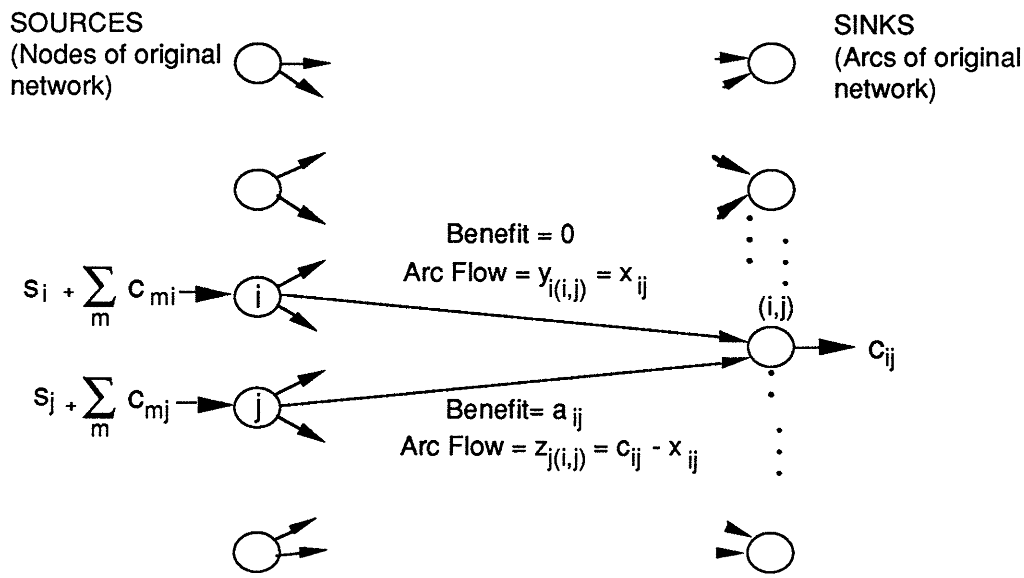

Figure 3: The equivalent transportation problem.

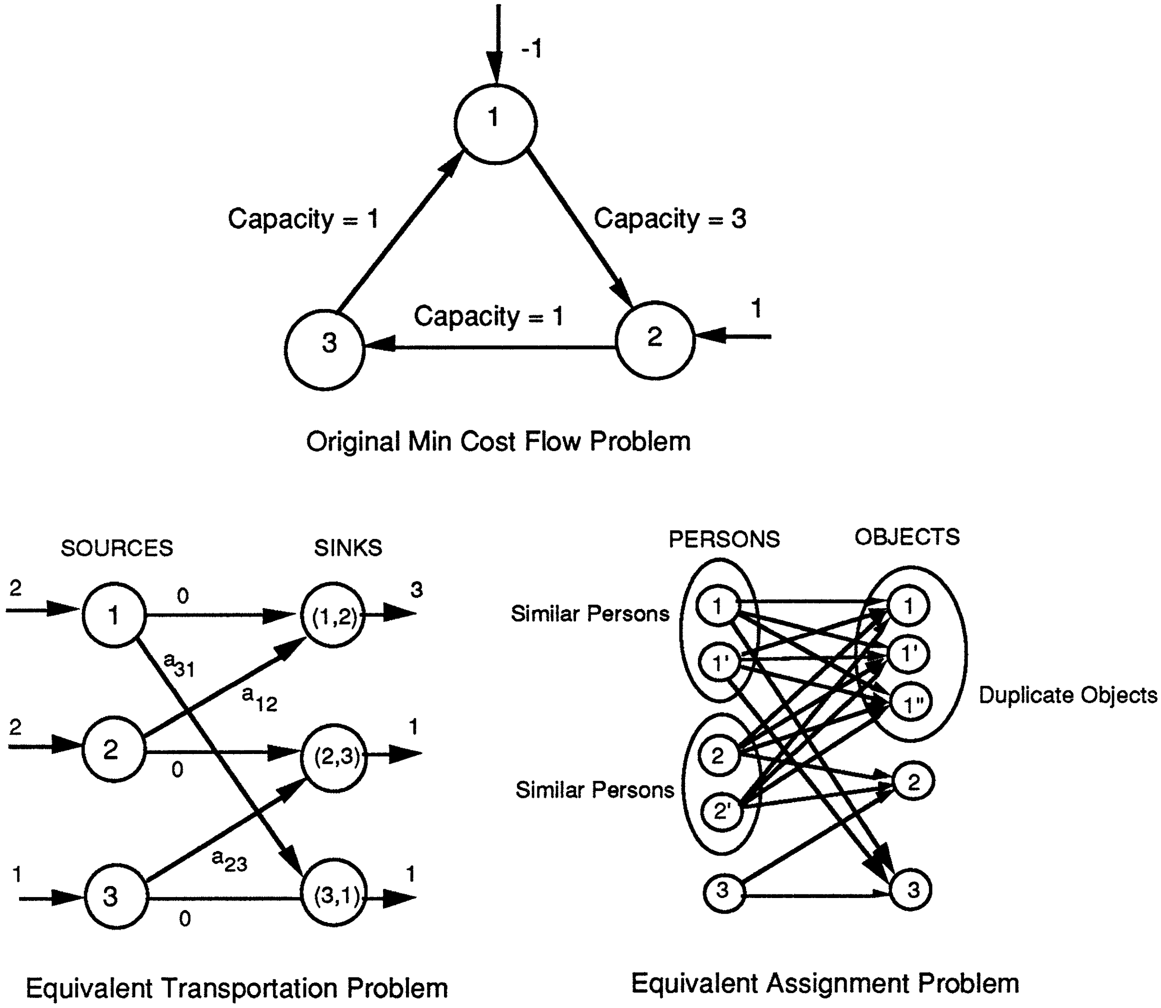

Figure 4: Example of a min cost flow problem, and its corresponding equivalent transportation and assignment problems.

in the similarity class of j [which correspond to the flow $z_{j(i,j)}$] have implied price $a_{ij} - \pi_j$. Using this fact, the ϵ-CS condition (7) is written as

$$\pi_j + (-\pi_i) \geq a_{ij} - \epsilon \qquad \text{if } y_{i(i,j)} > 0, \tag{21}$$

$$\pi_i + (a_{ij} - \pi_j) \geq -\epsilon \qquad \text{if } z_{j(i,j)} > 0. \tag{22}$$

We now introduce a *price* variable p_i for each node $i \in \mathcal{N}$, which is the negative profit π_i,

$$p_i = -\pi_i, \qquad \forall\, i \in \mathcal{N}, \tag{23}$$

and we express the flow vector x in terms of (y, z) as

$$x_{ij} = y_{i(i,j)} = c_{ij} - z_{j(i,j)}. \tag{24}$$

The conditions (21) and (22) for a profit-preassignment pair $(\pi, \overline{S})$ satisfying ϵ-CS are then written as

$$p_i \geq a_{ij} + p_j - \epsilon \qquad \text{for all } (i,j) \in \mathcal{A} \text{ with } x_{ij} > 0, \tag{25}$$

$$p_i \leq a_{ij} + p_j + \epsilon \qquad \text{for all } (i,j) \in \mathcal{A} \text{ with } x_{ij} < c_{ij}. \tag{26}$$

These are precisely the ϵ-CS conditions for the minimum cost flow problem as first introduced in connection with the ϵ-relaxation method in [Ber86a] and [Ber86b].

Table 2 provides a list of the corresponding variables and relations between the minimum cost flow problem and its equivalent transportation/assignment problem.

	Transportation/ Assignment	**Minimum Cost Flow**
Flows	$y_{i(i,j)}$, $z_{j(i,j)} = c_{ij} - y_{i(i,j)}$	$x_{ij} = y_{i(i,j)} = c_{ij} - z_{j(i,j)}$
Profits	π_i for all persons in the similarity class of node i	
Prices	$\begin{cases} -\pi_i & \text{for objects of } (i,j) \text{ paired with persons of } i \text{ via } \overline{S} \\ a_{ij} - \pi_j & \text{for objects of } (i,j) \text{ paired with persons of } j \text{ via } \overline{S} \end{cases}$	$p_i = -\pi_i$
ϵ-CS	$\pi_j + (-\pi_i) \geq a_{ij} - \epsilon$ for objects of (i,j) preferring persons of i under $\overline{S}$ $\pi_i + (a_{ij} - \pi_j) \geq -\epsilon$ for objects of (i,j) preferring persons of j under $\overline{S}$	$p_i \geq a_{ij} + p_j - \epsilon$ if $x_{ij} > 0$ $p_i \leq a_{ij} + p_j + \epsilon$ if $x_{ij} < c_{ij}$

Table 2: Correspondences between the minimum cost flow problem and its transportation/assignment equivalent version.

6. DERIVING THE ϵ-RELAXATION METHOD

We will now apply the profit-oriented auction algorithm of Section 4 (which uses M-auction iterations) to the equivalent assignment version of the minimum cost flow problem (MCF). This algorithm starts with and maintains a fair profit-preassignment pair $(\pi, \overline{S})$ satisfying ϵ-CS. Based on the equivalences of the preceding section (cf. Table 2), the corresponding minimum cost flow algorithm starts with and maintains a flow-price pair (x, p) satisfying the ϵ-CS conditions (25) and (26).

An auction iteration starts with finding a similarity class M containing some persons that are not preferred by any object under $\overline{S}$. This is equivalent to finding a node i such that the sum of outgoing flows in the equivalent transportation problem

$$\sum_{\{j|(i,j)\in\mathcal{A}\}} y_{i(i,j)} + \sum_{\{j|(j,i)\in\mathcal{A}\}} z_{i(j,i)}$$

is less than the supply

$$s_i + \sum_{\{j|(j,i)\in\mathcal{A}\}} c_{ji}.$$

The difference of the above two quantities is the surplus of node i, denoted g_i, which in view of the definition

$$y_{i(i,j)} = x_{ij}, \qquad z_{i(j,i)} = c_{ji} - x_{ji},$$

can be written as

$$g_i = \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji} - \sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij} + s_i.$$

We thus see that in the auction algorithm, each (M-auction) iteration involves a bid by a similarity class corresponding to a node i with positive surplus g_i. According to the rules of the auction algorithm, node i must:

(a) Rank the objects corresponding to its incident arcs in terms of their values.

(b) Select a sufficient number of objects to satisfy its surplus.

(c) Brings the profit π_i down to the implied value corresponding to the next best object minus ϵ [cf. Eqs. (19), (20)].

(d) Adjust the preassignment $\overline{S}$ so that it contains exactly one pair per object while each person in the similarity class of i is preferred by exactly one object out of the ones selected in (b) above.

Translating these operations in the context of the minimum cost flow problem, the iterating node i must rank order the outgoing arcs (i,j) and the incoming arcs (j,i) in terms of the values of the corresponding objects. The node must then push an increment of flow equal to its surplus along a sufficient number of arcs (in the order of their values), while lowering its own profit π_i or, equivalently, raising its own price p_i as necessary.

In particular, in order for node i to push flow to an outgoing neighbor node j along an arc (i,j), it must, by Eqs. (19), (20), set its profit π_i to the (common) implied value of the objects of arc (i,j) that prefer persons of the similarity class of j minus ϵ, which is

$$-a_{ij} + \pi_j - \epsilon.$$

Equivalently, by using the relations $p_i = -\pi_i$ and $\pi_j = -p_j$, we see that to push flow from i on an arc (i,j), we must have

$$x_{ij} < c_{ij}, \tag{27}$$

and the price p_i must be set to

$$p_i = a_{ij} + p_j + \epsilon. \tag{28}$$

When an arc (i,j) satisfies the conditions (27) and (28), it is said to be ϵ+*-unblocked*.

Similarly, in order for node i to push flow to an incoming neighbor node j along an arc (j,i) it must set its profit to

$$a_{ji} + \pi_j - \epsilon$$

or equivalently we must have

$$0 < x_{ji},$$

and the price p_i must be set to the level

$$p_i = p_j - a_{ji} + \epsilon. \tag{30}$$

When an arc (j,i) satisfies the conditions (29) and (30), it is said to be ϵ^-*-unblocked*.

We now transcribe the auction algorithm by using the correspondences derived above. At the start of each iteration, a node i with positive surplus g_i is chosen.

Typical M-Auction Iteration Applied to the Equivalent Assignment Problem

Step 1: (Scan incident arc) Select an arc a (i,j) that is an ϵ+–unblocked arc and go to Step 2, or an arc (j,i) that is ϵ^-–unblocked arc and go to Step 3. If no such arc can be found go to Step 4.

Step 2: (Push flow forward along arc (i,j)) Increase x_{ij} by $\delta = \min\{g_i, c_{ij} - x_{ij}\}$. If now $g_i = 0$ and $x_{ij} < c_{ij}$, stop; else go to Step 1.

Step 3: (Push flow backward along arc (j,i)) Decrease x_{ji} by $\delta = \min\{g_i, x_{ji}\}$. If now $g_i = 0$ and $b_{ji} < x_{ji}$, stop; else go to Step 1.

Step 4: (Increase price of node i) Increase p_i to the level

$$\min\Big\{\{p_j + a_{ij} + \epsilon \mid (i,j) \in \mathcal{A} \text{ and } x_{ij} < c_{ij}\},\ \{p_j - a_{ji} + \epsilon \mid (j,i) \in \mathcal{A} \text{ and } b_{ji} < x_{ji}\}\Big\}.$$

Go to Step 1.

This is precisely the ϵ-relaxation method first proposed in [Ber66a] and [Ber66b]. Note that Steps 2 and 3 correspond to changing the preassignment by associating the persons in the similarity class of node i to their best objects corresponding to the incident arcs of i, up to the point where the surplus of i is exhausted. This modification of the preassignment is done via perhaps multiple passes through Steps 2 and 3. Step 4 raises the price p_i to the appropriate level, so that the corresponding profit-preassignment pair of the equivalent assignment problem is fair and satisfies ϵ-CS. Step 1 checks whether there are objects that can change their preferred persons at the current profit level $(-p_i)$ of the persons corresponding to i, and switches to Step 4 to increase p_i when no such objects exist.

REFERENCES

[AMO89] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B., "Network Flows," Sloan W. P. No. 2059-88, M.I.T., Cambridge, MA, 1989, (also in Handbooks in Operations Research and Management Science, Vol. 1, Optimization, G. L. Nemhauser, A. H. G. Rinnooy-Kan, and M. J. Todd (eds.), North-Holland, Amsterdam, 1989, pp. 211-369).

[AOT89] Ahuja, R. K., Orlin, J. B., and Tarjan, R. E., "Improved Time Bounds for the Maximum Flow Problem," SIAM Journal of Computing, Vol. 18, 1989, pp. 939-954.

[AhO89] Ahuja, R. K., and Orlin, J. B., "A Fast and Simple Algorithm for the Maximum Flow Problem," Operations Research, Vol. 37, 1989, pp. 748-759.

[BeE87] Bertsekas, D. P., and Eckstein, J., "Distributed Asynchronous Relaxation Methods for Linear Network Flow Problems," Proc. of IFAC '87, Munich, Germany, July 1987.

[BeE88] Bertsekas, D. P., and Eckstein, J., "Dual Coordinate Step Methods for Linear Network Flow Problems," Math. Programming, Series B, Vol. 42, 1988, pp. 203-243.

[BeT89] Bertsekas, D. P., and Tsitsiklis, J. N., Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, Englewood Cliffs, N. J., 1989.

[Ber79] Bertsekas, D. P., "A Distributed Algorithm for the Assignment Problem," Lab. for Information and Decision Systems Working Paper, M.I.T., March 1979.

[Ber86a] Bertsekas, D. P., "Distributed Asynchronous Relaxation Methods for Linear Network Flow Problems," Lab. for Information and Decision Systems Report P-1606, M.I.T., November 1986.

[Ber86b] Bertsekas, D. P., "Distributed Relaxation Methods for Linear Network Flow Problems," Proceedings of 25th IEEE Conference on Decision and Control, 1986, pp. 2101-2106.

[Ber88] Bertsekas, D. P., "The Auction Algorithm: A Distributed Relaxation Method for the Assignment Problem," Annals of Operations Research, Vol. 14, 1988, pp. 105-123.

References

[Ber91] Bertsekas, D. P., Linear Network Optimization: Algorithms and Codes, M.I.T. Press, Cambridge, Mass., 1991

[Ber92] Bertsekas, D. P., "Auction Algorithms for Network Problems: A Tutorial Introduction," Computational Optimization and Applications, Vol. 1, 1992, pp. 7-66.

[ChM89] Cheriyan, J., and Maheshwari, S. N., "Analysis of Preflow Push Algorithms for Maximum Network Flow," SIAM J. Comput., Vol. 18, 1989, pp. 1057-1086.

[GoT86] Goldberg, A. V., and Tarjan, R. E., "A New Approach to the Maximum Flow Problem," Proc. 18th ACM STOC, 1986, pp. 136-146.

[GoT90] Goldberg, A. V., and Tarjan, R. E., "Solving Minimum Cost Flow Problems by Successive Approximation," Math. of Operations Research, Vol. 15, 1990, pp. 430-466.

[Gol85] Goldberg, A. V., "A New Max-Flow Algorithm," Tech. Mem. MIT/LCS/TM-291, Laboratory for Computer Science, M.I.T., Cambridge, MA., 1985.

[Gol87] Goldberg, A. V., "Efficient Graph Algorithms for Sequential and Parallel Computers," Tech. Report TR-374, Laboratory for Computer Science, M.I.T., Cambridge, MA., 1987.

[MPS91] Mazzoni, G., Pallotino, S., and Scutella', M. G., "The Maximum Flow Problem: A Max-Preflow Approach," European J. of Operational Research, Vol. 53, 1991.