An approach for solving a class of transportation scheduling problems

B. GAVISH

Graduate School of Management, University of Rochester, Rochester, NY 14627, U.S.A.

E. SHLIFER

Technion, Israel Institute of Technology, Haifa, Israel

Received February 1978
Revised May 1978

An algorithm is developed for solving a class of transportation scheduling problems. It applies for a variety of problems such as: the Combining Truck Trip problem, the Delivery problem, the School Bus problem, the Assignment of Buses to Schedules, and the Travelling Salesman problem. The objective functions of the above problems differ from each other. Yet, by using the "savings method" proposed by Clarke and Wright, and extended by Gaskell, we are able to define each one of the above problems as a series of assignment problems. The cost matrix entries of each one of the assignment problems are a function of the constraints of the particular routing or scheduling problem. The solution to the assignment problem determines an upper bound of the optimal solution to the original problem. By combining the above procedure with a Branch and Bound procedure, it is possible to obtain the optimal solution in a finite number of steps. In some cases the Branch and Bound process can be eliminated due to the nature of the problem and in those cases the algorithm is efficient.

Introduction

An algorithm is developed for solving a class of transportation scheduling problems. It applies for a variety of problems such as: the Combining Truck Trip problem [14], the Delivery problem [7], the School Bus problem [4], the Assignment of Buses to Schedules [15], and the Travelling Salesman problem [2]. The objective functions of the above problems differ from each other. Yet, by using the "savings method" proposed by Clarke and Wright [7], and extended by Gaskell [12], we are able to define each one of the above problems as a series of assignment problems. The cost matrix entries of each one of the assignment problems are a function of the constraints of the particular routing or scheduling problem. The solution to the assignment problem determines an upper bound of the optimal solution to the original problem. By combining the above procedure with a Branch and Bound procedure, it is possible to obtain the optimal solution in a finite number of steps. Nevertheless, computational time might be excessive. In some cases, the Branch and Bound process can be eliminated due to the nature of the problem [14,15] and in those cases the algorithm was found to be very efficient.

In Section 1 we formulate a class of transportation scheduling problems and present an algorithm for solving them. The algorithm is outlined in Section 2 and the branching and separation rules are specified. In Section 3 we formulate the Travelling Salesman problem [8] as a savings problem and formulate and test the School Bus problem [4], and the Delivery problem [9]. By using the special structure of the Combining Truck Trip problem [14], and the Bus Scheduling problem [15], the algorithm was successfully implemented and proved to be efficient. However, when tested on the School Bus problem, it yielded results faster than known methods though still not satisfactory.

Description of the problem

A class of transportation scheduling problems as defined in this paper is characterized by: (a) vehicles that are located in a central depot have to be assigned to routes; (b) each vehicle starts from the depot, drives through a subset of collection points and returns to its origin; (c) each pickup point must be visited by one and only one vehicle. Often such schedules are designed in such a way that will minimize the cost function determined by the number of utilized vehicles and the cost of driving, subject to operational constraints. The constraints are dictated by the nature of the problem such as: maximum load per vehicle; maximal driving time in each route; minimal drivers rest time; maximal driving time from each collection point to the final destination.

European Journal of Operational Research 3 (1978) 122–134.

Most of the above constraints can be described by sets of zero-one variables and constraints. Unfortunately, the number of zero-one variables might increase to a stage which prohibits the solution of the problem by existing computers. Moreover, some constraints might be verbally easy to communicate and yet require complex mathematical formulation such as the last restriction above.

Bushnell and Venkatramn [4] and Svestka and Huckfeldt [27] have developed methods to solve special cases of the Transportation Scheduling problem for which the number of vehicles was given *a priori*. This special case was solved in [4] and [27] by a series of assignment problems which assure that each stop is visited by exactly one vehicle, and then followed by a Branch and Bound procedure. The procedure outlined in [4] and [27] was repeated for different fleet sizes until a global optimum was obtained. The method developed in this paper simultaneously determines the optimal number of vehicles while minimizing driving costs. However, the proposed method is not efficient for all kinds of problems.

The following procedure is proposed to solve the class of transportation scheduling problems. First, we define an equivalent problem, Problem II, whose objective function is to maximize the savings incurred by combining the collection points into partial routes.
Problem II is solved by applying a branch and bound procedure in which a sequence of assignment problems are solved. The entries of the cost matrix, for each one of the assignment problems, change according to the violations of the operational constraints evolving from combining pairs of collection points or partial routes into routes. Some problems lend themselves to the elimination of the Branch and Bound procedure. Those are cases in which a complete ordering of trips is a priori given (by time of departure for example). In those cases the algorithm is improved and performs efficiently [14, 15].

The above procedure was successfully implemented for Combining Truck Trips [14], and Bus Scheduling problems [15] and has been applied for the School Bus problem, at two orders of magnitude faster than the results obtained by Davis [9] and Bushnell et al. [4]. However, this procedure is not efficient and no optimal solution was obtained for a 21 pick-up point problem, in 900 seconds of CPU time on an IBM 370/165 (for 13 pick-up points an optimal solution was obtained in 9 seconds).

1. Formulation of the transportation scheduling problem

Identical vehicles located at a central depot $i = 0$, have to be allocated to tours in which they visit zero or more of the n collection points ($i = 1, 2, ..., n$). All vehicles originate from the central depot $i = 0$ and terminate their tours at $i = 0$. Along the tour, each vehicle visits a sequence of collection points, and returns to the depot. One, and only one, vehicle stops at each and every collection point. The objective function is to determine the optimal number of vehicles needed, the set of collection points visited by each one of the vehicles and their sequence.
Let:

$$X_{ij} = \begin{cases} 1 & \text{if a vehicle that stops at } i \text{ will have its next stop at } j.\ i, j = 0, 1, 2, ..., n, \\ 0 & \text{otherwise.} \end{cases}$$

V – a fixed cost incurred by each vehicle that is part of the schedule.

C_{ij} – cost for driving from point i to point j. C_{ij} is a function of the distance between i and j and the driving time.

t_{ij} – driving time from point i to point j.

q_i – a quantity to be loaded (or unloaded) at i.

κ – set of constraints characterized by the nature of the problem, where $k = (1, 2, ..., K) \in \kappa$.
There exists a set of transportation scheduling problems that could be formulated as follows:
Find values of x_{ij} that minimize:

$$Z_1 = \sum_{i=0}^{n} \sum_{j=0}^{n} C_{ij} X_{ij} + \sum_{j=1}^{n} V \cdot X_{0j} \tag{1}$$

subject to:

$$\begin{cases} \sum_{i=0}^{n} X_{ij} = 1 & j = 1, 2, ..., n, \\ \sum_{j=0}^{n} X_{ij} = 1 & i = 1, 2, ..., n, \\ X_{ij} = 0 \text{ or } 1 & \forall i, j \end{cases} \tag{2}$$

$$H_k(X) \leqslant E_k \qquad \forall k \in \kappa. \tag{3}$$

The set of constraints (2) assures that each collection point will be visited, by exactly one vehicle. The set of constraints (3) assures that other operational constraints are not violated. Yet, their explicit formulation is not relevant to the outlined algorithm in

Section 2. The set of constraints κ is specific for each type of transportation scheduling problem. Those are the loop constraints in the Travelling Salesman problem, the loop and load constraints in the Delivery problem, and the loop, load and time constraints in the School Bus problem.

It was long observed that (1) and (2) constitute an assignment problem, and that this fact could be used to aid in solving the transportation scheduling problem. This characteristic was utilized in some of the algorithms that were applied to solve transportation scheduling problems.

The main disadvantages of some of the algorithms were: (a) the need to add artificial sources and destinations to the original formulation equaling the number of vehicles that are being tested; (b) the objective function does not simultaneously handle both the cost associated with the number of vehicles and the cost associated with the driving costs.

These difficulties can be overcome by defining the problem as a "saving problem". The proposed algorithm commences by assuming that one vehicle is allocated to each collection point ($x_{0j} = 1$, $\forall j$), and therefore, n vehicles are allocated. The objective function value for this schedule Z_0 is:

$$Z_0 = n \cdot V + \sum_{i=1}^{n} (C_{0i} + C_{i0}) \,. \tag{4}$$

Starting from this schedule, and combining collection points i and j, $i \neq j$, into a partial route $(0, i, j, 0)$, we obtain a savings S_{ij} due to the reduction of one vehicle and shortening the driving time and cost:

$$S_{ij} = \begin{cases} V + C_{i0} + C_{0j} - C_{ij} & i \neq j,\ i, j = 1, 2, \ldots, n, \\ 0 & i = j,\ i = 1, 2, \ldots, n. \end{cases}$$

We assume that the fixed cost V is large enough to ensure that,

$$V + C_{i0} + C_{0j} - C_{ij} > 0 \quad \text{for} \quad i \neq j,\ i,j = 1, 2, \ldots, n.$$

This is justified by the fact that by saving one vehicle, we save not only the initial investment in an additional vehicle, but also reduce the number of drivers, being proportional to the number of vehicles needed for the schedule.

We define points i and j to be connected if they belong to the same partial route and a vehicle drives directly from point i to point j. If we form a route $0, i_1, i_2, \ldots, i_l, 0$ by connecting collection points $i_1, i_2, \ldots, i_l$, then the savings $S_{i_1,i_2,\ldots,i_l}$ obtained by forming that route in the above order are:

$$S_{i_1,i_2,\ldots,i_l} = \sum_{k=1}^{l-1} S_{i_k i_{k+1}}$$

since:

$$\begin{aligned} S_{i_1,i_2,\ldots,i_l} &= V \cdot (l-1) + \sum_{k=1}^{l-1} C_{i_k 0} + \sum_{k=2}^{l} C_{0 i_k} - \sum_{k=1}^{l-1} C_{i_k i_{k+1}} \\ &= \sum_{k=1}^{l-1} S_{i_k i_{k+1}} \\ &= \sum_{k=1}^{l-1} (V + C_{i_k 0} + C_{0 i_{k+1}} - C_{i_k i_{k+1}}) \,. \end{aligned}$$

We define *Problem II* where the total savings are maximized by connecting the collection points into a set of routes and then show that the optimal solutions of the two problems are identical.

Problem II. Let:

$$Y_{ij} = \begin{cases} 1 & \text{if } i \text{ is directly connected to } j \\ & i, j = 1, 2, \ldots, n,\ i \neq j, \\ 0 & \text{otherwise.} \end{cases}$$

$$\tilde{S}_{ij} = \begin{cases} 0 & \text{for } i = 0 \text{ or } j = 0, \\ S_{ij} & \text{otherwise.} \end{cases}$$

Find values of Y_{ij} that maximize:

$$Z_{\mathrm{II}} = \sum_{i=0}^{n} \sum_{j=0}^{n} Y_{ij} \cdot \tilde{S}_{ij} \tag{5}$$

subject to:

$$\begin{cases} \sum_{i=0}^{n} Y_{ij} = 1 \,, & j = 1, 2, \ldots, n. \\ \sum_{j=0}^{n} Y_{ij} = 1, & i = 1, 2, \ldots, n. \\ Y_{ij} = 0 \text{ or } 1 & \forall i, j \end{cases} \tag{6}$$

$$H_k(Y) \leqslant E_k \qquad \forall k \in \kappa. \tag{7}$$

Problem I and Problem II are equivalent. The optimal solution Y^*_{ij} of Problem II yields the optimal solution X^*_{ij} of Problem I.

Proof. The constraint set of the two problems are equivalent and therefore it is enough to show that the objective function values are equivalent.

We shall prove that X^*_{ij} obtained by minimizing Z_{I} is identical to Y^*_{ij}, obtained by maximizing Z_{II}, or:

$$Z_0 - \min\{Z_{\mathrm{I}}\} = \max\{Z_{\mathrm{II}}\},$$

and Z_0 is defined in (4). (8)

the optimal solution to Problem II is:

$$Z^*_{\mathrm{II}} = \sum_{j=0}^{n} \sum_{i=0}^{n} Y^*_{ij} \tilde{S}_{ij}$$

$$= \sum_{j=1}^{n} \sum_{i=1}^{n} (V + C_{i0} + C_{0j} - C_{ij})\, Y^*_{ij}\,. \tag{9}$$

Let N^*_{II} be the number of vehicles determined by the optimal solution to Problem II. Since $Y^*_{00} = 0$, then:

$$N^*_{\mathrm{II}} = \sum_{i=0}^{n} Y^*_{i0} = \sum_{j=0}^{n} Y^*_{0j}\,. \tag{10}$$

We add the expression $\Sigma_{i=1}^{n} (S_{i0} Y^*_{i0} + S_{0i} Y^*_{0i}) - \Sigma_{i=1}^{n} (S_{i0} Y^*_{i0} + S_{0i} Y^*_{0i}) = 0$ to (9) and by rearrangement of terms and (6) we obtain:

$$Z^*_{\mathrm{II}} = (n + N^*_{\mathrm{II}})\, V + \sum_{j=1}^{n} C_{0j} + \sum_{i=1}^{n} C_{i0}$$

$$+ 2 \cdot N^*_{\mathrm{II}} \cdot C_{00} - \sum_{i=1}^{n} \sum_{j=0}^{n} C_{ij} Y^*_{ij}$$

$$- N^*_{\mathrm{II}} \cdot V - N^*_{\mathrm{II}} C_{00} - N^*_{\mathrm{II}} V - N^*_{\mathrm{II}} C_{00}$$

$$= Z_0 - \{\sum_{j=0}^{n} \sum_{i=0}^{n} C_{ij} Y^*_{ij} + N^*_{\mathrm{II}} V\} \tag{11}$$

substituting $N^*_{\mathrm{II}} = \Sigma_{j=1}^{n} Y^*_{0j}$ in (11) yields:

$$Z^*_{\mathrm{II}} = Z_0 - \{\sum_{i=0}^{n} \sum_{j=0}^{n} C_{ij} Y^*_{ij} + \sum_{j=1}^{n} V \cdot Y^*_{0j}\}$$

$$= Z_0 - Z^*_{\mathrm{I}} \tag{12}$$

therefore $\max\{Z_{\mathrm{II}}\} = Z_0 - \min\{Z_{\mathrm{I}}\}$ and the two problems are equivalent.

Solving Problem II involves the same difficulties as Problem I. By relaxing constraint (7) we obtain the assignment Problem III which yields upper bounds to the solution of Problem II.

Problem III. Find variables Y_{ij}, $i, j = 1, 2, ..., n$ that maximize:

$$Z_{\mathrm{III}} = \sum_{j=1}^{n} \sum_{i=1}^{n} Y_{ij} \tilde{S}_{ij} \tag{13}$$

subject to:

$$\begin{cases} \sum_{i=1}^{n} Y_{ij} = 1\,, & j = 1, 2, ..., n, \\ \sum_{j=1}^{n} Y_{ij} = 1 & i = 1, 2, ..., n, \\ Y_{ij} \geqslant 0 & \forall i, j. \end{cases} \tag{14}$$

Let Z^*_{III} be the optimal value of the objective function of Problem III. Then, $Z^*_{\mathrm{II}} \leqslant Z^*_{\mathrm{III}}$, i.e., the value of the optimal solution to Problem III is an upper bound to the optimal solution to Problem II. This relationship makes it possible to formulate a Branch and Bound algorithm to solve Problem II. However, the algorithm will essentially be the same as for algorithms which were developed to solve Problem I, and were found to be inefficient.

In order to improve the efficiency of the algorithm, some of the operational constraints of (7) are implicitly incorporated into the subproblems which are generated during the solution process. When partial routes are formed at any stage of the solution process and they do violate any of the operational constraints, those partial routes are split and their costs are adjusted to prevent their reoccurrance in later stages. This is done using the following procedure:

Let L_k be a given route which consists of an ordered set of k collection points: $L_k = \{i_1, i_2, ..., i_g, ..., i_k\}$. A route could be described as a single collection point in the savings matrix, where i_k is an origin (row) and i_1 is a destination (column). Those nodes have special properties. For example, the vehicle is delayed in i_1 and i_k an amount of time equal to the driving time along the partial route L_k. The cargo weight that has to be collected in those nodes is the sum of cargos of the nodes belonging to partial route L_k.

If during the solution process the collection points $\{i_1, i_2, ..., i_k\}$ become a partial route, then the original assignment Problem III is changed into a new assignment problem in which all the data pertaining to the collection points $\{i_1, i_2, ..., i_k\}$ are deleted and a new point is added which represents the partial route. This latter point will appear twice, once as the origin of the partial route and once as its end.

Once k collection points were combined into a route $L_k = \{i_1, i_2, ..., i_k\}$, the following updating process takes place:

(i) Delete from the previous assignment problem (with dimension $p \times p$, $p > k$) all rows and columns which belong to $\{i_1, i_2, ..., i_k\}$.

(ii) Add to the objective function a constant which equals the savings that are obtained by connecting the collection points $\{i_1, i_2, ..., i_k\}$ into a partial route. Consequently, a new assignment problem is obtained with dimensions $(p - k) \times (p - k)$. The rows and columns of the modified cost matrix are renumbered from 1 to $h - 1$ where $h = p - k + 1$.

(iii) A new row h is added to the new assignment problem which indicates the beginning of the partial route i_1, and a new column h that indicates the end of the route i_k before returning to the depot $i = 0$.

(iv) Redefine the savings:

$$\tilde{S}_{ij} = \begin{cases} \tilde{S}_{ij} & \text{for } i, j = 1, 2, ..., h-1, \\ \tilde{S}_{i_k j} & \text{for } i = h, j = 1, 2, ..., h-1, \\ \tilde{S}_{i i_1} & \text{for } j = h, i = 1, 2, ..., h-1, \\ 0 & \text{for } i = j = h. \end{cases}$$

The cargo requirements:

$$q_i = \begin{cases} q_i & i = 1, 2, ..., h-1, \\ \sum_{r=1}^{k} q_{i_r} & i = h. \end{cases}$$

Driving times between collection points:

$$t_{ij} = \begin{cases} t_{ij} & i, j = 1, 2, ..., h-1, \\ t_{i_k j} & i = h, j = 1, 2, ..., h-1, \\ t_{i i_1} & j = h, i = 1, 2, ..., h-1, \\ 0 & i = j = h. \end{cases}$$

Driving times between the center and the collection points:

$$t_{i0} = \begin{cases} t_{i0} & i = 1, 2, ..., h-1, \\ t_{i_k 0} & i = h. \end{cases}$$

$$t_{0j} = \begin{cases} t_{0j} & i = 1, 2, ..., h-1, \\ t_{0 i_1} & j = h. \end{cases}$$

Let τ_i be the time spent by a vehicle at point i. If i represents a partial route, then τ_i will be the time that the vehicle spends in the route from i_1 to i_k.

$$\tau_i = \begin{cases} \tau_i & i = 1, 2, ..., h-1, \\ \sum_{r=1}^{k} \tau_{i_r} + \sum_{r=1}^{k-1} t_{i_r i_{r+1}} & i = h. \end{cases} \tag{15}$$

At the beginning of the process $\tau_i = 0\ \forall i$.

(v) The saving matrix is updated by examining all possible connections between the origin points $i_1, i_2, ..., i_h$ to all destination points $j_1, j_2, ..., j_h$. The connection of the origin i_r to the destination j_s forms a new partial route that is the ordered union of all the points which were contained in i_r and i_s. Such a connection is permitted only when the new partial route fulfills all the operational constraints and is denoted by $i_r \to j_s$ if the connection is permitted. $i_r \not\to j_s$ denotes connections that are not permitted.

The savings matrix is updated to

$$\tilde{S}_{ij} = \begin{cases} S_{ij} & \forall i \to j \\ 0 & \forall i \not\to j \end{cases} \tag{16}$$

which means that the savings are equal to zero when the connection is not permitted.

The optimal solution to the updated assignment problem is an upper bound to the optimal solution for the updated sub-problem and this enables the development of a branch and bound algorithm.

2. Outline of the algorithm

The proposed algorithm for solving transportation scheduling problems is based on a branch and bound procedure (see Lawler and Wood [18] or Mitten [21]). In each step of the proposed algorithm, the set of possible solutions of Problem II is examined and is separated into disjoint subsets. Each subset is characterized by a different set of trip constraints. Each subset consists of two types of trips: determined and nondetermined. The determined trips can be further split into connected trips and nonfeasible (are not permitted). Each subset can be presented as a node (node i) in a branching tree. The arcs along the path from the root of the tree to node i, represent operational constraints as well as those generated by the branch and bound procedure. For each node i, a subproblem which includes the additional constraints is generated and solved. The solution yields an upper bound to the contrained Problem II. Whenever the upper bound of a node is less or equal to the value of the current best feasible solution to Problem II, the node is fathomed.

A branching tree might contain two types of nodes. Those which have successors (i.e., their solution subset has been separated) and those without successors. The nodes without successors are divided into two classes; fathomed and pending nodes (those who were not fathomed). At the initiation of the solution procedure, the branching tree consists of one node (the root, $i = 0$) whose subproblem is Problem III. During the solution process, new nodes are added to the branching tree. The process terminates when a feasible solution is found and all nodes are fathomed.

We will outline the algorithm and then present a set of rules for separation, branching and fathoming.

Let: W_i – be the bound on the savings of the optimal solution through node i; F – the value of the current best feasible solution; Z^k_{III} – the value of the objective function of the k^{th} subproblem (node-k) in the branching tree.

The algorithm consists of the following steps:

1. *Initialization* – Start at node $i = 0$ (root). Set $F = -\infty$. Modify Problem II into Problem III, and solve Problem III. If a feasible solution is obtained to Problem II, go to step 6. Otherwise: If $Z^0_{\mathrm{III}} = 0$, fathom the node; if $Z^0_{\mathrm{III}} \neq 0$ set $W_0 = Z^0_{\mathrm{III}}$ and go to step 2.

2. *Branching* – Select next pending node-j for branching (the one with the largest upper bound value); generate the appropriate subproblem and go to step 3. If all nodes were fathomed go to step 6.

3. *Separation* – Choose a set of variables as separation variables, and generate the successive nodes to node j. The separation rules follow the outline.

4. *Calculating Bounds* – for each direct successive node k to node j, generate a modified subproblem k and solve. If the solution to subproblem k is:
   (a) a feasible solution to Problem II, fathom node k and if $W_k > F$, set $F = W_k$ and go to step 5.
   (b) a nonfeasible solution to Problem II and
   (1) $Z^k_{\mathrm{III}} = 0$, fathom node k and go to step 2, or
   (2) $W_k \leq F$ then fathom node k, otherwise add node k to the branching tree as a direct successor to node j and go to step 2.

5. *Fathoming* – If $W_i \leq F$ fathom node i and go to step 2.

6. If $F = -\infty$ then no feasible solution exists for the problem. The solution corresponding to $F \neq -\infty$ is optimal.

Identification of feasible solutions

In step 4a of the outline we identify the feasible solution to the transportation scheduling problem. The identification of feasible solutions is achieved by analyzing the loops in the solution to the assignment problem. The loops are the positive variables that are presented as arcs in the assignment graph. We divide the indices of the positive basic variables in the optimal solution to the assignment subproblem into two sets Ω and $\bar{\Omega}$ where

$$\Omega = \{(i,j) \mid Y^*_{ij} = 1, \tilde{S}_{ij} > 0\} \tag{17}$$

$$\bar{\Omega} = \{(i,j) \mid Y^*_{ij} = 1, \tilde{S}_{ij} = 0\}.$$

After obtaining the optimal solution to the assignment subproblem III, a distinction between two types of loops is made.

(a) Loops which are formed of partial routes. If in loop $L = \{i_1, i_2, ..., i_p, i_{p+1}, i_1\}$, there exist at least one index r, $r = 1, 2, ..., p$ such that $(i_r, i_{r+1}) \in \bar{\Omega}$, then L is composed of partial routes. The collection points i_r and i_{r+1} can be disconnected without affecting the value of the objective function and therefore point i_{r+1} can be defined as a beginning of a partial route. If k arcs belong to $\bar{\Omega}$, then loop L is divided into k partial routes (some of them possibly containing a single collection point).

(b) Loop $L = \{i_1, i_2, ..., i_p, i_{p+1} = i_1\}$ is a pure loop if $(i_r, i_{r+1}) \in \Omega$, $\forall r = 1, 2, ..., p$.

A feasible solution to a transportation scheduling problem is a solution to an assignment problem that satisfies the following conditions:

1. It contains loops that consist of partial routes only.
2. Each one of the partial routes satisfies the operational constraints which are given for that particular scheduling problem.

If the solution to the subproblem is nonfeasible, then a new node is selected for branching among the pending nodes in the branch and bound tree. This is a node with a nonfeasible solution to Problem II.

Separation rules

Three basic methods for separation were previously proposed in the literature.

Method A – was proposed by Eastman [10] and implemented by Little et al. [19] for solving the travelling salesman problem. One of the connections (e.g., (i_1, i_2)) in the nonfeasible partial route is selected and two subproblems are generated. In one subproblem, $Y_{i_1i_2}$ is set to zero while in the second subproblem, $Y_{i_1i_2}$ is set to one.

Method B – was proposed by Shapiro [26]. A partial route or a loop that contains $k+1$ points $\{i_1, i_2, ..., i_{k+1}\}$ is selected and the set of solutions is separated into k subsets. In subset g, $y_{i_gi_{g+1}}$ is set to zero.

Method C – was proposed by Murty [22] and was found to be the most effective method in tests performed by Bellmore and Malone [3] and by Svestka and Huckfeldt [27]. As in method B, we select a partial route or loop that contains $k+1$ collection points and generate k subsets by setting

in subset 1; $Y_{i_1i_2} = 0$
in subset 2; $Y_{i_1i_2} = 1$, $Y_{i_2i_3} = 0$
in subset 3; $Y_{i_1i_2} = 1$, $Y_{i_2i_3} = 1$, $Y_{i_3i_4} = 0$
in subset k; $Y_{i_1i_2} = 1$, $Y_{i_2i_3} = 1$, $Y_{i_3i_4} = 1, ..., Y_{i_ki_{k+1}} = 0$.

Computational experiments that we have conducted have confirmed the experience of Bellmore and Malone [3] and Svestka and Huckfeldt [27]. That is, the number of branches which are generated by Murty's method are less than generated by other methods. Another advantage of Murty's method is that it enables us to use the optimal base for the p^{th} subproblem as an initial base for the $p+1$ subproblem. Therefore Murty's method was used during the computational tests.

Separation rule

Those were the variables contained in the first encountered pure loops in the subproblem. If the solution to the subproblem didn't contain pure loops then the variables forming the first encountered nonfeasible partial route were selected for separation.

In certain problems it is possible to use special properties of the problem given by the extra constraints in κ, in order to further restrict the solution space. In those cases it is possible to specialize the algorithm and reduce or to completely eliminate the branching during the branch and bound process. Examples of such specializations are the preordering of trips according to their arrival and departure times in the Combining Truck Trips [14] and the bus scheduling problem [15]. Other examples are the Clustered Travelling Salesman problem [5] and the Bottleneck Travelling Salesman problem [11] which have extra constraints that can be used in order to further prune the branch and bound tree.

3. Application of the algorithm to various transportation scheduling problems

In this section we describe several problems which can be solved using the algorithm proposed in Section 2. In Section 3a we formulate the Multi-Travelling Salesman problem and outline a procedure for solving it; however, the procedure was not computationally tested. In Section 3b, we formulate the School Bus problem and report results of computational tests. The tests yield results faster than in previously published studies [4,9]. Yet, due to the complexity of the problem, computation time grows excessively with the increase in the problem size. In Section 3c, we formulate and report computational results for the Bus Scheduling problem [15] which was successfully implemented. A similar approach to the one described in 3c was used for the Combining Truck Trips problem and is reported in [14].

3a. The multi-travelling salesman problem

The Travelling Salesman problem as formulated by Miller et al. [20] was extended in [13] to the Multi-Travelling Salesman problem. M salesman visit n cities $(1 \leqslant M \leqslant n-1)$. They originate from one city (city n), and after visiting the cities they return to the origin. No more than one salesman is allowed to visit a city and all cities must be visited. M tours are planned, one for each salesman, such that the total travelling costs will be minimized. The problem was formulated in [13] as:

Find variables X_{ij} and U_i $i, j = 1, 2, ..., r$ that minimize:

$$Z = \sum_{i=1}^{r} \sum_{j=1}^{r} \hat{C}_{ij} X_{ij}, \quad \text{where } r = M + n - 1$$

subject to

$$\left.\begin{aligned} &\sum_{i=1}^{r} X_{ij} = 1 && j = 1, 2, ..., r, \\ &\sum_{j=1}^{r} X_{ij} = 1 && i = 1, 2, ..., r, \\ &X_{ij} = 0 \text{ or } 1 && \forall i, j \end{aligned}\right\} \tag{18}$$

$$U_i - U_j + (n - M) X_{ij} \leqslant n - M - 1,$$

$$i \leqslant j, j \leqslant n-1, i \neq j \tag{19}$$

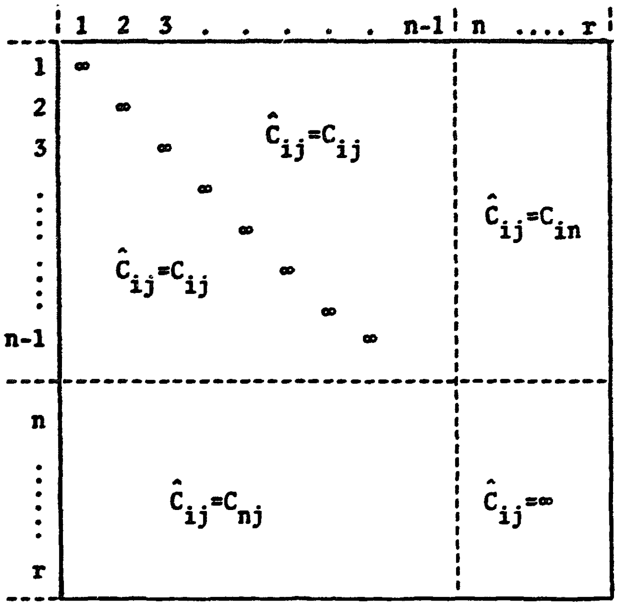


Fig. 1. The modified cost matrix structure.

where $\hat{C}_{ij}$ is defined as:

$$\hat{C}_{ij} = \begin{cases} \infty & \text{for } i=j \text{ or } i \geqslant n \text{ and } j \geqslant n. \\ C_{ij} & \text{for } i \neq j,\ i,j = 1,2,...,n. \\ C_{nj} & \text{for } i \neq j,\ j > n,\ j = 1,2,...,n-1. \\ C_{in} & \text{for } i \neq j,\ j > n,\ i = 1,2,...,n-1. \end{cases}$$

A schematic diagram of the new cost matrix is given in Fig. 1. By choosing an arbitrarily origin, e.g., city p, the problem is directly transformed into a savings problem where the savings are:

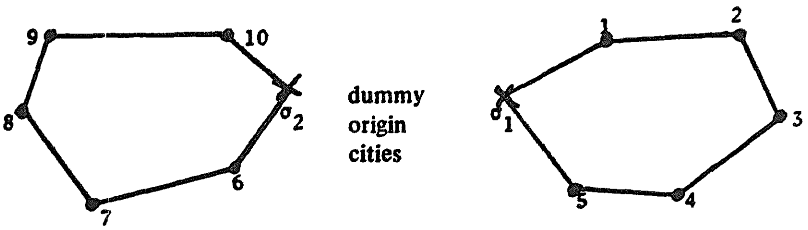
$$\tilde{S}_{ij} = \begin{cases} C_{ip} + C_{pj} - C_{ij} + \lambda > 0 & i,j = 1,2,...,r \\ & i \neq j,\ i \neq p \text{ and } j \neq p \\ 0 & \text{otherwise.} \end{cases}$$

λ is a positive constant to ensure that $\tilde{S}_{ij} \geqslant 0\ \forall i,j$. The addition of λ does not alter the optimal solution.

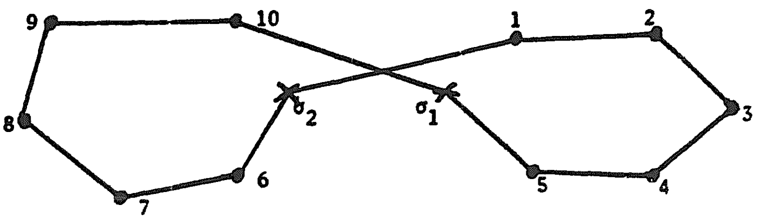
As shown in [27] the solution to a Single Travelling Salesman problem is also an optimal solution to the Multi-Travelling Salesman problem. Therefore, the same solution method can be applied to solve the Multi-Travelling Salesman problem. Note that not all feasible solutions to the Multi-Travelling Salesman problem are feasible solutions to the Single Travelling Salesman problem formed on the r cities (see Fig. 2 for an example). Therefore, a different criterion should be developed to determine when a nonfeasible solution to the Travelling Salesman problem is feasible for the Multi-Travelling Salesman problem. The application of this criterion might improve the solution procedure.

3b. The school bus problem

School buses are centrally located and have to collect waiting students at n pick-up points and to



(a) A feasible solution to the multi-travelling salesman problem which is not a feasible solution to the single travelling salesman problem.



(b) A feasible solution for the single and multi-travelling salesman problem.

Fig. 2. Examples of feasible and nonfeasible solutions to the single travelling salesman problem.

drive them to School. The number of students that wait in pick-up point i is $q_i, q_i > 0, i = 1,2, ..., n$. The capacity of each bus is limited to Q students, $q_i \leq Q$ $\forall i$. Let τ_{i0} be the time spent on the bus by a student that is picked up at i until he reaches school. Security and operational considerations limit τ_{i0} not to exceed τ $\forall i$.

The objective function to the School Bus problem is composed of two costs: (a) cost incurred by the number of buses used by the schedule; (b) driving costs (fuel, maintenance, drivers salary, etc.,). Costs (a) and (b) have to be minimized, subject to operational constraints.

No known method minimizes (a) and (b) simultaneously. It was proposed by Davis [9], Bushnell et al. [4], and Christofides and Eilon [6] to minimize the driving costs for a given number of buses M and then repeat the above procedure for different values of M. The solution that minimizes the costs incurred by the number of buses and driving costs, is the optimal one.

For a given number M of buses, the School Bus problem can be formulated as a zero one programming problem.

Let X_{ijk} $i, j = 0, 1, 2, ..., n$, $k = 1, 2, ..., M$ be variables that attain the value 1 if pick-up points i and j are visited by the k^{th} bus, and pick-up point j is visited directly after i. Otherwise, X_{ijk} is 0. Let U_{ik} $i = 0, 1, ..., n$, $k = 1, 2, ..., M$ be variables that may attain any value. The School Bus problem is formulated as follows:

Find variables X_{ijk} and U_{ik} that minimize

$$Z = \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{M} \hat{C}_{ij} X_{ijk}$$

where:

$$\hat{C}_{ij} = \begin{cases} C_{ij} & \forall i, j, i \neq j, \\ \infty & \forall i = j, \end{cases}$$

subject to:

$$\left\{ \begin{aligned} & \sum_{k=1}^{M} \sum_{j=0}^{n} X_{ijk} = 1 && i = 1, 2, ..., n, \\ & \sum_{k=1}^{M} \sum_{i=0}^{n} X_{ijk} = 1 && j = 1, 2, ..., n, \\ & \sum_{k=1}^{M} \sum_{j=0}^{n} X_{0jk} = \sum_{k=1}^{M} \sum_{i=0}^{n} X_{i0k} = M \end{aligned} \right. \tag{20}$$

$$\left\{ \begin{aligned} & \sum_{j=0}^{n} X_{ijk} = \sum_{j=0}^{n} X_{jik} && i = 1, 2, ..., n, \\ & && k = 1, 2, ..., M, \\ & U_{ik} + U_{jk} + (n - m + 1) X_{ijk} \leq (n - M) && \\ & && 1 \leq i, j \leq n, i \neq j, \\ & && k = 1, 2, ..., M. \end{aligned} \right. \tag{20a}$$

Constraints (20) and (20a) ensure the formation of exactly M tours, where each one passes through the school. In addition, the following constraints are added:

capacity constraints:

$$\sum_{i=0}^{n} \sum_{j=0}^{n} X_{ijk} q_i \leq Q \qquad k = 1, 2, ..., M,$$

time constraints:

$$\sum_{i=1}^{n} \sum_{j=0}^{n} X_{ijk} \cdot t_{ij} \leq \tau \qquad k = 1, 2, ..., M.$$

The three dimensional assignment problem given in (20) could be transformed into a regular assignment problem by duplicating $M - 1$ times the row and column corresponding to city 0, and obtaining an assignment problem with dimensions $(n + M)$ by $(n + M)$. This representation was used by Davis [9] and by Bushnell et al. [4] to develop a Branch and Bound algorithm to solve the problem. The largest problem solved by Davis [9] was 6 collection points and 2 buses. Bushnell et al. [4] solved a case of 10 collection points and 2 buses. Both methods did not obtain a solution of a 13 collection point problem in 15 minutes of CPU time on a CDC 6600 machine.

The School Bus problem can be solved by the proposed algorithm in Section 2. We set V to be the fixed value obtained by saving one bus from the schedule and C_{ij} the driving cost from point i to point j. Connection of two partial routes i and j into a new partial route is not allowed when one of the following relations holds:

(1) $\hat{q}_i + \hat{q}_j > Q$, i.e. the capacity constraint is violated.

(2) $\tau_i + t_{ij} + \tau_j + t_{j0} > \tau$, i.e. the time constraint is violated.

(3) the savings $\hat{S}_{ij} = 0$, i.e. the connection of the two partial routes was not allowed in previous steps.

The saving obtained by combining the partial routes

i and j into a new partial route is given by

$$S_{ij} = \begin{cases} V + C_{i0} + C_{0j} - C_{ij} & i, j = 1, 2, ..., k, i \to j \\ 0 & i \not\to j. \end{cases}$$

Based on the above rules, a program was written to check the efficiency of the algorithm for the School Bus problem. Murty's [22] branching rule was applied. The program was tested on problems presented in Bushnell et al. [4], i.e. of dimensions (6×6), (10×10) and (13×13). The problems were solved in 0.43, 4 and 9 seconds of CPU times on an IBM 370/165.

Encouraged by those results, an attempt was made to solve a 21 point delivery problem ($\tau = \infty$) that is presented in Gaskell [12] and solved there using a heuristic method. The problem was not solved in 15 minutes of CPU time. No solutions were obtained in 15 minutes when the program was run on problems of 22, 29 and 31 pick-up points that were presented in Gaskell [12].

It appears that the proposed algorithm, though superior to others, is not suitable for practical School Bus problems. The CPU time increases exponentially as the size of the problem increases. However, it can be used for examining the efficiency of different heuristics which were proposed in [1] and [23] to solve the School Bus problem or the Delivery Problem, by providing lower bounds on the value of the optimal solution.

3c. The bus scheduling problem

Bus scheduling is a problem encountered by large bus companies operating in metropolitan areas. It is concerned with the scheduling of buses to thousands of relatively short trips. The time tables for those trips reflect the varying demand for transportation and are planned in advance. It is our objective to schedule buses to trips in such a way that costs incurred by the fleet size and dead heading are minimized while satisfying operational constraints. Such problems were formulated by Kirkman [17], Orloff [24], Saha [25], Wren [28,29]. Kirkman [17] minimized the total dead heading time which in most cases will lead to reduction in fleet size. Saha [25] developed an efficient algorithm to minimize the fleet size. Wren developed the Vampire program which applies a two step heuristic algorithm that first determines the fleet size and then minimizes dead heading given that fleet size.

Gavish et al. [15], formulated the problem as a modified assignment problem. In [15] the costs incurred by the fleet size, dead heading and control are simultaneously minimized while meeting the time table requirements. The modified assignment formulation lends itself to; (a) changes in existing time tables and/or operational control trade-offs and (b) handling large scale and realistic problems. We will first describe and define the problem and then report computational results as presented in [15].

The algorithm in [15] assumes that exactly n bus trips are considered during the planning interval. The solution procedure is initiated with a fictitious fleet of n buses all located at one fictitious depot ($i = n + 1$). First, the optimal chains of trips are generated by applying the modified assignment algorithm. In a second phase, a separate simple transportation algorithm is applied to assign the buses to depots relative to their capacity and operational constraints. The ratio of traveling to and from depots to total traveling time and mileage during one day is relatively small. Therefore, the decoupling leads to insignificant deviations from the "true" optimum.

Let $I = \{1, 2, ..., i, ...; n\}$ denote the set of events corresponding to the *end* of trips in the planning interval T. Each event $i \in I$ is characterized by its time a_i and place of ending. Let $J = \{1, 2, ..., j, ...; n\}$ denote the set of events corresponding to the *start* of a trip in the planning horizon T. Each event $j \in J$ is characterized by its time b_j and place of starting. It is assumed that to each trip end in I, there corresponds a trip start in J, and vice versa, so that T contains only trips which both begin and end within the planning interval, a_i and b_j are indexed so that: $a_1 \leqslant a_2 \leqslant \cdots \leqslant a_n$ and $b_1 \leqslant b_2 \leqslant \cdots \leqslant b_n$, i.e. event i occur before k if $i < k$ $\forall i, k$. Obviously $b_i < a_i$ $\forall i$. The decision variables for $i \in I$, $j \in J$, are:

$$X_{ij} = \begin{cases} 1 & \text{if } i^{\text{th}} \text{ trip end is connected to } j^{\text{th}} \text{ trip start,} \\ 0 & \text{otherwise.} \end{cases}$$

$$X_{n+1,j} = \begin{cases} 1 & \text{if depot supplies a bus for } j^{\text{th}} \text{ trip start,} \\ 0 & \text{if not.} \end{cases}$$

$$X_{i,n+1} = \begin{cases} 1 & \text{if, after } i^{\text{th}} \text{ trip end, bus returns to depot,} \\ 0 & \text{if not.} \end{cases}$$

$$X_{n+1,n+1} = \begin{cases} \text{number of the } n \text{ available buses at the depot which remain there unused.} \end{cases}$$

The bus scheduling problem is formulated as:

Problem A

Find values of X_{ij} that satisfy

$$Z_A = \min_x \{\sum_{i=1}^{n+1} \sum_{j=1}^{n+1} Q_{ij}X_{ij}\} .$$

subject to:

$$\sum_{i=1}^{n+1} X_{ij} = 1 \qquad j = 1, 2, ..., n \tag{21}$$

$$\sum_{j=1}^{n+1} X_{ij} = 1 \qquad i = 1, 2, ..., n \tag{22}$$

$$\sum_{j=1}^{n+1} X_{n+1,j} = n \tag{23}$$

$$\sum_{i=1}^{n+1} X_{i,n+1} = n \tag{24}$$

$$X_{ij}(b_j - a_i - t_{ij} - t_1) \geqslant 0 \qquad i, j = 1, 2, ..., n \tag{25}$$

$$X_{ij}(b_j - a_i - t_{ij} - t_2) \leqslant 0 \qquad i, j = 1, 2, ..., n \tag{26}$$

$$X_{ij} \geqslant 0, \text{ and integer} \qquad i, j = 1, 2, ..., n+1. \tag{27}$$

The constraints (25), (26) ensure that trip end i can be linked to trip start j only if $t_1 \leqslant b_j - a_i - t_{ij} \leqslant t_2$, $i, j = 1, 2, ..., n$ where t_{ij} is the known driving time from event i to event j, $t_1 \geqslant 0$ is a safety measure for ensuring that the bus arrives on time at start i, and t_2 is an upper limit on driver idle time between two successive trips. Constraint (21) ensures that each trip start is supplied a bus either from the depot or from some (previous) trip completion. Constraint (22) implies that after each trip completion, the released bus heads either to the depot or some (later) trip start. Constraint (23) implies that the n buses at the depot either remain there idle or head out to trip starts; (24) has a similar interpretation.

The costs Q_{ij} are:

$$Q_{ij} = \begin{cases} D & i = n+1; j+1, 2, ..., n \\ 0 & i = 1, 2, ..., n+1; j = n+1 \\ L_{ij} = E_{ij} + K_{ij} + R_{ij} + W_{ij} & i = j = 1, 2, ..., n, \end{cases} \tag{28}$$

where $\tilde{D}$ = annual cost incurred by reducing the fleet size by one bus. It is assumed that the current fleet size is large enough so that $\tilde{D}$ is constant over a wide range of possible changes in the fleet size.

D = that portion of $\tilde{D}$ which is attributed to the planning period T.

E_{ij} = direct dead-heading cost from event i to event j including operating costs and direct costs of driver.

K_{ij} = imputed penalty cost due to disconnecting an existing link between event i and event j. This imputed cost was introduced by the bus company in order to reduce the number of frequent changes. In our case, we assume that $K_{ij} = K \geqslant 0$ if events i and j were previously not linked and is equal to zero if events i and j were previously linked.

W_{ij} = cost of idle time of driver between event i and event j. W_{ij} is a function of $b_j - a_i - t_{ij}$.

R_{ij} = imputed penalty cost due to combining the trips associated with events i and j when they correspond to two different control centers. Bus companies control the buses, drivers and bus routes through regional control centers. Each bus driver and bus route are assigned to one center only. Managerial consideration might dictate that buses and drivers will preferably be operated within one control center. In that case, the imputed cost of linking two trips belonging to two different control regions is infinite. If, however, linking is permitted, the bus company will often associate a penalty cost to compensate for the additional control costs. In our case,

$$R_{ij} = \begin{cases} R \geqslant 0 & \text{if the trips associated with events } i \text{ and } j \text{ do not belong to the same control center,} \\ 0 & \text{otherwise.} \end{cases}$$

Equation (28) indicates that the costs are comprised of a cost D for supplying a fresh bus from a depot, zero cost for buses remaining idle at the depot, and a linking cost L_{ij} for connecting the i^{th} trip end to the j^{th} trip start. If $L_{ij} \geqslant D$, then there are no cost savings if events i and j are linked. It is cheaper then to supply trip start j with a fresh bus from the depot. Hence, Ω, the set of potential linkages of interest:

$$\Omega \equiv \{(i, j) \mid i, j = 1, 2, ..., n.$$
$$t_j \leqslant b_j - a_i - t_{ij} \leqslant t_2 \text{ and } L_{ij} < D\}$$

and set

$$X_{ij} = 0 \qquad \text{for } (i, j) \in \bar{\Omega} \tag{29}$$

where $\bar{\Omega} \equiv \{(i, j) \mid i, j = 1, 2, ..., n \text{ but } (i, j) \neq \Omega\}$. This may be imposed by replacing Problem A by

Problem B

$$Z_B = \min\{\sum_{i=1}^{n+1} \sum_{j=1}^{n+1} C_{ij}X_{ij}\} \tag{30}$$

subject to (21–24), (27), where

$$C_{ij} \equiv \begin{cases} Q_{ij} & (i = n+1; j = 1, 2, ..., n+1) \text{ or} \\ & (i = 1, 2, ..., n+1; j = n+1) \\ & \text{or } (i, j) \in \Omega \\ M & (i, j) \in \bar{\Omega} \end{cases} \tag{31}$$

where M is a positive number chosen sufficiently large that every optimal solution to Problem B satisfies (29). By (28) and (31) C_{ij} is either M or at most D, it suffices to set $M > D$, because if X is a feasible solution to Problem B with $X_{rs} = 1$ for some $(r, s) \in \bar{\Omega}$, a modified solution $\bar{X}$

$$\bar{X} = \begin{cases} \bar{X}_{rs} = 0 & \bar{X}_{r,n+1} = \bar{X}_{n+1,s} = 1 \\ & \bar{X}_{n+1,n+1} = X_{n+1,n+1} - 1 \geqslant 0 \\ \bar{X}_{ij} = X_{ij} & \text{otherwise} \end{cases}$$

will improve the objective function by at least $M - D$.

Since any optimal feasible solution to Problem B satisfies (29), it will be feasible, hence suboptimal, for Problem A, hence $Z_A \leqslant Z_B$. Since any optimal feasible solution to Problem A is feasible, hence suboptimal, for Problem B and satisfies (29), $Z_B \leqslant Z_A$. Therefore, $Z_A = Z_B$. Any optimal feasible solution to Problem B is also an optimal feasible solution to Problem A.

The advantage of replacing Problem A by Problem B is that one may drop the integrality constraint (27). Problem B then reduces to the classical transportation problem, "Problem C" which may be solved by efficient algorithms. Any optimal basic feasible solution to Problem C will satisfy (27) and thus solve Problem B (hence Problem A) as well.

The special structure of restrictions (25) and (26) eliminates the need for branching in the branch and bound procedure. The initial root of the branch and bound tree identifies the optimal solution to the bus scheduling problem. The special structure of the cost matrix $[Q]$ enables to further reduce the computational time by scanning a subset of variables to enter the base. Only when no candidates are identified within the subset, we proceed to examine all possible candidates. This procedure reduces the computational time of the improvement phase by appoximately 50%. The program was written in FORTRAN. The initial feasible solution was generated by the Chain Rule method proposed in [16] and the improvement phase as described in [15]. Some computational results tested by a bus company and presented in [15] are given in Table 1. Note that problems 5–8 in Table 1 were run on an IBM 370/145. If problems 4 and 5 are used as a benchmark for comparison then a problem of 1000 to 1400 trips will require less than 10 minutes on an IBM 370/168. Detailed description of the problem is given in [15].

Table 1
Solution times as a function of problem size.

Problem number	Problem size: Number of trips	Number of buses	Number of buses saved	Generating an initial feasible solution (sec)	The improvement phase (sec)	Total solution time (sec)	Type of computer used	Comments
1	303	32	6	2.9	6.8	9.7	IBM-370/168	a
2	303	32	6	2.9	15.3	18.[illegible]	IBM-370/168	a
3	732	196	14	13	166	279	IBM-370/168	
4	883	291	31	16	253	269	IBM-370/168	b
5	883	291	31	326	2921	3247	IBM-370/145	b
6	930	215	9	370	2387	2757	IBM-370/145	
7	1203	457	30	522	5074	5596	IBM-370/145	
8	1407	484	21	617	6266	6883	IBM-370/145	

[a] The problem differs in operational constraints and in cost data.
[b] The two problems are the same.

References

[1] R.D. Angel, W.L. Caudle, R. Noonan and A. Whinston, Computer Assisted school bus scheduling, Management Sci. 18 (6) (1972) 279–288.

[2] M. Bellmore and G. Nemhauser, The travelling salesman problem: A survey, Operations Res. 16 (1968) 538–550.

[3] M. Bellmore and J.C. Malone, Pathology of travelling salesman subtour – Elimination algorithms, Operations Res. 19 (1971) 278–307.

[4] R. Bushnell and G. Venkatramn, The optimum solution to a school bus scheduling problem, Wayne State University, Detroit, Michigan (1973).

[5] J.A. Chisman, The clustered travelling salesman problem, Computers and Operations Res. 2 (1975) 115–119.

[6] N. Christofides and S. Eilon, Algorithms for large-scale travelling salesman problems, Operational Res. Quart. 23 (1972) 551–518.

[7] G. Clarke and J.W. Wright, Scheduling of vehicles from a central depot to a number of delivery points, Operations Res. 12 (1964) 568–581.

[8] G.B. Dantzig, D.R. Fulkerson and S.M. Johnson, Solution of a large scale travelling salesman problem, Operations Res. 2 (1954) 393–410.

[9] R.D. Davis, On the delivery problem and some related topics, Doctoral Thesis, Northwestern University (1968).

[10] W.L. Eastman, Linear programming with pattern constraints, Ph.D. Dissertation, Harvard University (1958).

[11] S.R. Garfinkel and C.K. Gilbert, The bottleneck travelling salesman problem, Working Paper No. 32, College of Business Administration, The University of Tennessee (1976).

[12] T.J. Gaskell, Bases for vehicle fleet scheduling, Operational Res. 18 (1967) 281–295.

[13] B. Gavish, A note on the formulation of the M-salesman traveling salesman problem, Management Sci. 22 (1976) 704–705.

[14] B. Gavish and P. Schweitzer, An algorithm for combining truck trips, Transport Sci. 8 (1974) 13–23.

[15] B. Gavish, P. Schweitzer and E. Shlifer, Assigning buses to schedules in a metropolitan area, Computers and Operations Res. 5 (2) (1978).

[16] B. Gavish, P. Schweitzer and E. Shlifer, The zero pivot phenomenon in transportation and assignment problems and its computational implicitions, Math. Programing 12 (1977) 226–240.

[17] F. Kirkman, Problems of innovation in the transport industry: A bus scheduling program, Proc. PTRC Public Transport Analysis Seminar, London, 1 (1968) 1–15.

[18] E.L. Lawler and D.E. Wood, Branch and bound methods: A survey, Operations Res. 14 (1966) 669–719.

[19] J.D.C. Little, K.G. Murty, D.W. Sweeney and C. Karel, An algorithm for solving the travelling salesman problem, Operations Res. 16 (1963) 972–989.

[20] C.E. Miller, A.W. Tucker and R.A. Zemlin, Integer programming formulation of traveling salesman problems, J. ACM (1960) 326–329.

[21] L.G. Mitten, Branch and bound methods: General formulation and properties, Operations Res. 18 (1970) 24–34.

[22] K.G. Murty, An algorithm for ranking all the assignments in order of increasing cost. Operations Res. 15 (1967) 682–687.

[23] M.R. Newton and W.H. Thomas, Bus routing in a multischool system, Computers and Operations Res. 1 (1974) 213–222.

[24] C. Orloff, Route constrained fleet scheduling, Transportation Sci. 10 (1976) 149–168.

[25] J.L. Saha, An algorithm for bus scheduling problems, Operational Res. Quart. 21 (1972) 137–148.

[26] D. Shapiro, Algorithms for the solution of the optimal cost travelling-salesman problem, Doctoral Thesis, Washington University, St. Louis (1966).

[27] J.A. Svestka and V.A. Huckfeldt, Computational experience with an M-salesman travelling salesman algorithm, Management Sci. 19 (1973) 790–799.

[28] A. Wren, Bus scheduling an interactive computer method, Center for Computer Studies, University of Leeds (1972).

[29] A. Wren, Computer in Transport Planning and Operation (Ian Allan, London, 1971).