A MULTIPLIER ADJUSTMENT METHOD FOR MULTIPLE SHORTEST PATH PROBLEMS WITH COUPLING CONSTRAINTS

Nejat Karabakal
and
James C. Bean
Department of Industrial & Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109-2117

Technical Report 95-11

June 1995

A Multiplier Adjustment Method for Multiple Shortest Path Problems with Coupling Constraints*

Nejat Karabakal and James C. Bean
Department of Industrial and Operations Engineering
University of Michigan, Ann Arbor, Michigan 48109-2117

June 28, 1995

Abstract

We describe a problem that requires joint determination of multiple shortest paths due to some shared resource constraints. A common procedure is to relax the resource constraints and solve the Lagrangian dual by a multiplier adjustment method (MAM). We describe a general MAM that unifies various features of similar methods for specific problems in the literature. The MAM always achieves monotonic ascent and, in some cases, steepest ascent with little extra computation. We discuss how to adopt the general MAM into various applications, including equipment replacement, production planning, and pavement maintenance. We use the generalized assignment problem to illustrate the steepest ascent feature of the MAM.

*This research was supported in part by National Science Foundation grant DDM-9202849 to the University of Michigan. Previous version of this paper was presented at the 1994 ORSA/TIMS meeting in Detroit.

1 Introduction

Many important decision problems, such as equipment replacement and production planning, can be formulated as acyclic shortest path problems that are efficiently solved by dynamic programming (DP). Difficulties arise, however, in implementing such solutions if there are limited resources to be shared by shortest path problems, such as budgets and storage space. For example, in fleet vehicle replacement, each individual vehicle replacement induces a shortest path. A capital expenditure budget links all individual problems [19].

The resource-constrained shortest path problem is an NP-hard combinatorial optimization problem [14]. A common way to handle resource constraints is to relax them using Lagrange multipliers and solve the Lagrangian dual problem [2, 14]. Although there is no guarantee to solve the original problem optimally, this approach is attractive for two reasons. First, shortest paths obtained by this approach are sometimes acceptable when slight resource violations are considered tolerable, for example, under soft capital rationing [17, 15]. Second, the Lagrangian dual problem provides a lower bound which can be used in implicit enumeration algorithms [1, 2, 7, 8, 12, 19].

In this paper, we describe a general multiplier adjustment method (MAM) for solving the Lagrangian dual. The MAM unifies various aspects of similar methods designed for the specific applications of the resource-constrained shortest path problem [8, 12, 13, 17, 19, 22], and applies the general structure to new specific problems. Since the most difficult part of a MAM is usually the determination of a step size, for given a set of ascent directions, most existing methods select the first ascent direction that gives a nonzero step size. A novel feature of our method is its efficiency in calculating the step size along any given ascent direction. With this feature, selecting the steepest ascent direction is justified computationally in many applications that satisfy some mild network conditions.

In the following sections, we first formulate the multiple acyclic shortest path problem

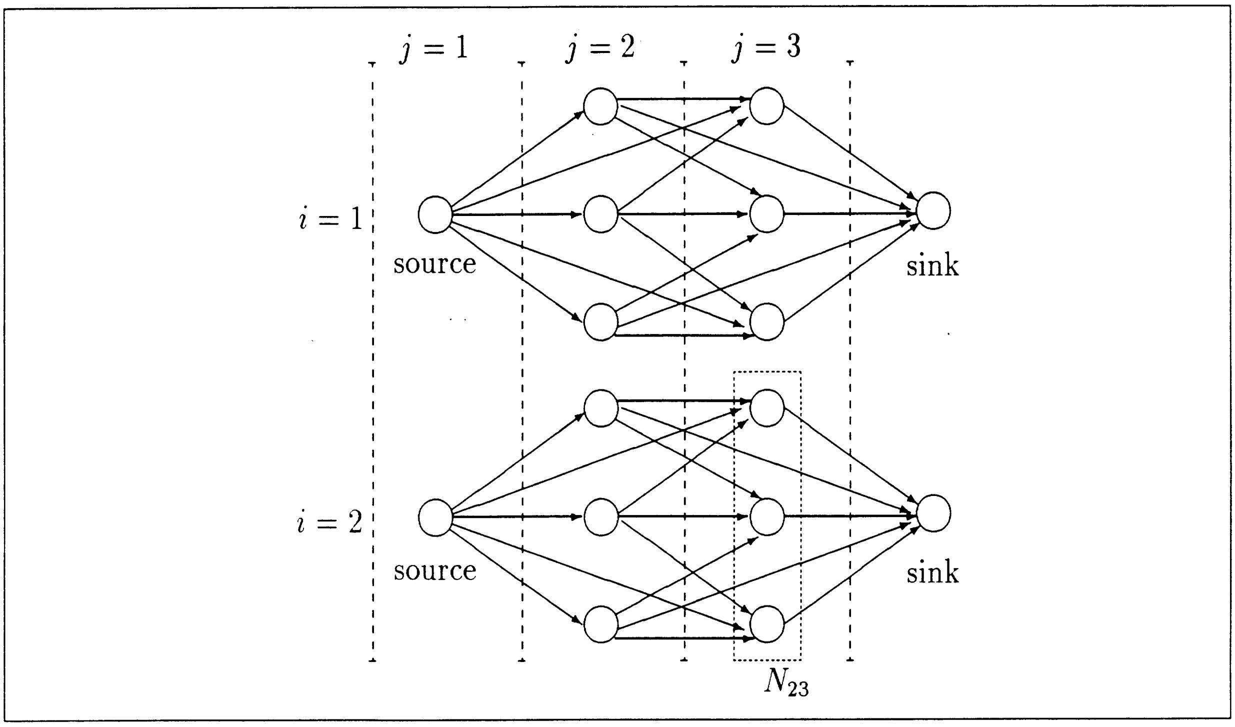

Figure 1: Two shortest path problems with three common resources.

with complicating resource constraints. We then present an algorithm for the MAM and establish its properties through a number of lemmas. Next, we discuss how to adopt the general MAM for various applications. We illustrate the steepest ascent property through the generalized assignment problem. Finally, we present the concluding remarks.

2 The Model

Consider m shortest path problems that must be solved jointly due to n common resources. Let the i-th problem be defined over a directed, acyclic graph $G_i = (N_i, A_i)$, $i = 1, \ldots, m$, as the problem of finding a shortest path between a source and sink node. Each N_i can be partitioned into n disjoint node sets such that $N_i = \bigcup_{j=1}^{n} N_{ij}$, and all arcs emanating from N_{ij} consume resource j. Essentially, each G_i is a DP network with resources corresponding to stages, and the set N_{ij} corresponding to the states in stage j. Figure 1 illustrates a case where two shortest path problems must be solved jointly, each using three common resources.

Let

$$
\begin{aligned}
c(i,k,\ell) &= \text{length of arc } (k,\ell) \in A_i \\
a(i,k,\ell) &= \text{resource } j \text{ consumption of arc } (k,\ell) \in A_i,\ k \in N_{ij} \text{ (} j \text{ implicit} \\
&\quad\ \text{from } k\text{)} \\
b(j) &= \text{resource } j \text{ availability/requirement} \\
x(i,k,\ell) &= \begin{cases} 1 & \text{if arc } (k,\ell) \in A_i \text{ is on the optimal path in } G_i \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

We can formulate this problem as an integer program (IP) as follows. For the ease of exposition, add to each A_i a dummy arc with zero length from the sink node to the source node.

$$
\text{minimize} \quad \sum_{i=1}^{m} \sum_{(k,\ell)\in A_i} c(i,k,\ell)\, x(i,k,\ell)
$$

subject to

$$
\sum_{\ell\in N_i} x(i,k,\ell) - \sum_{\ell\in N_i} x(i,\ell,k) = 0 \qquad i = 1,\ldots,m \tag{1}
$$

$$
\sum_{i=1}^{m} \sum_{\substack{(k,\ell)\in A_i \\ k\in N_{ij}}} a(i,k,\ell)\, x(i,k,\ell) \left\{ \begin{matrix} \leq \\ = \\ \geq \end{matrix} \right\} b(j) \qquad j = 1,\ldots,n \tag{2}
$$

$$
x(i,k,\ell) \in \{0,1\} \quad \forall i,k,\ell \tag{3}
$$

Network constraints (1) assure the conservation of unit flow for each i. Resource constraints (2) link the individual shortest path problems, rendering an NP-complete problem [14].

Let $\lambda \in \Re^n$ be a vector of Lagrange multipliers for constraint set (2) with the following sign conventions.

λ	resource constraint type
≥ 0	$\leq$
unrestricted in sign	$=$
≤ 0	$\geq$

Dualizing all resource constraints yields the following relaxation.

$$L(\lambda) = \text{minimize} \sum_{i=1}^{m}\sum_{j=1}^{n} \sum_{(k,\ell)\in A_i} \bar{c}_\lambda(i,k,\ell)\, x(i,k,\ell) - \sum_{j=1}^{n} b(j)\,\lambda(j)$$

subject to (1) and (3), where $\bar{c}_\lambda(i,k,\ell) = c(i,k,\ell) + a(i,k,\ell)\,\lambda(j)$, $k \in N_{ij}$.

Without resource constraints, individual shortest path problems can be solved efficiently by a DP recursion [6]. Let $f_\lambda(i)$ be the length of the shortest path in G_i for a given λ. Then

$$L(\lambda) = \sum_{i=1}^{m} f_\lambda(i) - \sum_{j=1}^{n} b(j)\,\lambda(j) \tag{4}$$

For a given λ, $L(\lambda)$ is a lower bound on the optimal value to the IP. The tightest bound is obtained by solving the Lagrangian dual problem.

$$\max_{\lambda}\ L(\lambda) \tag{5}$$

In the next section, we present a MAM for solving (5).

3 Multiplier Adjustment Method

MAMs are heuristic algorithms for solving Lagrangian duals by exploiting the special structure of a particular problem. A MAM has the advantage of achieving monotonic bound improvement in most applications unlike the more common subgradient algorithms. However, a MAM does not guarantee bounds better than those obtained by subgradient algorithms. For details and successful application examples of the MAMs, see [5, 7, 8, 19, 22].

Although most MAMs use the violated constraints to select an ascent direction, they differ considerably in finding the step size along a chosen direction. Based on the total float concept borrowed from the critical path method, our MAM features an efficient procedure for determining exact step sizes. Under some conditions, we can guarantee steepest ascent. We discuss these issues in detail following the formal presentation of the MAM.

We now present a generic framework for the MAM.

Step 1. Pick initial multipliers, λ^0. With $\bar{c}_{\lambda^0}$, solve m shortest path problems and obtain x_{λ^0}. Set $t \leftarrow 0$.

Step 2. If x_{λ^t} satisfies (2), stop; otherwise select a violated constraint j^* from constraint set (2).

Step 3. Let LHS(j^*) be the left hand side of constraint j^*. Determine a step size, $\Delta > 0$, as follows.

(a) If the LHS(j^*)> $b(j^*)$ for "$\leq$" or "=" type constraints, increase $\lambda^t(j^*)$ by Δ until one of the following arcs is deselected.

$$\{(k,\ell) \in N_{ij^*} \mid x_{\lambda^t}(i,k,\ell) = 1\}.$$

(b) If the LHS(j^*)< $b(j^*)$ for "$\geq$" or "=" type constraints, decrease $\lambda^t(j^*)$ by Δ until one of the following arcs is selected.

$$\{(k,\ell) \in N_{ij^*} \mid x_{\lambda^t}(i,k,\ell) = 0\}.$$

If finding $\Delta > 0$ is not possible, stop; otherwise, adjust multipliers

$$\lambda^{t+1}(j^*) \leftarrow \begin{cases} \lambda^t(j^*) + \Delta & \text{if case (a)} \\ \lambda^t(j^*) - \Delta & \text{if case (b)} \end{cases} \qquad \lambda^{t+1}(j) \leftarrow \lambda^t(j) \quad \text{for } j \neq j^*.$$

With $c_{\lambda^{t+1}}$, solve m shortest path problems to obtain $x_{\lambda^{t+1}}$. Set $t \leftarrow t+1$, go to step 2.

The most critical step of the MAM is the determination of the step size, Δ. We start with some definitions.

Definition 1 *Given a Lagrangian solution x_λ, define*

$$\begin{aligned} I^1_\lambda(j) &= \{i \mid x_\lambda(i,k,\ell) = 1,\ k \in N_{ij},\ \textit{for some } \ell\},\ \textit{and} \\ I^0_\lambda(j) &= \{i \mid x_\lambda(i,k,\ell) = 0,\ k \in N_{ij},\ \textit{for some } \ell\}. \end{aligned}$$

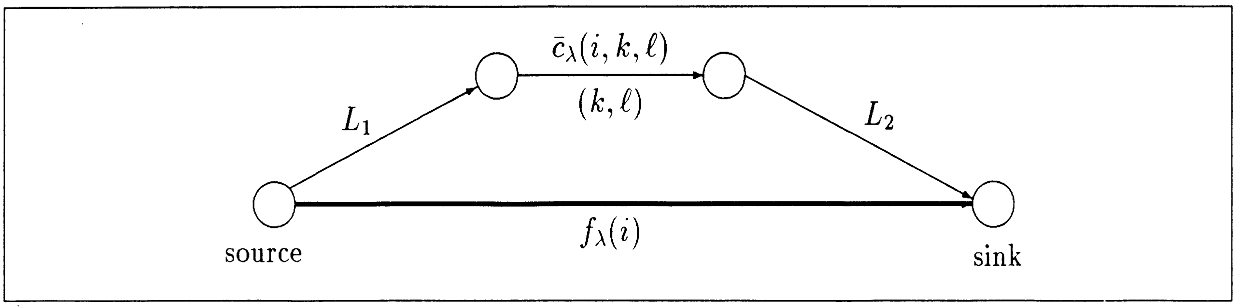

Figure 2: Determination of $\theta_\lambda(i,k,\ell)$.

Definition 2 (Slack of an arc) *Given $x_\lambda(i,k,\ell) = 0$ in the current solution, define $\theta_\lambda(i,k,\ell)$ as the minimum decrease in $\bar{c}_\lambda(i,k,\ell)$ necessary to set $x_\lambda(i,k,\ell) = 1$. We refer to $\theta_\lambda(i,k,\ell)$ as the slack of arc (k,ℓ) in G_i.*

The value of $\theta_\lambda(i,k,\ell)$ can be determined by a procedure similar to that of the critical path method for finding *total floats* in activity networks. This concept is related to sensitivity in dynamic programming as discussed in [20]. The idea is illustrated in Figure 2. Let L_1 and L_2 be the shortest path lengths from the source node to the tail of (k,ℓ) and from the sink node to the head of arc (k,ℓ), respectively. The lengths L_1 and L_2 can be computed efficiently by forward and backward DP recursions [6]. The value of $\theta_\lambda(i,k,\ell)$ is the difference between the length of a shortest path from source to sink constrained to include (k,ℓ), $\tilde{f}_\lambda(i,k,\ell)$, and the length of the current shortest path, $f_\lambda(i)$.

Lemma 1 *Given $x_\lambda(i,k,\ell) = 0$ in the current solution,*

$$\theta_\lambda(i,k,\ell) = \tilde{f}_\lambda(i,k,\ell) - f_\lambda(i) \geq 0$$

where $\tilde{f}_\lambda(i,k,\ell) = L_1 + \bar{c}_\lambda(i,k,\ell) + L_2$.

Hence, we have

Lemma 2 *Consider a Lagrangian solution x_λ for a given λ. If constraint j^* is violated because consumption is less than what is required, i.e., LHS(j^*)$<$ $b(j^*)$, then the step size in* *step 3 of the MAM can be computed*

$$\Delta = \min_{\substack{i \in I_\lambda^0(j^*) \\ (k,\ell) \in A_i,\, k \in N_{ij^*}}} \left\{ \frac{\theta_\lambda(i,k,\ell)}{a(i,k,\ell)} \right\}. \tag{6}$$

Proof: Consider $x_\lambda(i,k,\ell) = 0$, $i \in I_\lambda^0(j^*)$ and $k \in N_{ij^*}$. Decrease $\bar{c}_\lambda(i,k,\ell)$ until another path in G_i that contains $(k,\ell) \in A_i$ emerges as an alternative to the current shortest path. Suppose this occurs at increment $\delta > 0$, which, in turn, must satisfy

$$f_\lambda(i) = \tilde{f}_\lambda(i,k,\ell) - \delta\, a(i,k,\ell).$$

From Lemma 1, (6) follows. ■

Definition 3 *Let $f_\lambda(i \mid \sim E)$ be the length of a shortest path in G_i constrained to exclude arcs in set $E \in A_i$.*

Lemma 3 *Consider a Lagrangian solution x_λ for a given λ. If constraint j^* is violated because consumption exceeds what is available, i.e., LHS(j^*)> $b(j^*)$, then the step size in step 3 of the MAM can be computed*

$$\Delta = \min_{\substack{i \in I_\lambda^1(j^*) \\ \{(k,\ell) \mid x_\lambda(i,k,\ell)=1\}}} \left\{ \min_{(k',\ell') \in S_{ij^*}} \frac{\theta_\lambda(i,k',\ell')}{a(i,k,\ell) - a(i,k',\ell')}, \frac{f_\lambda(i \mid \sim \{(k,\ell)\}_{k \in N_{ij^*}}) - f_\lambda(i)}{a(i,k,\ell)} \right\} \tag{7}$$

where

$$S_{ij^*} = \{(k',\ell') \mid k' \in N_{ij^*},\ a(i,k',\ell') < a(i,k,\ell)\}.$$

Proof: Consider $x_\lambda(i,k,\ell) = 1$, $i \in I_\lambda^1(j^*)$ and $k \in N_{ij^*}$. Increase $\bar{c}_\lambda(i,k,\ell)$ until another path in G_i, not containing $(k,\ell) \in A_i$, emerges as an alternative to the current shortest path. Suppose this occurs at increment δ, which can be either zero or positive. Since the step 3 of the MAM requires a positive step size, assume $\delta > 0$. There are two possibilities for the new path; it will either 1) pass through one of the nodes in N_{ij^*}, or 2) skip all the nodes in N_{ij^*}.

In the first case, say the new path includes arc $(k', \ell') \in N_{ij^*}$ for which $x_\lambda(i, k', \ell') = 0$. Then, δ must satisfy

$$f_\lambda(i) + \delta\, a(i, k, \ell) = \tilde{f}_\lambda(i, k', \ell') + \delta\, a(i, k', \ell').$$

From Lemma 1

$$\theta_\lambda(i, k', \ell') = \delta[a(i, k, \ell) - a(i, k', \ell')].$$

A sufficient condition for $\delta > 0$ is $a(i, k', \ell') < a(i, k, \ell)$, that is, if the i-th problem is to use resource j^*, it should consume less than what it is currently consuming.

In the second case, similarly, δ must satisfy

$$f_\lambda(i) + \delta\, a(i, k, \ell) = f_\lambda(i \mid \sim \{(k, \ell)\}_{k \in N_{ij^*}}).$$

The value of $\lambda(j^*)$ is increased until either case occurs. ■

The MAM is a dual ascent procedure in that it monotonically increases the value of $L(\lambda)$. This property is established below.

Lemma 4 (Monotonicity) *Let j^* be a violated resource constraint in iteration t. Adjusting $\lambda^t(j^*)$ by $\Delta > 0$ yields $L(\lambda^{t+1}) > L(\lambda^t)$.*

Proof: If the violation is because LHS(j^*)> $b(j^*)$, increasing $\lambda(j^*)$ at the end of iteration t increases both components of $L(\lambda^t)$ in (4). First component increases because each $f_{\lambda^t}(i)$, $i \in I^1_{\lambda^t}(j^*)$, increases by $\Delta \times a(i, k, \ell)$, where $x_{\lambda^t}(i, k, \ell) = 1$, $k \in N_{ij^*}$. The second component increases by $\Delta \times b(j^*)$. Hence,

$$L(\lambda^{t+1}) = L(\lambda^t) + \Delta \sum_{i \in I^1_{\lambda^t}(j^*)} a(i, k, \ell)\, x_{\lambda^t}(i, k, \ell) - \Delta b(j^*) = L(\lambda^t) + \Delta[\text{LHS}(j^*) - b(j^*)].$$

Similarly, if the violation is because LHS(j^*)< $b(j^*)$, decreasing $\lambda(j^*)$ at the end of iteration t decreases both components of $L(\lambda^t)$ in (4). Hence,

$$L(\lambda^{t+1}) = L(\lambda^t) - \Delta \sum_{i \in I^1_{\lambda^t}(j^*)} a(i, k, \ell)\, x_{\lambda^t}(i, k, \ell) + \Delta b(j^*) = L(\lambda^t) + \Delta[b(j^*) - \text{LHS}(j^*)].$$

In either case, the Lagrangian function strictly improves. ■

In step 2 of the MAM, an ascent direction is chosen as a violated resource constraint. Let J_λ be the set of violated resource constraints for a given λ. Two popular rules for selecting $j^* \in J_\lambda$ are a) first ascent [8, 12, 13], which scans J_λ in a predetermined order and selects j^* as the first encountered, and b) steepest ascent, which scans J_λ and selects j^* as follows.

Corollary 1 (Steepest ascent) *Let $\Delta(j)$ be the step size as determined by either Lemma 2 or 3 for an ascent direction $j \in J_\lambda$. The steepest ascent direction j^* satisfies*

$$j^* = \operatorname*{argmax}_{j \in J_\lambda} \{ \Delta(j) \times |LHS(j) - b(j)| \}.$$

Proof: Follows from Lemma 4.

Determining the steepest ascent direction requires the computation of both terms in (7) for all $j \in J_\lambda$. The first term is obtained basically by a table search, but the second term calls for solving a shortest path problem with some nodes deleted. In problems where all arcs having tail nodes in set j have their head nodes in set $j+1$, the second term in (7) does not need to be computed, and hence, the steepest ascent improvement is possible with little extra computation. For example, shortest path formulations of certain knapsack problems have this structure. More discussion of this issue is given in Section 4.4.

Implementation of this general approach requires structural information from the underlying problem. The next section gives examples of this approach applied to several such structures: equipment replacement with capital budgets [19, 17], multiple item capacitated lot sizing, pavement maintenance scheduling [18], and the generalized assignment problem.

4 Applications

4.1 Equipment Replacement with Capital Budgets

Equipment replacement decisions typically involve a) which equipment to install, and b) how long to use it. The problem is to find a chain of serial replacements that minimizes the total discounted cost over n periods. Suppose there are m pieces of equipment with a limited capital budget encompassing all replacements in each period. If serial replacements are modeled by network constraints while periodic budgets are resources, our general resource constrained shortest path formulation directly applies to this problem as follows.

For a single asset, consider a directed graph where nodes correspond to the *beginning* of time periods and arcs correspond to the replacement decisions. Associated with each arc are three parameters: equipment type (technology), discounted cost (arc length), and installation cost (resource usage) of the replacement. An arc from node k to $\ell > k$ represents a particular technology to be acquired in period k and salvaged in period ℓ. Without resource constraints, a single equipment's replacement problem is solved by finding a shortest path from node 1 to node $n+1$, as illustrated in Figure 3 for a three-period problem. Since time periods define states, we have $N_{ij} = \{j\}$ for all i, j. See [17] for implementation details and computational experiments with large replacement problems. To briefly summarize, problems with up to 500 pieces of equipment and 20 periods were solved in, on the average, 178 seconds on an IBM RS/6000 model 320 to within 0.5% of optimality with an average budget violation of under 1%.

4.2 Multiple Item Capacitated Lot Sizing

Consider the problem of determining the quantity and timing of production for several products over a planning horizon so as to satisfy a known demand in each period and minimize

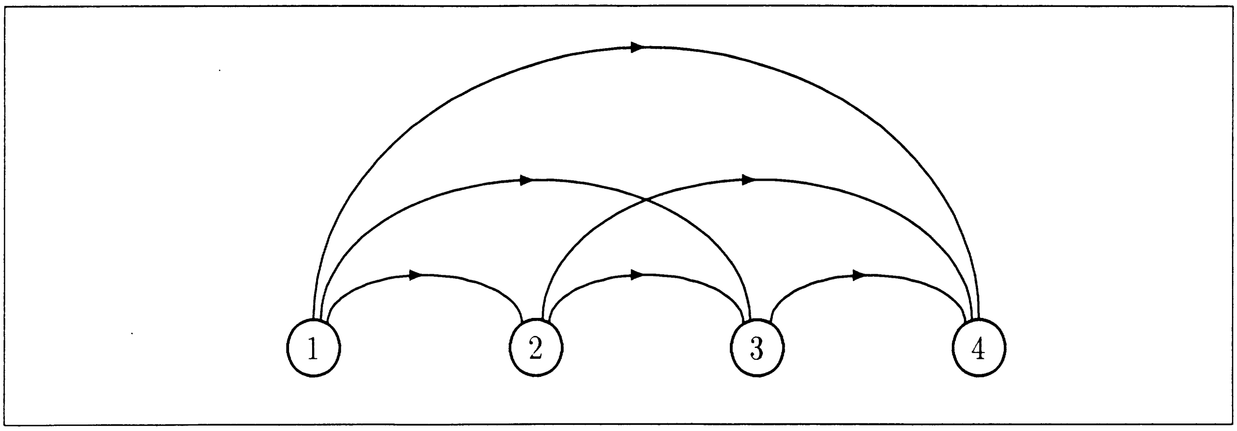

Figure 3: Shortest path representation of equipment replacement and lot sizing problems.

the sum of the production and inventory costs without incurring backlogs. Production costs consist of a fixed (setup) and a variable cost component, whereas the inventory costs are proportional to the quantity and time carried. A production capacity is imposed in each period. See [21] for a review of this literature.

If the resource constraints (production capacity) are relaxed, each item's lot sizing problem can be solved by Wagner-Whitin's shortest path algorithm [25]. The underlying network structure for this problem is identical to that of the equipment replacement problem in that the nodes correspond to the beginning of time periods and arcs correspond to the lot sizing decisions (see Figure 3). An arc (k, ℓ) represents a decision to supply the demand requirements for period $k, k+1, \ldots, \ell-1$ by production in period k. Two parameters are specified for each arc: cost of production and inventory holding (arc length) and resource usage of the production. Hence, our general MAM can be directly applied. A major drawback with this approach is that it restricts itself to schedules satisfying the Wagner-Whitin property, which states that the demand in a period is satisfied either entirely from production in the period or from production in a prior period. The optimal solution may not satisfy this property.

4.3 Pavement Maintenance Scheduling

The problem of scheduling road pavement maintenance is of serious concern in practice because of the rate at which pavement deteriorates to unacceptable levels. The pavement condition can be improved by selecting a maintenance alternative ranging from simple routine maintenance such as pothole patching and chip sealing to major maintenance practices such as reconstruction and rehabilitation [3, 10, 24].

The selection of a maintenance alternative is not a straightforward task because of the differences in cost and effectiveness. The least expensive alternative may not be the best because it allows the pavement to deteriorate faster in subsequent periods. Hence, while the cost of a maintenance alternative depends mainly on the pavement's condition, the overall cost of maintaining a street pavement over time is a function of both timing and type of maintenance alternatives. The problem is further complicated by the periodic budget limitations. Most government agencies work to an annual budget process which imposes budgets on pavement maintenance programs.

Consider a street segment with a current pavement condition. The decision maker may either let it deteriorate further by applying only routine maintenance or choose to install one of the major maintenance alternatives to improve the pavement condition. We assume that both deterioration and improvement rates are deterministic and known with certainty once the pavement condition and the choice of maintenance alternative are specified.

A street segment's problem has a network structure similar to the ones shown in Figure 1. Nodes correspond to pairs (time period, pavement condition) and arcs correspond to maintenance decisions. A maintenance decision consists of the type of maintenance alternative and its life. Associated with each arc are two parameters: length which is the discounted cost of the maintenance decision over its life, and resource usage which is the required investment. An arc from node (t_1, c_1) to (t_2, c_2) is selected only if its maintenance alternative is installed in period t_1 when the pavement condition is c_1, leading to condition c_2 in period t_2 and performing only routine maintenance between t_1 and t_2. A single segment's problem is solved by finding a shortest path from a node in the current period to a node in the horizon period. See [18] for details and computational experiments with large maintenance problems. To briefly summarize, problems with up to 200 street segments and 20 periods were solved in, on the average, 244 seconds on an IBM RS/6000 model 320 to within 0.1% of optimality with an average budget violation of under 0.1%.

4.4 Generalized Assignment Problem

The generalized assignment problem (GAP) seeks to optimally assign n tasks to m agents such that each task is assigned to exactly one agent but each agent is constrained only by the amount of a resource. Let v_{ij} be the cost of assigning task j to agent i, w_{ij} be the resource required by agent i to perform task j, and b_i be the resource available to agent i. Suppose a 0-1 variable, y_{ij}, is set to one if task j is assigned to agent i, zero otherwise. Then, an integer programming formulation for the GAP is:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} v_{ij}\, y_{ij} && && (8)\\
\text{subject to} \quad & \sum_{j=1}^{n} w_{ij}\, y_{ij} \le b_i && i = 1,\ldots,m && (9)\\
& \sum_{i=1}^{m} y_{ij} = 1 && j = 1,\ldots,n && (10)\\
& y_{ij} \in \{0,1\} && \text{all } i,j. && (11)
\end{aligned}$$

GAP has many applications and many algorithms were developed for its solution, see [4] for a recent survey. A common solution technique has been branch-and-bound with lower bounds obtained by various relaxations. A popular MAM was developed by Fisher, Jaikumar. and Van Wassenhove [8] for the Lagrangian dual obtained by relaxing the multiple-choice constraints (10). Let $\lambda(j)$ be the multiplier associated with multiple-choice constraint j. Then

$$L(\lambda) = \min_y \left\{ \sum_i \sum_j \bar{v}_{ij} y_{ij} \mid y \text{ satisfies (9) and (11)} \right\} - \sum_j \lambda(j). \tag{12}$$

where $\bar{v}_{ij} = v_{ij} + \lambda(j)$. Later, Guignard and Rosenwein [12, 13] suggested several enhancements over this MAM but the basic structure remained the same. We will refer to the enhanced algorithm as the *traditional MAM* in subsequent discussions.

For a given λ, Lagrangian solution x_λ is obtained by solving m shortest path problems. This is possible because each capacity constraint in (9) has a knapsack structure which lends itself to a shortest path formulation.

A knapsack problem is to select from various possible items that fit in a knapsack while minimizing a cost measure [23]. Suppressing the i-index in the variables for clarity, let v_j and w_j be the value and weight of item j, respectively. The knapsack can carry up to b units of weight. Set $y_j = 1$ if item j is put in the knapsack, otherwise $y_j = 0$. Also let $s_j = 1 - y_j$, with zero value and weight. Then we have

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} -v_j y_j \\ \text{subject to} \quad & \\ & \sum_{j=1}^{n} w_j y_j \leq b \\ & y_j + s_j = 1 \qquad j = 1, \ldots, n \\ & y_j, s_j \in \{0, 1\} \quad \forall j \end{aligned}$$

Knapsack problems can be solved as shortest path problems [6, 19, 16]. Consider a directed graph in which nodes are ordered pairs $[j, \beta]$, where j enumerates items and β enumerates the knapsack capacity ($\beta = 0, \ldots, b$). Arcs represent variable selection decisions such that an arc selecting item j must be from node $[j-1, \beta - w_j]$ to node $[j, \beta]$. Arc lengths are assigned the objective values of corresponding variables. Add a dummy source node and, using dummy arcs, connect it to the tail nodes of the arcs for the first item. Similarly, add a

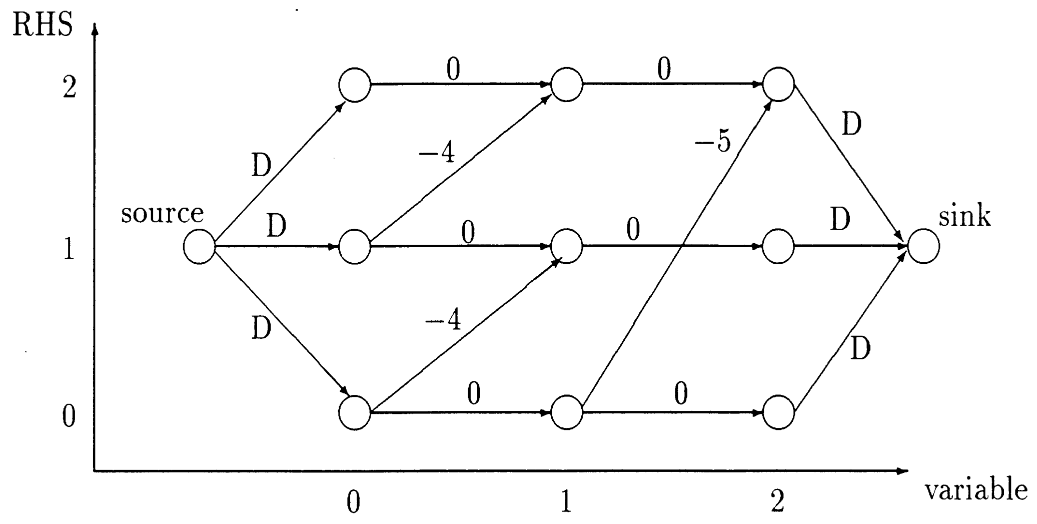

Figure 4: Shortest path representation of a 0-1 knapsack problem.

dummy sink node and, using dummy arcs, connect it to the head nodes of the arcs for the last item, n. All dummy arcs have zero lengths. Then, the knapsack problem is solved by finding a shortest path from the source node to the sink node. To illustrate, suppose we have two items which are valued 4 and 5 units and weigh 1 and 2 units, respectively. The knapsack can carry 2 units of weight. A shortest path graph for this problem is shown in Figure 4. Note that the diagonal arcs correspond to selecting item $j > 0$ (hence setting $y_j = 1$), whereas horizontal arcs correspond to not selecting item $j > 0$ (hence setting $s_j = 1$).

The traditional MAM uses the following approach for solving (12). For x_λ, let $p_j = \sum_i y_{ij}$ for all j and J_λ be the set of violated multiple choice constraints. For $j \in J_\lambda$, the method finds δ_{ij}, the least decrease (increase) in $\lambda(j)$ required for setting $y_{ij} = 1$ ($y_{ij} = 0$) in an optimum knapsack solution of row i if $p_j = 0$ ($p_j > 1$). A candidate step size is given by either $\Delta(j) = \min2_i\{\delta_{ij}\}$, where the min2 operator selects the second smallest element of its set, as suggested in [8] or $\Delta(j) = \min_i\{\delta_{ij}\}$ as suggested in [12, 13]. Both min and min2 operators are over

$$\begin{cases} \text{all } i, & \text{if } p_j = 0 \\ \{i \mid y_{ij} = 1\}, & \text{if } p_j > 1 \end{cases}$$

If $\Delta(j) > 0$, it is accepted as the step size, Δ. Otherwise, another $j \in J_\lambda$ is tried. If no strictly positive step size is found, the algorithm stops.

Lemma 5 *Let j be a violated multiple-choice constraint for x_λ.*

(a) When $\Delta(j) = \mathrm{min2}_i\{\delta_{ij}\}$, adjusting $\lambda(j)$ by $\Delta(j)$ improves the lower bound by

$$\begin{cases} \delta_{i^*j}, & \textit{if } p_j = 0 \\ (p_j - 2)\Delta(j) + \delta_{i^*j}, & \textit{if } p_j > 1 \end{cases} \tag{13}$$

(b) When $\Delta(j) = \min_i\{\delta_{ij}\}$, adjusting $\lambda(j)$ by $\Delta(j)$ improves the lower bound by

$$\begin{cases} \Delta(j), & \textit{if } p_j = 0 \\ (p_j - 1)\Delta(j), & \textit{if } p_j > 1 \end{cases} \tag{14}$$

Proof: See [16] for a derivation of (13). The result (14) follows directly from Lemma 4. ■

Note that when min2 operator is used in determining the step size, the MAM is no longer monotonic ascent.

The procedure suggested in [8, 12, 13] to determine δ_{ij} quantities is as follows. Let $z_i = \sum_j \bar{v}_{ij} y_{ij}$. Suppose we removed variable y_{ij} from knapsack i. We will call the remaining problem with $(n-1)$ variables a *restricted knapsack*. Let $z_i^R(\beta)$ be the optimum objective value of the restricted knapsack when its right hand side capacity is β. Now, if $p_j = 0$ and $y_{ij} = 0$, the least decrement in $\bar{v}_{ij}$ that would cause $y_{ij} = 1$ in an alternative optimum solution is given by

$$\delta_{ij} = \bar{v}_{ij} + z_i^R(b_i - w_{ij}) - z_i.$$

Similarly, if $p_j > 1$ and $y_{ij} = 1$, the least increment in $\bar{v}_{ij}$ that would cause $y_{ij} = 0$ in an optimum solution of knapsack i is given by

$$\delta_{ij} = z_i^R(b_i) - z_i.$$

Table 1: Summary table

	Traditional	Karabakal-Bean
Knapsack problems solved at initialization	m	$2m$
Knapsack problems solved in each iteration	$O(mn)$	$O(m)$
Steepest ascent?	No	Yes

This procedure implies that the number of shortest path problems solved to determine Δ range from m to $m \times n$, hence it is $O(mn)$. The procedure implied by Corollary 2 below solves $2m$ problems, hence it is $O(m)$.

Corollary 2 *Suppose multiple-choice constraint j is violated. Define $\theta[i,(j-1,\beta_1),(j,\beta_2)]$ as the slack of an arc from $(j-1,\beta_1)$ to (j,β_2), where $0 \leq \beta_1$, $\beta_2 \leq b_i$, in the directed graph of knapsack i. Then,*

$$\delta_{ij} = \begin{cases} \min\limits_{\beta=w_{ij},\ldots,b_i} \{\, \theta[i,(j-1,\beta-w_{ij}),(j,\beta)] \,\}, & \textit{if } p_j = 0 \\ \min\limits_{\beta=0,\ldots,b_i} \{\, \theta[i,(j-1,\beta),(j,\beta)] \,\}, & \textit{if } p_j > 1 \end{cases}$$

Proof: Follows from Lemma 3. ∎

The improvement over the traditional method is due to the "slack of an arc" (Definition 2) which avoids all the restricted knapsack solutions. In addition, with δ_{ij} quantities readily available as a table, and the bound improvement results of Lemma 5, it is easy to select a direction/step length pair to guarantee a steepest ascent improvement of $L(\lambda)$. Table 1 shows a comparison of the complexity of the two methods.

5 Conclusions

We describe an effective heuristic for solving multiple shortest path problems under coupling resource constraints. The heuristic is a MAM that solves a Lagrangian dual formulation. Starting with a solution obtained by relaxing the resource constraints, the MAM reduces the resource violations systematically until a feasible or acceptable solution is obtained. This MAM calculates the step size along a given ascent direction efficiently due to the "slack of an arc" concept. With this feature, developing the steepest ascent MAMs is computationally feasible in certain applications.

Implementation of this procedure is discussed for equipment replacement with capital budgets, multi-item capacitated lot sizing, and pavement maintenance scheduling. The steepest ascent feature is illustrated using the generalized assignment problem.

References

[1] Aneja, Y. P., V. Aggarwal, and P. K. Nair, "Shortest Chain Subject to Side Constraints," *Networks*, **13** (1983), 295–302.

[2] Beasley, J. E. and N. Christofides, "An Algorithm for the Resource Constrained Shortest Path Problem," *Networks*, **19** (1989), 379–394.

[3] Carnahan, J. V., "Analytical Framework for Optimizing Pavement Maintenance," *Journal of Transportation Engineering*, **114** (1988), 307–322.

[4] Cattrysse, D. G. and L. N. Van Wassenhove, "A Survey of Algorithms for the Generalized Assignment Problem," *European Journal of Operations Research*, **60** (1992), 260–272.

[5] Chan, T. J. and C. A. Yano C, "A Multiplier Adjustment Approach for the Set Partitioning Problem," *Operations Research*, **40** (1992), S40–S47.

[6] Denardo, E. V., *Dynamic Programming: Models and Applications*, Prentice Hall Inc. (1982).

[7] Fisher, M. L., "The Lagrangian Relaxation Method for Solving Integer Programming Problems," *Management Science*, **27** (1981), 1–18.

[8] Fisher, M. L., R. Jaikumar, and L. N. Van Wassenhove, "A Multiplier Adjustment Method for the Generalized Assignment Problem," *Management Science*, **32** (1986), 1095–1103.

[9] Geoffrion, A. M., "Lagrangian Relaxation for Integer Programming," *Mathematical Programming Study*, **2**, (1974) 82–114.

[10] Golabi, K., R. B. Kulkarni, and G. B. Way, "A Statewide Pavement Management System," *Interfaces*, **12** (1982), 5–21.

[11] Gopalan, R., R. Batta, and M. H. Karwan, "The Equity Constrained Shortest Path Problem," *Computers and Operations Research*, **17** (1990), 297–307.

[12] Guignard, M. and M. B. Rosenwein, "An Improved Dual Based Algorithm for the Generalized Assignment Problem," *Operations Research*, **37** (1989), 658–663.

[13] Guignard, M. and M. B. Rosenwein, "An Application-Oriented Guide for Designing Lagrangean Dual Ascent Algorithms," *European Journal of Operations Research*, **43** (1989), 197–205.

[14] Handler, G. Y. and I. Zang, "A Dual Algorithm for the Constrained Shortest Path Problem," *Networks*, **10** (1980), 293–310.

[15] Kaplan, S., "Solution of the Lorie-Savage and Similar Integer Programming Problems by the Generalized Lagrange Multiplier Method," *Operations Research*, **14** (1966), 1130–1136.

[16] Karabakal, N., "A Survey of Multiplier Adjustment Methods for the Generalized Assignment Problem," Technical Report, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, 1995.

[17] Karabakal, N., J. C. Bean and J. R. Lohmann, "Solving Large Replacement Problems With Budget Constraints," Technical Report, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, 1993.

[18] Karabakal, N., J. C. Bean and J. R. Lohmann, "Scheduling Pavement Maintenance with Deterministic Deterioration and Budget Constraints," Technical Report, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, 1994.

[19] Karabakal, N., J. R. Lohmann, and J. C. Bean, "Parallel Replacement Under Capital Rationing Constraints," *Management Science*, **40** (1994), 305–319.

[20] Lee, I.-S., "On Sensitivity Analysis for Shortest-Path Dynamic Programming Models," Working Paper, Graduate School of Management, University of California, Los Angeles, 1986.

[21] Maes, J. and L. Van Wassenhove, "Multi-Item Single-Level Capacitated Dynamic Lot-Sizing Heuristics: A General Review," *Journal of Operational Research Society*, **39** (1988), 991–1104.

[22] Neebe, A. W, "An Improved Multiplier Adjustment Procedure for the Segregated Storage Problem," *Journal of Operational Research Society*, **38** (1987), 815–825.

[23] Martello, S. and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*, John Wiley and Sons, 1990.

[24] Ritchie, S., C. Yeh, J. Mahoney, and N. Jackson, "Surface Condition Expert System for Pavement Rehabilitation Planning," *Journal of Transportation Engineering*, **113** (1987), 155–167.

[25] Wagner, H. M. and T. Whitin, "Dynamic Version of the Economic Lot Sizing Model," *Management Science*, **5** (1958), 89–96.