December, 1992

LIDS- P 2150

Research Supported By:

NSF grant CCR-9103804
ARO grant DAAL03-92-G-0115

Modified Auction Algorithm for Shortest Paths

Bertsekas, D.P.

December 1992 LIDS-P-2150

MODIFIED AUCTION ALGORITHMS FOR SHORTEST PATHS[1]

by

Dimitri P. Bertsekas[2]

Abstract

Recently Cerulli, De Leone, and Piacente [CDP92] have proposed a modified auction algorithm for shortest paths, which uses larger price increases than the original method. Motivated by their algorithm, we propose an alternative modified auction algorithm that uses even larger price increases. This algorithm is the naive auction algorithm applied to an equivalent assignment problem.

[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. THE ORIGINAL AUCTION ALGORITHM

Suppose that we have a graph with node set $\mathcal{N}$, arc set $\mathcal{A}$, and a length a_{ij} for each arc (i, j). By a *path* we mean a sequence of nodes $(i_1, i_2, \ldots, i_k)$ such that (i_m, i_{m+1}) is an arc for all $m = 1, \ldots, k-1$. If the nodes $i_1, i_2, \ldots, i_k$ are distinct, the sequence $(i_1, i_2, \ldots, i_k)$ is called a *simple path*. The length of a path is defined to be the sum of its arc lengths. Assuming that all cycles have positive length, we want to find a path of minimum length over all paths that start at a given origin (node 1) and end at a given destination (node t). We assume for convenience that the origin node has at least two outgoing arcs, and that each node has at least one outgoing arc to a node other the origin.

The original auction algorithm for shortest paths, as given in [Ber91a] and [Ber91b], maintains at all times a simple path $P = (1, i_1, i_2, \ldots, i_k)$. If i_{k+1} is a node that does not belong to P and (i_k, i_{k+1}) is an arc, *extending* P *by* i_{k+1} means replacing P by the path $(1, i_1, i_2, \ldots, i_k, i_{k+1})$. If P does not consist of just the origin node 1, *contracting* P means replacing P by the path $(1, i_1, i_2, \ldots, i_{k-1})$. The algorithm maintains also a price vector p satisfying together with P the following property:

$$p_i \leq a_{ij} + p_j, \qquad \forall\ (i, j) \in \mathcal{A}, \tag{1a}$$

$$p_i = a_{ij} + p_j, \qquad \forall\ (i, j) \in P, \tag{1b}$$

which is referred to as *complementary slackness* (CS for short). If the arc lengths are nonnegative, the zero price vector together with the trivial path $P(1)$ satisfy CS and can be used to initialize the algorithm. Otherwise the choice of an initial pair (P, p) satsfying CS may be complicated.

The algorithm proceeds in iterations, transforming a pair (P, p) satisfying CS into another pair satisfying CS. The iteration is as follows.

Typical Iteration of the Auction/Shortest Path Algorithm

Let i be the terminal node of P. If

$$p_i < \min_{\{j \mid (i,j) \in \mathcal{A}\}} \{a_{ij} + p_j\},$$

go to Step 1; else go to Step 2.

Step 1 (Contract path): Set

$$p_i := \min_{\{j \mid (i,j) \in \mathcal{A}\}} \{a_{ij} + p_j\},$$

and if $i \neq 1$, contract P. Go to the next iteration.

Step 2 (Extend path): Extend P by node j_i where

$$j_i = \arg \min_{\{j \mid (i,j) \in \mathcal{A}\}} \{a_{ij} + p_j\}.$$

If j_i is the destination t, stop; P is the desired shortest path. Otherwise, go to the next iteration.

2. THE EQUIVALENT ASSIGNMENT PROBLEM

Let $1, 2, \ldots, N$ be the nodes of the graph. We consider the following assignment problem, which is equivalent to the shortest path problem. This equivalence is described in [Ber91a], [Ber91b], and is repeated here for convenience.

Let $2, \ldots, N$ be the "object" nodes of the assignment problem, and for each node $i \neq t$, introduce a "person" node i'. For every arc (i,j) of the shortest path problem with $i \neq t$ and $j \neq 1$, introduce the arc (i', j) with cost a_{ij} in the assignment problem. Introduce also the zero cost arc (i', i) for each $i \neq 1, t$.

It can be shown (see [Ber91a] or [Ber91b]) that the auction/shortest path algorithm is equivalent to applying the naive auction algorithm to this assignment problem, starting from a price vector $(p_2, \ldots, p_N)$ satisfying the CS condition

$$p_i \leq a_{ij} + p_j, \qquad \forall\ (i,j) \in \mathcal{A},\ i \neq 1, \tag{2}$$

and the partial assignment

$$(i', i), \qquad \forall\ i \neq 1, t.$$

The equivalence holds under two restrictions/modifications in the naive auction algorithm:

(1) If at some iteration involving the unassigned person i', the arc (i', i) is the best arc and is equally desirable with some other arc (i', j_i) (i.e., $p_i = a_{ij_i} + p_{j_i} = \min_{\{j \mid (i,j) \in \mathcal{A}\}} \{a_{ij} + p_j\}$), then the tie is broken in favor of the latter arc, that is, (i', j_i) is added to the assignment rather than (i', i).

(2) The naive auction iteration as it applies to the bid of person $1'$ (and only person $1'$) is modified so that the bid of $1'$ consists of finding an object j_1 attaining the minimum in

$$\min_{\{j \mid (1,j) \in \mathcal{A}\}} \{a_{1j} + p_j\},$$

assigning j_1 to $1'$, and deassigning the person assigned to j_1 (in the case $j_1 \neq t$), but *not* changing the price p_{j_1}. Thus the second best level $\min_{(1,j)\in\mathcal{A}, j\neq j_1}\{a_{1j} + p_j\}$ is not used in bids of person $1'$.

As explained in [Ber91a] and [Ber91b], the equivalence of this (slightly modified) naive auction algorithm to the auction/shortest path can be verified by showing inductively the following:

(a) Each assignment generated by the naive auction algorithm consists of a sequence of the form

$$(1', i_1), (i_1', i_2), \ldots, (i_{k-1}', i_k),$$

together with the additional arcs

$$(i', i), \quad \text{for } i \neq i_1, \ldots, i_k, t,$$

and corresponds to a path $P = (1, i_1, \ldots, i_k)$ generated by the shortest path algorithm. As long as $i_k \neq t$, the (unique) unassigned person in the naive auction algorithm is person i_k', corresponding to the terminal node of the path. When $i_k = t$, a feasible assignment results, in which case the naive auction algorithm terminates, consistently with the termination criterion for the shortest path algorithm.

(b) In an iteration corresponding to an unassigned person i' with $i \neq 1$, the arc (i', i) is always a best arc; this is a consequence of the complementary slackness condition (2). Furthemore, there are three possibilities: (1) (i', i) is the unique best arc, in which case (i', i) is added to the assignment, and the price p_i is increased by

$$\min_{\{j|(i,j)\in\mathcal{A}\}}\{a_{ij} + p_j\} - p_i; \tag{3}$$

this corresponds to contracting the current path by the terminal node i. (2) There is an arc (i', j_i) with $j_i \neq t$, which is equally preferred to (i', i), that is,

$$p_i = a_{ij_i} + p_{j_i},$$

in which case, in view of the tie-breaking rule specified earlier, (i', j_i) is added to the assignment and the price p_{j_i} remains the same. Furthermore, the object j_i must have been assigned to j_i' at the start of the iteration, so adding (i', j_i) to the assignment [and removing (j_i', j_i)] corresponds to extending the current path by node j_i. (3) The arc (i', t) is equally preferred to (i', i), in which case the heretofore unassigned object t is assigned to i', thereby terminating the naive auction algorithm; this corresponds to the destination t becoming the terminal node of the current path, thereby terminating the shortest path algorithm.

3. THE NEW MODIFIED AUCTION ALGORITHM

The new modified algorithm is just the naive auction algorithm applied to the preceding equivalent assignment problem, but without the modification in the bid of person $1'$ described above. In particular, the bid of $1'$ will consist of finding an object j_1 attaining the minimum in

$$\min_{\{j \mid (1,j) \in \mathcal{A}\}} \{a_{1j} + p_j\},$$

assigning j_1 to $1'$, and deassigning the person assigned to j_1 (in the case $j_1 \neq t$), *and also changing the price* p_{j_1} *by the difference between the best level* $\min_{\{j \mid (1,j) \in \mathcal{A}\}} \{a_{1j} + p_j\}$ *and the second best level* $\min_{\{j \mid (1,j) \in \mathcal{A}, j \neq j_1\}} \{a_{1j} + p_j\}$, according to the rules of the naive auction algorithm. This increase of p_{j_1} in the extension from the origin to j_1 allows price increases in subsequent extensions.

We now translate the operations of the naive auction algorithm in the terms of the shortest path problem. The algorithm maintains at all times a simple path $P = (1, i_1, i_2, \ldots, i_k)$ and a price vector p with $p_1 = \infty$. However, the pair (P, p) does not satisfy the CS condition (1). Instead, it satisfies the conditions

$$\pi_i = a_{ij} + p_j, \qquad \forall\ (i,j) \in P, \tag{4}$$

$$\pi_i = p_i, \qquad \forall\ i \notin P, \tag{5}$$

where

$$\pi_i = \min \left\{ p_i,\ \min_{\{j \mid (i,j) \in \mathcal{A}\}} \{a_{ij} + p_j\} \right\}, \qquad \forall\ i \in \mathcal{N}. \tag{6}$$

Note that the conditions (4)-(6) are the complementary slackness conditions for the equivalent assignment problem. As in the preceding section, the algorithm can be initialized with the trivial path $P = (1)$ and any price vector p satisfying $p_i \leq a_{ij} + p_j$ for all $(i,j) \in \mathcal{A}$ with $i \neq 1$. It terminates when the destination t becomes the terminal node of P.

To describe the typical iteration of the algorithm, define for each node $i \in \mathcal{N}$

$$A(i) = \{j \mid (i,j) \in \mathcal{A}\} \cup \{i\}, \tag{7}$$

$$a_{ii} = 0. \tag{8}$$

Typical Iteration of the Modified Auction/Shortest Path Algorithm

Let i be the terminal node of P, and let j_i be such that

$$j_i = \arg\min_{j \in A(i)} \{a_{ij} + p_j\}, \tag{9}$$

with the extra requirement that $j_i \neq i$ whenever possible, that is, $j_i \neq i$ whenever the minimum above is attained for some $j \neq i$. Set

$$p_{j_i} := \min_{j \in A(i), j \neq j_i} \{a_{ij} + p_j\} - a_{ij_i}. \tag{10}$$

If $j_i = i$ contract P; otherwise extend P by node j_i.

Note that when P extends from the terminal node i to a neighbor j_i, the price p_{j_i} may be increased strictly, while in the original auction algorithm there is no price change. This is also a difference from the Cerulli-De Leone-Piacente algorithm where in the case of an extension, the price p_i may be increased (rather than p_{j_i}).

The following properties can be verified:

(a) The CS conditions (4)-(6) are maintained by the algorithm. To see this, note that if the iteration is a contraction, the terminal node price p_i increases to the level

$$\min_{\{j | (i,j) \in A\}} \{a_{ij} + p_j\},$$

so that the condition $\pi_i = p_i$ [cf. Eq. (5)] is satisfied following the contraction. If the iteration is an extension to j_i, p_{j_i} is raised to a level such that

$$a_{ij_i} + p_{j_i} = \min_{j \in A(i), j \neq j_i} \{a_{ij} + p_j\} \leq p_i,$$

so that following the extension, the condition $\pi_i = a_{ij_i} + p_{j_i}$ is satisfied [cf. Eq. (4)].

(b) P is a shortest path between its endnodes. One way to prove this is to use the equivalent assignment problem. For a direct proof, note that if i is the terminal node of P, we have using Eq. (4)

$$\text{Length of } P = \sum_{(m,n) \in P} a_{mn} = \sum_{(m,n) \in P} (\pi_m - p_n) = \pi_1 - p_i + \sum_{\substack{k \in P \\ k \neq 1,i}} (\pi_k - p_k). \tag{11}$$

On the other hand if $\tilde{P}$ is any other path from 1 to i, we have from Eq. (6),

$$\text{Length of } \tilde{P} = \sum_{(m,n) \in \tilde{P}} a_{mn} \geq \sum_{(m,n) \in \tilde{P}} (\pi_m - p_n) = \pi_1 - p_i + \sum_{\substack{k \in \tilde{P} \\ k \neq 1,i}} (\pi_k - p_k).$$

From the last two relations we obtain

$$\text{Length of } \tilde{P} - \text{ Length of } P = \sum_{\{k | k \in \tilde{P}, k \notin P\}} (\pi_k - p_k) - \sum_{\{k | k \in P, k \notin \tilde{P}\}} (\pi_k - p_k).$$

Since $\pi_k = p_k$ for all $k \notin P$ [cf. Eq. (5)], and $\pi_k \leq p_k$ for all k [cf. Eq. (6)], we obtain

$$\text{Length of } \tilde{P} - \text{ Length of } P \geq 0.$$

Thus P cannot be longer that any path $\tilde{P}$ having the same endnodes.

(c) In the case of an extension, we have $j_i \notin P$, so that P remains a simple path. To see this, note that if, with the addition of j_i to P, a cycle C is formed, then by adding Eq. (4) along the arcs of C, and by using the fact $\pi_k \leq p_k$ for all k [cf. Eq. (6)], we see that

$$\text{Length of } C = \sum_{(m,n)\in C} a_{mn} = \sum_{(m,n)\in C} (\pi_m - p_n) \leq 0.$$

This contradicts our assumption that all cycles have positive length.

(d) If an extension to j_i produces an increase δ of the price p_{j_i}, there will be no contraction at j_i until all nodes in each path starting at i and having at the time of the extension reduced cost length less or equal to δ have been visited by the algorithm. The proof of this result is a little lengthy and is ommited. The result is significant, however, because it shows that the modified algorithm is capable of a lot more successive extensions than the original.

(e) If no price increase occured at some iteration, which necessarily must be an extension from the terminal node i, there will be no price increases in subsequent extensions up to the time when i becomes again the terminal node of P. To see this, note that if an extension occurs from i to j_i and p_{j_i} does not increase, we will have following the extension $\pi_{j_i} = p_{j_i}$, which by using Eq. (6) implies that $p_{j_i} \leq a_{j_i k} + p_k$ for all outgoing arcs (j_i, k). Therefore, if a further extension occurs from j_i to some node k, it is seen from Eq. (10) that the price p_k will not increase.

(f) As a corollary of (e) above, we see that if the extensions from the origin involve no price increase, there will be no price increase at any extension. In the original auction/shortest path algorithm, the condition of no price increase when performing an extension at the origin was artificially enforced by modifying the naive auction algorithm. As a result all extensions involve no price increase.

(g) From (f) above, we see that if we convert the problem to an equivalent shortest path problem where all extensions at the origin involve no price increase, the original and the modified algorithms will behave identically. This can be essentially done by duplicating the outgoing arcs of the origin node 1 [that is, for each arc $(1, j)$, we add a node j', an arc $(1, j')$ with length $a_{1j'} = a_{1j}$ and an arc (j', j) with length $a_{j'j} = 0$]. Then if we view extensions along the artificial path $(1, j', j)$ as equivalent to extensions along $(1, j)$, the modified auction algorithm and the original auction algorithm when applied to this problem behave essentially identically. This observation implies in particular that the modified algorithm cannot have better complexity than the original.

4. RELATION TO THE CERULLI-DE LEONE-PIACENTE ALGORITHM

The modified auction algorithm is actually not too different from the Cerulli-De Leone-Piacente algorithm, even though in the case of an extension from i to j_i, the former algorithm increases the price of j_i while the latter algorithm increases the price of i. This can be understood by realizing that the Cerulli-De Leone-Piacente algorithm *works in effect with the variables* π_i *of Eq. (6) rather than the variables* p_i.

Indeed one can equivalently describe our modified algorithm in terms of the variables π_i, using Eqs. (4)-(6), and when this is done, the algorithm becomes quite similar to the Cerulli-De Leone-Piacente algorithm. Here is the alternative description of the typical iteration of our algorithm for the case where the terminal node of P is not the origin.

Alternative Description of the Typical Iteration of the Modified Auction/Shortest Path Algorithm

Let i be the terminal node of P, and for each node $j \neq 1$ of P, let k_j be the predesessor node of j in P. Let j_i be such that

$$j_i = \arg\min \left\{ \min_{j \in A(i), j \in P, j \neq 1} \{a_{ij} + \pi_{k_j} - a_{k_j j}\}, \min_{j \in A(i), j \notin P} \{a_{ij} + \pi_j\} \right\}, \tag{12}$$

with the extra requirement that $j_i \neq i$ whenever possible, that is, $j_i \neq i$ whenever the minimum above is attained for some $j \neq i$. Set

$$\pi_i := \text{ second best value in the minimum of Eq. (12).}$$

If $j_i = i$ contract P; otherwise extend P by node j_i.

If the terminal node of P is the origin, the iteration is similar but the minimization in Eq. (12) does not involve the term $\pi_{k_i} - a_{k_i i}$.

It can be seen now that the difference between the above algorithm and the Cerulli-De Leone-Piacente algorithm is that the latter algorithm uses the expression

$$\min \left\{ \pi_{k_i} - a_{k_i i}, \min_{\{j \mid (i,j) \in \mathcal{A}, j \neq 1\}} \{a_{ij} + \pi_j\} \right\}, \tag{13}$$

in place of the expression

$$\min \left\{ \min_{j \in A(i), j \in P, j \neq 1} \{a_{ij} + \pi_{k_j} - a_{k_j j}\}, \min_{j \in A(i), j \notin P} \{a_{ij} + \pi_j\} \right\} \tag{14}$$

of Eq. (12). The expressions (13) and (14) are equal because j_i cannot belong to P, but *the corresponding second best values need not be equal.* The second best value in Eq. (13) may be smaller than the second best value in Eq. (14), since by Eqs. (4) and (6), we have for the nodes $j \in P$ with $j \neq 1$

$$\pi_{k_j} - a_{k_j j} = p_j \geq \pi_j.$$

A difference between the two algorithms arises only in iterations where the second best node j_i'

$$j_i' = \arg \min_{j \in A(i), j \neq j_i} \{a_{ij} + p_j\}$$

belongs to P. Then our modified algorithm makes a potentially larger price increase than the Cerulli-De Leone-Piacente algorithm. Note also that when graph reduction is used, our modified algorithm and the Cerulli-De Leone-Piacente algorithm behave identically, because the arcs (i,j) with i being the terminal node of P and $j \in P$ are deleted when graph reduction is used.

5. TERMINATION PROOF

We now establish that for a feasible problem the algorithm terminates with a shortest path from 1 to t.

Proposition 1: Assume that there exists at least one path from 1 to t. Then the modified auction/shortest path algorithm terminates with a shortest path from 1 to t.

Proof: Assume, in order to arrive at a contradiction, that the algorithm iterates indefinitely. Then we will show that the price of at least one node $i \neq 1$ will tend to ∞. To this end, we make the following observations:

(a) If an extension is performed along arc (i,j), we must have following the extension

$$p_i \geq a_{ij} + p_j. \tag{15}$$

This is evident from the algorithm, since following the extension we have $p_i \geq \pi_i = a_{ij} + p_j$ [cf. Eqs. (4) and (5)].

(b) If a node i belongs to P in infinitely many iterations, there must exist an outgoing arc (i,j) such that $j \in P$ in infinitely many iterations, and Eq. (15) holds at the end of infinitely many iterations. To see this, note that there are two possibilities: (1) i is the terminal node of P at finitely many iterations. Then following the last extension from i to, say, node j, the prices p_i and p_j stay unchanged and $j \in P$, while we have by observation (a) above that Eq. (15) holds. (2) i is the terminal node of P in infinitely many iterations. Then either an extension is performed along some arc (i, j) in infinitely many iterations, in which case $j \in P$ and Eq. (15) holds at the end of these iterations, or else a contraction is performed at i in infinitely many iterations. In the latter case, p_i is increased an infinite number of times to $\min_{\{j|(i,j)\in\mathcal{A}\}}\{a_{ij}+p_j\}$. Therefore there must exist some outgoing arc (i, j) such that $p_i = a_{ij} + p_j$ in infinitely many iterations, while p_j is increased in infinitely many iterations, implying that $j \in P$ in infinitely many iterations.

Let us start with $i_1 = 1$, and given i_k, let us select i_{k+1} such that $i_{k+1} \in P$ in infinitely many iterations and

$$p_{i_k} \geq a_{i_k i_{k+1}} + p_{i_{k+1}} \tag{16}$$

in infinitely many iterations. Such a selection is possible by observation (b) above. In this way some node will be repeated and a cycle $C = (i_m, i_{m+1}, \ldots, i_n, i_m)$ not containing node 1 will be formed. If the prices of all nodes $i \neq 1$ converge to some finite limits $\overline{p}_i$, by taking limit in Eq. (16), we obtain

$$\overline{p}_i \geq a_{ij} + \overline{p}_j, \qquad \forall\, (i, j) \in C. \tag{17}$$

By adding this equation over all $(i, j) \in C$, we obtain that the length of the cycle C is nonpositive, which contradicts our assumptions. Therefore, some price p_i with $i \neq 1$ tends to ∞.

Let $\mathcal{N}^\infty$ be the nonempty subset of nodes $i \neq 1$ such that p_i tends to ∞. Each node $i \in \mathcal{N}^\infty$ is the terminal node of P in infinitely many iterations. From Eq. (11) we have for these iterations

$$\text{Length of } P \geq \pi_1 - p_i,$$

so it follows that π_1 tends to ∞ and that all nodes $i \neq 1$ that belong to P in infinitely many iterations belong to $\mathcal{N}^\infty$. Since

$$\pi_1 = \min_{\{j|(1,j)\in\mathcal{A}\}}\{a_{1j} + p_j\},$$

and π_1 tends to ∞, all outgoing neighbors of node 1 belong to $\mathcal{N}^\infty$. Also, each node $i \in \mathcal{N}^\infty$ experiences an infinite number of price changes through a contraction, so we must have $p_i \leq a_{ij} + p_j$ in infinitely many iterations for all outgoing arcs (i, j). Thus all outgoing neighbors of all nodes in $\mathcal{N}^\infty$ belong to $\mathcal{N}^\infty$. We conclude that there is no arc with start node in $\{1\} \cup \mathcal{N}^\infty$ and end node not in $\{1\} \cup \mathcal{N}^\infty$. Since the destination t does not belong to $\{1\} \cup \mathcal{N}^\infty$, it follows that there is no path from 1 to t, which contradicts our hypothesis. **Q.E.D.**

REFERENCES

[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.

[Ber91a] Bertsekas, D. P., Linear Network Optimization: Algorithms and Codes, M.I.T. Press, Cambridge, Mass., 1991

[Ber91b] Bertsekas, D. P., "The Auction Algorithm for Shortest Paths," SIAM J. on Optimization, Vol. 1, 1991, pp. 425-447.

[CDP92] Cerulli, R., De Leone, R., and Piacente, G., "A Modified Auction Algorithm for the Shortest Path Problem," University of Salerno Report, Preprint.