PERGAMON

Transportation Research Part A 36 (2002) 309–333

TRANSPORTATION RESEARCH PART A

www.elsevier.com/locate/tra

Heuristic approaches for solving large-scale bus transit vehicle scheduling problem with route time constraints

Ali Haghani [a,*], Mohamadreza Banihashemi [b,1]

[a] Department of Civil and Environmental Engineering, University of Maryland, College Park, MD 20742, USA

[b] A/E Group, Inc., Geometric Design Lab, Turner-Fairbank Highway Research Center, FHWA, USA

Received 24 May 1999; received in revised form 23 October 2000; accepted 6 November 2000

Abstract

This paper presents new models for multiple depot vehicle scheduling problem (MDVS) and multiple depot vehicle scheduling problem with route time constraints (MDVSRTC). The route time constraints are added to the MDVS problem to account for the real world operational restrictions such as fuel consumption. Compared to existing formulations, this formulation decreases the size of the problem by about 40% without eliminating any feasible solution. It also presents an exact and two heuristic solution procedures for solving the MDVSRTC problem. Although these methods can be used to solve medium size problems in reasonable time, real world applications in large cities require that the MDVSRTC problem size be reduced. Two techniques are proposed to decrease the size of the real world problems. For real-world application, the problem of bus transit vehicle scheduling at the mass transit administration (MTA) in Baltimore is studied. The final results of model implementation are compared to the MTA's schedules in January 1998. The comparison indicates that, the proposed model improves upon the MTA schedules in all respects. The improvements are 7.9% in the number of vehicles, 4.66% in the operational time and 5.77% in the total cost.

Keywords: Vehicle scheduling; Heuristic; Optimization; Multiple depot; Transit

1. Introduction

Scheduling is one of the most important steps in the urban mass transit planning process. Transit schedules are usually changed three or four times a year. Vehicle scheduling or blocking is

* Corresponding author. Tel.: +1-301-405-1974; fax: +1-301-405-2585.

E-mail addresses: haghani@eng.umd.edu (A. Haghani), mohamadreza.banihashemi@fhwa.dot.gov (M. Banihashemi).

[1] Tel.: +1-202-493-3334

0965-8564/02/$ - see front matter
PII: S0965-8564(01)00004-0

sequencing series of bus trips into blocks or schedules that can be run by the buses. Each of these blocks is identified by a pull-out and a pull-in time. The length of these blocks are restricted by buses' fuel capacity and other operational constraints set forth by the agency. Designing these blocks generally takes about 30% of the total time that the schedulers spend during the schedule changing process. In most agencies the problem is split into several smaller scheduling problems for individual depots or garages and each of these problems is treated as a separate problem.

The blocking problem can be formulated and solved as a multiple depot vehicle scheduling (MDVS) problem. Bodin et al. (1983) and Ball and Bodin (1983) have a clear definition of the MDVS problem. Adding the "Route Time Constraints" restricts the blocks from being longer than a specific time. In general, the multiple depot vehicle scheduling problem with route time constraints (MDVSRTC) can be defined as chaining the trips together and constructing one day's work for each transit vehicle while satisfying the following conditions:

(a) A specific objective function is minimized. This is usually a combination of capital cost and operational cost of vehicles;
(b) Each trip is run by just one vehicle;
(c) Each depot has a limited number of vehicles;
(d) Each vehicle returns to the same depot from which it started;
(e) The total time during which a vehicle is away from its depot is limited to a pre-specified time.

The MDVSRTC can be formulated as an integer-programming problem. Due to their large-scale nature, real-world MDVSRTC problems cannot be solved exactly. Heuristic procedures are needed to find acceptable solutions for these problems. In general, the objective function of the MDVSRTC problem can be one of the followings.

(a) Minimizing the total number of vehicles, which is minimizing the capital cost or fixed cost of vehicles;
(b) Minimizing the total deadhead time or deadhead cost of the operation;
(c) Minimizing a combination of the two;

Experiences with mass transit vehicle scheduling problems have shown that choosing the third type of objective function, by considering a reasonable relation between the fixed cost and the operational cost of the vehicles, would also lead to a minimum vehicle solution.

This paper focuses on a new approach to formulating and solving the MDVSRTC problems. In the following sections, first, a brief review of existing models and formulations is presented; then, the proposed formulation is described followed by the proposed solution strategies; finally, the results of the implementations and the comparison of the results with the actual mass transit administration (MTA) schedules are discussed.

2. Existing models and solution approaches

2.1. MDVS problem

The MDVS problem is an NP hard problem. Two commonly used approaches are based on a single depot vehicle scheduling (SDVS) problem formulation and are called "Cluster First-Schedule Second" (Carraresi and Gallo, 1984), and "Schedule First-Cluster Second" (Gavish and Shifler, 1978).

The MDVS problem can be formulated as a MIP problem in two different ways: "Trip Based" formulations, in which the trips are the components to which the variables are related, and "Block Based" formulations, in which the blocks serve that purpose.

Bertossi et al. (1987) formulated the MDVS problem as a multi-commodity "matching" problem, and proposed a heuristic for solving this problem. Problems of size 50 trips and three depots are solved using the above procedure. Lamatsch (1990) proposed another multi-commodity approach that formulates the problem in a time-space network. Mesquita and Paixão (1990) presented a solution procedure for the MDVS problem represented by a multi-commodity formulation. They solved problems of size 250 trips with two depots, and 200 trips and three depots.

Forbes et al. (1994) presented a shorter form of the problem formulated as a multi-commodity problem that seems to be more practical. They solved the problem establishing a quasi assignment problem (QAP). Then from the QAP, a problem equivalent to the original problem is built, and the linear relaxation of this problem is solved starting from the QAP solution. A branch and bound technique is then used for resolving the non-integer variables to integrality. They solved problems with up to 600 trips and three depots to optimality (with average number of variables equal to 91556).

Mesquita and Paixão (1997) have reviewed different multi-commodity formulations and studied the application of the branch and bound algorithm to these formulations. The MF_x formulation presented in this paper is the same model presented by Forbes et al. (1994).

The most successful approach for solving the MDVS problem is the work of Löbel (1997). He solved large real-world problems using a specific type of column generation called "Lagrangean Pricing". He solved a problem consisting of 25,000 trips with more than 13 million variables to optimality and a similar problem with about 70 million variables to a good feasible solution. ILP-2, the basic model used by him is also equivalent to the model used by Forbes et al. (1994).

The above formulations are all trip-based. The only block-based formulation found in the literature is a column generation approach presented by Ribeiro and Soumis (1994). They formulated the problem as a set-partitioning problem with some side constraints.

Fig. 1 shows the network associated with the model presented by Forbes et al. (1994). We also present the mathematical formulation of this model. In the remainder of this paper, this model, that is also equivalent to the models used by Löbel (1997) and Mesquita and Paixão (1997), is referred as the Forbes formulation.

The mathematical form of the Forbes formulation is as follows:

$$\text{Min} \quad \sum_{d,i} a_{d,i}A_{d,i} + \sum_{i,j,d} c_{i,j,d}X_{i,j,d} + \sum_{d,i} b_{i,d}B_{i,d}, \tag{1a}$$

$$\text{s.t.} \quad \sum_{j} A_{d,i} \leqslant r_d \quad \forall d, \tag{2a}$$

$$A_{d,i} + \sum_{j} X_{j,i,d} - w_{i,d} = 0 \quad \forall i,d, \tag{3a}$$

$$B_{i,d} + \sum_{j} X_{i,j,d} - w_{i,d} = 0 \quad \forall i,d, \tag{4a}$$

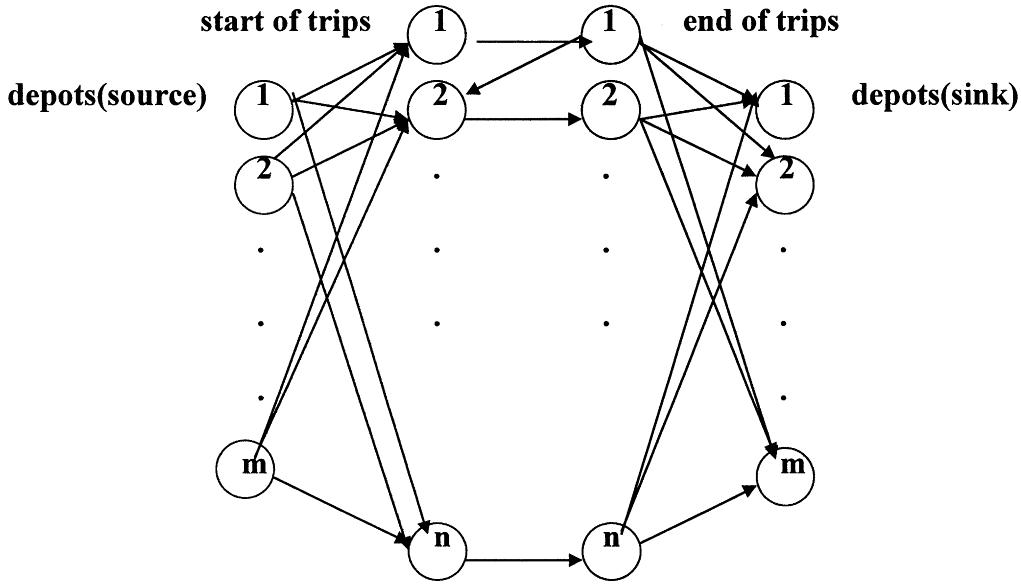

Fig. 1. The network corresponding to the Forbes formulation of MDVS problem.

$$\sum_j B_{i,d} \leqslant r_d \quad \forall d, \tag{5a}$$

$$\sum_d w_{i,d} = 1 \quad \forall i, \tag{6a}$$

All variables integer, (7a)

where *Fixed Cost* is the fixed cost of one vehicle converted into an equivalent operational cost,

$$A_{d,i} = \begin{cases} 1 & \text{if trip } i \text{ is the first trip run by a vehicle from depot } d, \\ 0 & \text{otherwise,} \end{cases}$$

$$X_{i,j,d} = \begin{cases} 1 & \text{if compatible trips } i \text{ and } j \text{ are run consecutively by a vehicle from depot } d, \\ 0 & \text{otherwise,} \end{cases}$$

$$B_{i,d} = \begin{cases} 1 & \text{if trip } i \text{ is the last trip run by a vehicle from depot } d, \\ 0 & \text{otherwise,} \end{cases}$$

$a_{d,i} =$ travel cost between depot d and the start point of trip i plus $(\mathit{Fixed\ Cost})/2$,

$$c_{i,j,d} = \begin{cases} \text{travel cost of trip } i \text{ plus the time between start time of trip } j \text{ and end time of trip } i \\ \quad \text{(if travel to the depot } d \text{ is not feasible in the time between 2 trips),} \\ \text{Min. of the above and the total of the cost of trip } i \text{ plus the travel cost from trip } i \\ \quad \text{to depot } d \text{ and from depot } d \text{ to trip } j \\ \quad \text{(if travel to the depot } d \text{ is feasible in the time between 2 trips),} \end{cases}$$

$b_{i,d}$ = travel cost from the ending point of trip i to the depot d plus travel time of trip i plus $(\mathit{Fixed\ Cost})/2$,

$$w_{i,d} = \begin{cases} 1 & \text{if trip } i \text{ is run by a vehicle from depot } d, \\ 0 & \text{otherwise,} \end{cases}$$

r_d = number of vehicles at depot d.

Considering the fixed cost for vehicles is necessary to force the optimization process to minimize both the costs associated with the number of vehicles and those associated with the deadhead times. This is done by adding this cost to the objective function through the decision variables that correspond to either vehicles leaving the depot for the first time or vehicles returning to the depot for the last time or both. Forbes et al. (1994) chose the latter by associating half of the cost to the variables indicating vehicles leaving the depot and the other half to the variables indicating vehicles returning to the depot. In the formulation presented in this paper we use the same approach as Forbes et al. (1994).

2.2. Considering the route time constraints (MDVSRTC problem)

Although schedule makers in transit agencies use manual methods to deal with route time constraints, there is some recognition of these constraints in the literature in the SDVS context. Bodin et al. (1983), Branco (1989) and Freling and Paixão (1993) have taken these considerations into account. All approaches consider the blocks as one continuous chain of trips that start with a pull-out and end with a pull-in. No consideration is given to the fact that buses that come back to the depot in the middle of the day, generally do not get refueled and therefore, their afternoon pull-out must be considered as a continuation of the morning block. These models only consider the time difference between one pull-out and the corresponding pull-in. This makes the models unsuitable for considering fuel consumption concerns.

We propose a new way for considering route time restrictions based on the actual time each vehicle might be away from the depot excluding the time that the vehicle may spend in the depot between two parts of its duty.

3. A new formulation for the MDVS problem

The following definitions are needed for presenting the new formulation.

(a) *Depot compatible trips*. Two compatible trips are referred to as "depot compatible" if it is feasible and less costly for the vehicle to return to the depot during the time between serving the trips. Löbel (1997) referred to the deadhead associated with this type of compatibility as a *pull-in-pull-out* trip.

(b) *Street compatible trips*. Two compatible trips are referred to as "street compatible" if it is infeasible or more costly for the vehicle to return to the depot during the time between serving the trips.

(c) *Morning trips, midday trips and afternoon trips*. Trips are grouped into three sets, namely, "morning trips", "midday trips" and "afternoon trips". The basic criterion for such a grouping is that all trips in the morning trips are depot compatible with all trips in the afternoon trips. The remaining trips are known as midday trips.

The new-formulation can be represented through a network as shown in Fig. 2. The components of this network are as follows. The set of nodes of the network is the same as the set of nodes in the Forbes formulation, except that there are extra transshipment depot nodes. The set of arcs of the network are the same as the set of arcs in the Forbes formulation, except that (1) there are extra arcs from the ending nodes of the morning trips to the depot transshipment nodes, and from depot transshipment nodes to the starting nodes of the afternoon trips, and (2) there are no links between the ending nodes of the morning trips and the starting nodes of the afternoon trips, although all the trips in any of these two sets are compatible with all the trips in the other set.

The lack of links between the above trip node sets is the source of advantage of this model compared to the Forbes formulation. The compatibility between trips of these sets is then replaced by different combinations of a very small number of extra links between these trips nodes and the depot transshipment nodes.

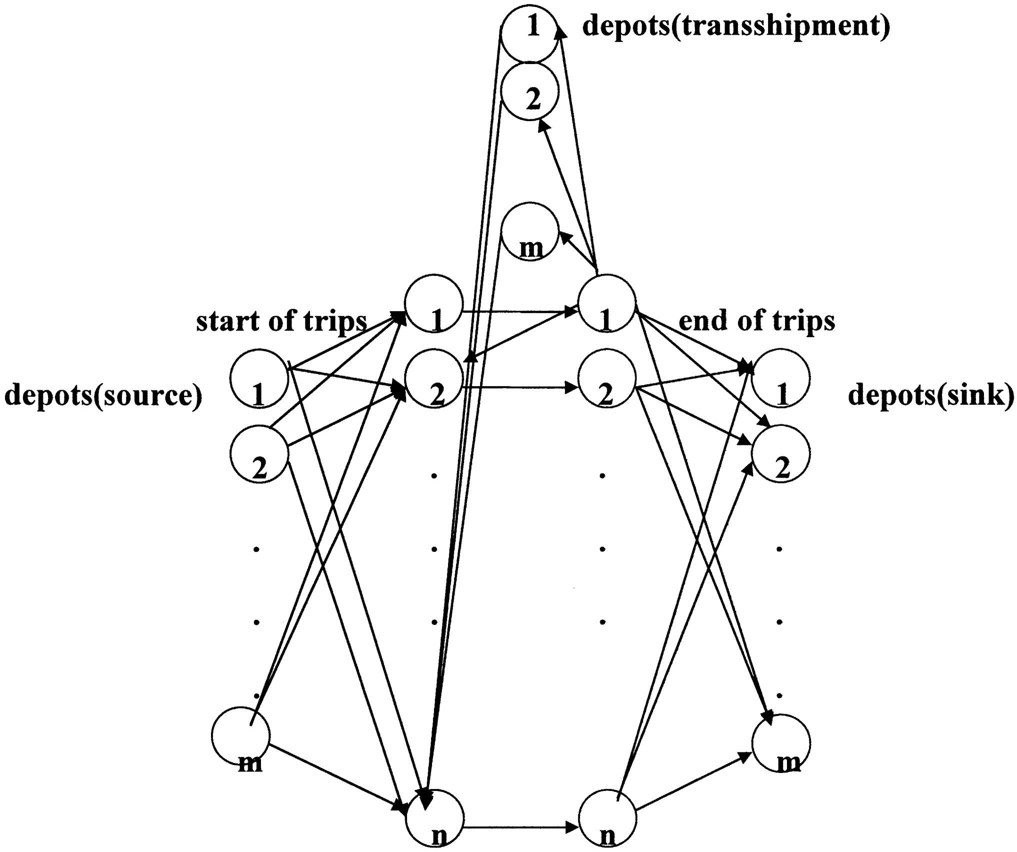

Fig. 2. Network corresponding to the new-formulation of MDVS problem.

To justify this replacement we can prove that in the MDVS problem considering the costs associated with vehicles waiting in the depot is avoidable. Therefore, for depot compatible trips the compatibility cost would be equal to the cost of the deadhead trip from the end of the first trip to the depot plus the cost of the deadhead trip from the depot to the start of the second trip.

The proposed formulation for the MDVS problem is as follows.

$$\text{Min} \quad \sum_{d,i} a_{d,i}A_{d,i} + \sum_{d,i} e_{d,i}E_{d,i} + \sum_{i,j,d} c_{i,j,d}X_{i,j,d} + \sum_{d,i} b_{i,d}B_{i,d} + \sum_{d,i} f_{i,d}F_{i,d}, \tag{1b}$$

$$\text{s.t.} \quad \sum_{i} A_{d,i} \leqslant r_d \quad \forall d, \tag{2b}$$

$$A_{d,i} + E_{d,i} + \sum_{j} X_{j,i,d} - w_{i,d} = 0 \quad \forall i, d, \tag{3b}$$

$$\sum_{i} E_{d,i} - \sum_{i} F_{i,d} = 0 \quad \forall d, \tag{4b}$$

$$B_{i,d} + F_{i,d} + \sum_{j} X_{i,j,d} - w_{i,d} = 0 \quad \forall i, d, \tag{5b}$$

$$\sum_{i} B_{i,d} \leqslant r_d \quad \forall d, \tag{6b}$$

$$\sum_{d} w_{i,d} = 1 \quad \forall i, \tag{7b}$$

$$\text{All variables integer}, \tag{8b}$$

where *Fixed Cost*, $A_{d,i}, B_{i,d}, a_{d,i}, b_{i,d}$ and r_d are the same as in the Forbes formulation.

$$E_{d,i} = \begin{cases} 1 & \text{if trip } i \text{ is in the afternoon trip set and is the first trip} \\ & \quad \text{run by a vehicle from depot } d \text{ returning to the street,} \\ 0 & \text{otherwise,} \end{cases}$$

$$X_{i,j,d} = \begin{cases} 1 & \text{if compatible trips } i \text{ and } j \text{ are run consecutively by a vehicle from depot } d \\ & \quad \text{(not for depot compatible trips } i \text{ and } j, \text{ with } i \text{ in the first and } j \\ & \quad \text{in the last trip sets),} \\ 0 & \text{otherwise,} \end{cases}$$

$$F_{i,d} = \begin{cases} 1 & \text{if trip } i \text{ is in the morning trip set and is the last trip} \\ & \quad \text{run by a vehicle from depot } d \text{ returning to the depot,} \\ 0 & \text{otherwise,} \end{cases}$$

$e_{d,i}$ = travel cost between depot d and the start point of trip i,

$$c_{i,j,d} = \begin{cases} \text{travel cost of trip } i \text{ plus the time between start time of trip } j \text{ and end time of trip } i \\ \quad \text{(if travel to the depot } d \text{ is not feasible in the time between 2 trips),} \\ \text{Min. of the above and the total of the travel cost of trip } i \\ \quad \text{plus travel cost from trip } i \text{ to depot } d \text{ and from depot } d \text{ to trip } j \\ \quad \text{(if travel to the depot } d \text{ is feasible in the time between 2 trips),} \end{cases}$$

$f_{i,d}$ = travel cost from the ending point of trip i to the depot d plus travel time of trip i.

There can be four different unit costs associated with the four different operating periods for the vehicles and crews. These periods are: (a) running the scheduled trips, (b) running the deadhead trips, (c) waiting on street (layover), (d) parking in the depot.

For defining the costs associated with the decision variables, we assume that the first three unit costs are the same and the fourth one is zero. We state these costs in terms of time. However, we can define different costs for the first three and as the following proposition states if the fourth cost is non-zero, we still can avoid its inclusion in the objective function.

Proposition 0. *In the MDVS problem, consideration of the costs associated with parking vehicles in the depots in the objective function is avoidable.*

Proof. The total time associated with the above four periods is the total vehicle availability time. Therefore, the time that vehicles are parked in the depot is not independent from the other three time periods and can be stated in terms of the other time periods and the number of vehicles. The unit cost associated with parking the vehicle in the depots is always equal to, or less than, the other three unit costs. Therefore, if this unit cost is not zero, we can eliminate it from the objective function and instead, use revised unit costs for the other periods. These revised unit costs are the original unit costs minus the unit cost associated with parking in the depot.

The above proposition justifies the cost definitions in our model. For street compatible trips, the summation of deadhead and layover costs is considered as the cost of the compatibility. For depot compatible trips, the summation of the cost of the deadhead from the first trip to the depot and the cost of the deadhead from the depot to the second trip is considered as the cost of the compatibility. And for deadhead trips from and to the depots, because there is no layover, the cost would only be the deadhead cost.

In practice, most transit agencies do not consider any cost for keeping vehicles in the depots and also consider the same unit cost for deadhead and layover. This is the basis for our assumption regarding the unit costs.

It should also be noted that this model and similar models for this problem all consider deterministic trip times and deadhead times. Different trip times or deadhead times for similar trips of a route can be incorporated for different times of day. However, transit agencies use deterministic trip times in setting time tables. In reality, the trip times are all stochastic. Due to the size of the MDVS problems an attempt to formulate it as a stochastic problem adds another level of complexity that makes the problem significantly more difficult to solve.

In the following paragraphs, we prove that our new-formulation is equivalent to the Forbes formulation. We prove that every feasible solution of each of these problems is also feasible for the other, and the objective function values are also the same.

Before proving the following propositions, it should be noted that the $X_{i,j,d}$ terms in the Forbes formulation exist for all compatible pairs of trips, but $X_{i,j,d}$ terms in the proposed formulation exist for compatible trips i and j only if trips i and j are not in the morning trip set and afternoon trip set, respectively. On the other hand, $E_{d,i}$ or $F_{i,d}$ in the proposed formulation, exist if trip i is in the afternoon trip set or morning trip set, respectively. In the following propositions, set of $X_{i,j,d}$ terms in the Forbes formulation would be referred to as X, set of $X_{i,j,d}$ in the proposed formulation would be referred to as X_1, and set $X \backslash X_1$ would be referred to as X_2.

Proposition 1. *All feasible solutions of the new-formulation are feasible for the Forbes formulation.*

Proof. Given a feasible solution for the new-formulation, constraint sets (2b)–(8b) hold. We need to prove that constraint sets (2a)–(7a) also hold as well. Constraint sets (2a), (5a)–(7a) are exactly the same as constraint sets (2b), (6b)–(8b). If constraint set (3b) holds, then the following possibilities exist.

Either $w_{i,d} = 0$, then, all terms of (3a) would be equal to zero and (3a) will hold.

Or $w_{i,d} = 1$, then the following possibilities exist.

Either $A_{d,i} = 1,\ E_{d,i} = 0,\ \sum_j X_{j,i,d} = 0$ (for $X_{j,i,d}$ in X_1) and because $E_{d,i} = 0$, we can conclude that $\sum_j X_{j,i,d} = 0$ (for $X_{j,i,d}$ in X_2) or in general $\sum_j X_{j,i,d} = 0$, and therefore, constraint set (3a) will hold.

Or $A_{d,i} = 0,\ E_{d,i} = 1,\ \sum_j X_{j,i,d} = 0$ (for $X_{j,i,d}$ in X_1) then, the following possibilities exist.

Either $\sum_j X_{j,i,d} = 0$ (for $X_{j,i,d}$ in X_2) which results in $E_{d,i} = 0$ which is not consistent with the initial assumption.

Or $\sum_j X_{j,i,d} = 1$ (for $X_{j,i,d}$ in X_2) or in general $\sum_j X_{j,i,d} = 1$, therefore, constraint set (3a) will hold ($\sum_j X_{j,i,d}$ for $X_{j,i,d}$ in X_2 cannot be more than 1 because then, $E_{d,i}$ would be more than 1).

Or $A_{d,i} = 0,\ E_{d,i} = 0,\ \sum_j X_{j,i,d} = 1$ (for $X_{j,i,d}$ in X_1) then, the following possibilities exist.

Either $\sum_j X_{j,i,d} = 0$ (for $X_{j,i,d}$ in X_2) then, constraint set (3a) will hold.

Or $\sum_j X_{j,i,d} > 0$ then $E_{d,i} > 0$ which is not consistent with the initial assumption.

Therefore in general, if (3b) holds, then (3a) would also hold. Using similar logic we can prove that if (5b) holds then (4a) would also hold. Therefore, feasible solutions of the new-formulation are feasible for the Forbes formulation.

Proposition 2. *All feasible solutions of the Forbes formulation are feasible for the new-formulation.*

Proof. For a feasible solution of a problem stated by the Forbes formulation, constraint sets (2a)–(7a) hold. We need to prove that constraint sets (2b)–(8b) also hold. Constraint sets (2b), (6b)–(8b) are exactly the same as (2a), (5a)–(7a). If constraint set (3a) holds, then following possibilities exist.

Either $w_{i,d} = 0$, then all terms of (3b) would be equal to zero and (3b) holds.

Or $w_{i,d} = 1$, then the following possibilities exist.

Either $A_{d,i} = 1,\ \sum_j X_{j,i,d} = 0$ (for $X_{j,i,d}$ in X) therefore, $\sum_j X_{j,i,d} = 0$ (for $X_{j,i,d}$ in X_1) therefore, constraint set (3b) hold.

Or $A_{d,i} = 0, \sum_j X_{j,i,d} = 1$ (for $X_{j,i,d}$ in X) then the following possibilities exist.

Either $\sum_j X_{j,i,d} = 1$ (for $X_{j,i,d}$ in X_1) and $\sum_j X_{j,i,d} = 0$ (for $X_{j,i,d}$ in X_2) and therefore, $E_{d,i} = 0$ and constraint set (3b) holds.

Or $\sum_j X_{j,i,d} = 0$ (for $X_{j,i,d}$ in X_1) and $\sum_j X_{j,i,d} = 1$ (for $X_{j,i,d}$ in X_2) and therefore, $E_{d,i} = 1$ and constraint set (3b) holds. Using similar logic we can prove that if (4a) holds, then (5b) would also hold. If we add [(3b) − (5b) − (3a) + (4a)] over d, the result would be equal to zero, which is equal to (4b), therefore it holds. Therefore, each feasible solution of the Forbes formulation is feasible for the new-formulation.

Proposition 3. *A solution that is feasible for both models has the same objective function value in both models.*

Proof. We can rewrite the objective function of the Forbes formulation as follows.

$$\sum_{d,i} a_{d,i}A_{d,i} + \sum_{i,j,d} c_{i,j,d}X_{i,j,d}(\text{for } X_1) + \sum_{i,j,d} c_{i,j,d}X_{i,j,d}(\text{for } X_2) + \sum_{d,i} b_{i,d}B_{i,d}.$$

Considering the fact that for each nonzero variable $X_{i,j,d}$ in X_2, there is one nonzero variable $E_{d,i}$ and one nonzero variable $F_{i,d}$ and considering the equality of $c_{i,j,d}$ with sum of $e_{d,j}$ and $f_{i,d}$ from definitions, we can see that the value of objective functions are also the same.

From the above three propositions we can conclude that the Forbes formulation and the new-formulation are equivalent.

4. Considering route time constraints

To deal with route time restrictions we add the following set of constraints to the problem.

$$A_{d,t_1} + X_{t_1,t_2,d} + X_{t_2,t_3,d} + \cdots + X_{t_{(p-1)},t_p,d} + B_{t_p,d} \leqslant p$$
$$\forall d \text{ and } \forall \text{ blocks with block time greater than } T_{\max}, \tag{9b}$$

where p is the number of trips of the block, t_i is the ith trip of the block. $T_{\max}$ is the maximum allowed block time.

If we add this set of constraints to the MDVS problem, they would prevent the solutions containing any infeasible block with respect to illegal block time. However, these constraints must be written for every possible block that can be constructed. In reality, the number of possible blocks is extremely large and it is impractical to incorporate all of them in the model. Instead, we build and add these constraints to a streamlined version of the MDVS problem formulation as needed to prevent the generation of infeasible blocks.

The new formulation for the MDVS problem can be streamlined into the following shorter form.

$$\text{Min} \quad \sum_{d,i} a_{d,i}A_{d,i} + \sum_{d,i} e_{d,i}E_{d,i} + \sum_{i,j,d} c_{i,j,d}X_{i,j,d} + \sum_{d,i} b_{i,d}B_{i,d} + \sum_{d,i} f_{i,d}F_{i,d}, \tag{1c}$$

$$\text{s.t.} \quad \sum_{i} A_{d,i} \leqslant r_d \quad \forall d, \tag{2c}$$

$$\sum_{d} B_{i,d} + \sum_{j,d} X_{i,j,d} + \sum_{d} F_{i,d} = 1 \quad \forall i, \tag{3c}$$

$$\sum_{i} E_{d,i} - \sum_{i} F_{i,d} = 0 \quad \forall d, \tag{4c}$$

$$A_{d,i} + E_{d,i} + \sum_{j} X_{j,i,d} - B_{i,d} - F_{i,d} - \sum_{j} X_{i,j,d} = 0 \quad \forall i, d, \tag{5c}$$

$$\text{All variables integer.} \tag{6c}$$

The definitions of variables are as before and one set of variables ($w_{i,d}$) is eliminated from the formulation. It can easily be proven that this formulation is equivalent to the formulation of

Table 1
A comparison of the number of varaibles in the Forbes and the new formulations

Number of trips	Average number of variables in the Forbes formulation	Average number of variables in the new formulation	Percentage decrease in the number of variables
300	149,363	91,096	39.0
400	272,688	160,132	39.6
500	417,888	247,252	40.9
600	605,753	358,928	40.8
700	824,540	484,004	42.3
800	1,071,612	619,184	41.4
900	1,350,148	786,360	41.0

MDVS represented by (1b)–(8b). In this formulation, the objective function (1c) minimizes the total cost including the capital cost and the operational cost. The cost is stated in terms of "minutes of operation". Constraint set (2c) ensures that the total number of the blocks started by vehicles from each depot is limited to the number of available vehicles in that depot. Constraint set (3c) ensures that if trip i is run by a vehicle from depot d, it is either run before another trip run by a vehicle from that depot, or is the last trip of a block run by a vehicle from that depot, or is the last trip of a vehicle that returns to the street in the middle of the day. This constraint ensures that all of the trips are run. Constraint set (4c) ensures that the number of the vehicles that return to the depots in the middle of the blocks is equal to the number of vehicles that return to the street in the middle of the blocks. Constraint set (5c) ensures that the vehicles return to the same depot they started from. Finally, constraint set (6c) is the set of integrality constraints.

Constraints that are built based on the limitation stated in the inequality (9b) would be added as needed to the MDVS problem to ensure that the construction of the blocks with route times more than the maximum allowed route time is prevented.

A comparison of the number of variables in the new formulation and that of the Forbes formulation as suggested by Forbes et al. (1994) for several test problems is depicted in Table 1. As Table 1 indicates, the new formulation reduces the number of variables by about 40%.

In the next section, we briefly discuss the test problem generation procedure. For all of the experiments (except for some of the MDVS problems) a PENTIUM Pro 200 PC is used which is referred to as the Test Machine.

5. Test problem generation based on the MTA operation

The characteristics of the January 1998 schedule of the MTA operation and the details of the code that was used to generate test problems base on those characteristics are given in Banihashemi (1998). The MTA operation information is summarized in Table 2 and Fig. 3.

Fig. 3 shows the number of trips starting and also the number of trips operating in one-hour intervals during a day. For each specific time interval, "Trips Starting" is the number of trips that start in that time interval, and "Trips Running" is the number of the trips that are operating during that time interval.

Table 2
Summary of the January 1998 schedule of MTA operation

Deadhead time for trips from the depots to the starting points (min)	18530
Deadhead time for trips between the compatible trips (min)	4131
Deadhead time for trips from the ending point to the depots (min)	18111
Total layover time (min)	37805
Total vehicle cost in terms of minutes of operation (min)	186000
Total trip time (min)	277666
Total cost in terms of minutes of operation (min)	542243

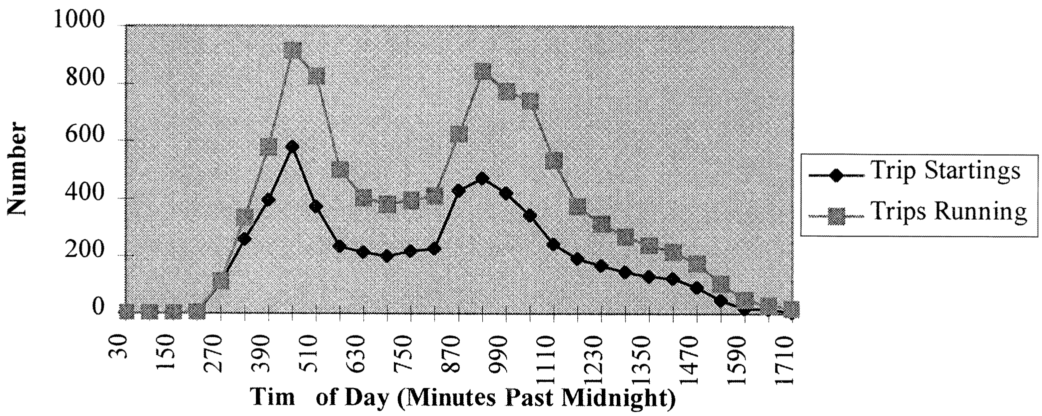

Fig. 3. Starting times and trips running in one-hour intervals for MTA trip data.

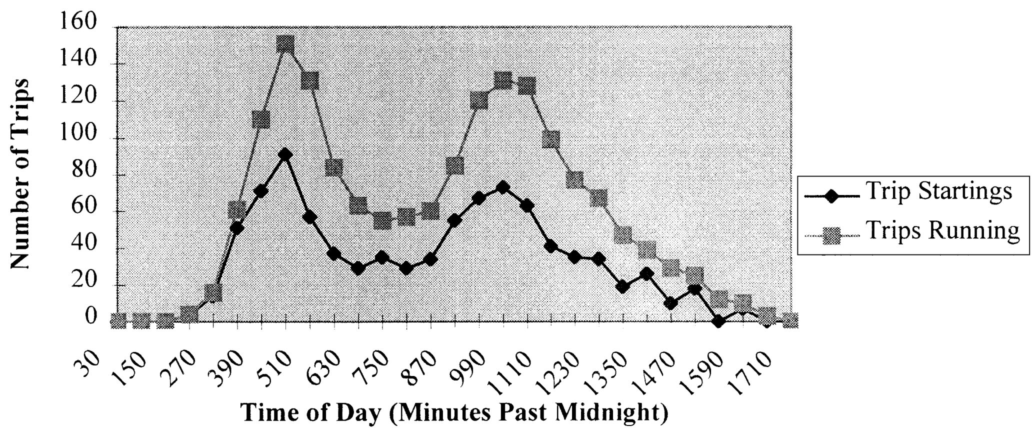

Fig. 4. Starting times and trips running in one-hour intervals for 900-trip problem.

Seven classes of test problems were generated, with 300, 400, 500, 600, 700, 800 and 900 trips. Five test problems were generated in each of the 300–600 trip categories, and one test problem was generated in each of the 700–900 trip categories.

Fig. 4 shows the number of the trips starting and the number of trips operating in one-hour intervals for the 900-trip test problem. Comparing this figure to Fig. 3 we can see that the distributions of the trips in a day is quite similar to the distribution of the trips in the MTA operation. All test problems have characteristics similar to the 900-trip problem.

6. Solution procedures for the MDVSRTC problem

One exact and two heuristic procedures are proposed for solving the MDVSRTC problem. The procedures are applied to the test problems and the solutions are all compared to the solutions of the corresponding MDVS problems whose objective function values can be considered as lower bounds for the MDVSRTC problems. Table 3 shows the solutions of the MDVS test problems.

6.1. The exact solution procedure

There is one constraint associated with each block ensuring that the total route time for the block does not exceed the maximum allowed route time. The number of these constraints is very high and all of these constraints cannot be generated and added to the problem. Instead, a constraint generation approach is proposed to solve the MDVSRTC problem. The procedure is summarized as below.

Step 1. Solve the MDVS problem to optimality.

Step 2. Construct the established blocks and identify those violating the route time constraint. If there is no such block, the solution is optimal for the MDVSRTC problem as well, STOP.

Step 3. Corresponding to each violated block with p trips, there are $p+1$ variables in the solution equal to 1. For each violated block add a constraint to the current problem which insures that the summation of the $p+1$ variables corresponding to the violated block is less than or equal to p.

Step 4. Solve the newly created problem in Step 3 and go to Step 2.

Based on our experience, the time required for solving the problem with the addition of these constraints usually increases with each iteration and there is no guarantee to obtain the optimum solution in a reasonable time. In our test problems, the number of additional iterations due to the generated route time constraints varied from 0 to 60, and in some cases at each additional iteration more non-integer variables resulted that had to be added to the problem as integer variables. The test problems were solved using the exact solution procedure. Table 4 shows the results of the solutions.

As we can see the gap between the MDVS solution and MDVSRTC solution is almost zero in all cases. Our experience shows that the gap between the LP solution of MDVS and the IP solution of the MDVS is also almost zero.

Table 3
Test result for the MDVS problem

Number of trips (Problem number)		Number of variables	LP solution		Integer	IP solution		Integer
			Value	Time (min)		Value	Time (min)	
300	(1)	99556	38092	14	Yes	38092	–	–
	(2)	88204	31529	10	Yes	31529	–	–
	(3)	88212	40278.1	15	No	40283	4	Yes
	(4)	85436	38798	14	Yes	38798	–	–
	(5)	94072	37394.5	12	No	37396	3	Yes
400	(1)	168632	49691	54	No	46964	6	Yes
	(2)	160132	45493.4	34	No	45503	200	Yes
	(3)	158356	50488.5	38	No	50496	67	Yes
	(4)	158700	43141	46	Yes	43141	–	–
	(5)	169360	46543.7	32	No	46549	15	Yes
500	(1)	233220	60228	89	Yes	60228	–	–
	(2)	267616	50795	108	Yes	50795	–	–
	(3)	249444	59544.5	95	No	59547	47[a]	Yes
	(4)	252856	58998.9	93	No	59000	201[a]	Yes
	(5)	233124	56315	103	No	56317	203[a]	Yes
600	(1)	371180	55999	319[a]	No	55999	350[a]	Yes
	(2)	363088	59346	335[a]	Yes	59346	–	–
	(3)	337388	69537.7	268[a]	No	69541	317[a]	Yes
	(4)	378076	61030	494[a]	Yes	61030	–	–
	(5)	344908	77947	312[a]	No	77947	497[a]	Yes
700	(1)	484004	75413.2	965[a]	No	75415	60[a]	Yes
800	(1)	628256	83720.2	1750[a]	No	83722	60[a]	Yes
900	(1)	786360	98015.3	2510[a]	No		Not solved	

[a] These were solved on a SUN SPARC5 machine. The solution time cannot be compared to the solution times on the PC.

Table 4
Test results of the exact solution procedure for the MDVSRTC problem

Number of trips	(Problem number)	MDVS problem solution	Additional iterations due to RT constraints	Total solution time (min)	Number of constraints added	Optimum solution	Gap with MDVS (%)
300	(1)	38092	–	14	–	38092	0.0
	(2)	31529	1	37	1	31529	0.0
	(3)	40283	–	19	–	40283	0.0
	(4)	38798	–	14	–	38798	0.0
	(5)	37396	13	282	20	37399	0.0
400	(1)[a]	46961	60	4230	90	46970	[a]
	(2)	45503	1	336	2	45504	0.0
	(3)	50496	–	105	–	50496	0.0
	(4)	43141	7	463	6	43141	0.0
	(5)	46549	–	47	–	46549	0.0

[a] After 60 iterations the optimum solution was not found and the procedure was stopped. The current value is the last solution with two violated blocks. The 500 trip problems and larger problems could not be solved using the "exact solution procedure" on the test machine.

6.2. Heuristic procedure 1 (H1)

This heuristic is a constraint generation approach. Its major steps are similar to the steps of the exact solution procedure, except that at each iteration, a legal block (with block time equal to, or less than, the maximum allowed time) is generated from each violated block. The variables associated with these blocks are set to 1, and these equalities are added to the problem as generated constraints. The added constraints are fixed in the solution by saving the problem at each iteration, after adding the constraints. The legal blocks are built by eliminating the latest trip(s) from the list of trips one after another until the block time becomes legal. Using this procedure, first the legal blocks are added to the schedule and are fixed, and second, the potential of violated blocks being built in the solution is decreased.

The steps of this solution procedure are summarized as follows.

Step 1. Solve the MDVS problem to optimality.

Step 2. Construct the established blocks and identify the blocks with route time violation. If there is no such block, the solution is the optimal solution for the MDVSRTC problem as well, STOP.

Step 3. Corresponding to each violated block with p trips, there are $p+1$ variables equal to 1 in the solution. For each violated block containing p trips ($p+1$ variables), decrease the number of trips one by one until the block time becomes legal. This would be a legal block built from a violated block. Build a constraint set containing the equality constraints (variable = 1) for all of the variables associated with all of the legal blocks and add these constraints to the problem.

Step 4. Solve the newly created problem in Step 3 and go to Step 2.

Table 5 shows the results of applying the heuristic solution H1 on the 300 and 400 trip problems.

Table 5
Test results for the heuristic procedure 1

Number of trips	(Problem number)	Additional iterations due to RTCs	H1 solution	Total solution time (min)	Total # of constraints added	MDVS solution	Gap with MDVS solution (%)
300	(1)	–	38092	14	–	38092	0.00
	(2)	1	31529	27	1	31529	0.00
	(3)	–	40283	19	–	40283	0.00
	(4)	–	38798	15	–	38798	0.00
	(5)	3	37504	30	3	37396	0.28
400	(1)	3	47184	190	6	46961	0.51
	(2)	1	45561	322	1	45503	0.13
	(3)	–	50496	105	–	50496	0.00
	(4)	2	43167	102	2	43141	0.06
	(5)	–	46549	47	–	46549	0.00

As the results show, this procedure reaches an acceptable solution within a reasonable time. Similar to the exact solution procedure, the 500 trip problems could not be solved using this procedure on the test machine.

6.3. Heuristic procedure 2 (H2)

This heuristic relies on the possibility of getting integer feasible solutions without using the integer-programming module of the optimization software. From both integer and non-integer solutions of the MDVS problem we can always build a complete set of blocks, with each block represented by $p+1$ nonzero variables, where p is the number of trips in that block.

The following definitions are necessary for describing the heuristic procedure 2.

Integer block. A block is called an integer block if all of the variables associated with that block are integer (are equal to 1). In this case, the block has a weight of 1.

Non-integer blocks. A block is called a non-integer block if at least one of the variables associated with that block is non-integer. In this case the weight of the block is equal to the value of the smallest variable constructing that block.

If the solution were integer, all of the blocks would be integer. If the solution were non-integer, some of the blocks would be non-integer.

Similar to the procedure H1, this heuristic is also takes an iterative constraint generation approach, but there are two sets of constraints added to the original problem at each iteration, as follows.

1. Besides the maximum block time, a minimum route time is also determined. Variables involved in the construction of integer blocks are set to one based on the following rules:
   (a) For the integer blocks with route times between the minimum route time and the maximum route time, all variables are fixed to 1.
   (b) For the integer blocks with route times more than the maximum route time, a legal block

is built by eliminating as many trips as needed from the end of the blocks. Then the variables corresponding to this legal block are fixed to 1.

(c) For the integer blocks with route times less than the minimum route time, all variables except those corresponding to the return of vehicles to the depots are fixed to 1.

(d) At each iteration the number of integer blocks is compared to the number of integer blocks of the previous iteration. If it were equal or less, then the variables corresponding to one of the non-integer blocks would also be set to 1 following the above rules.

2. If the solution is non-integer, an inequality constraint is added to the problem to prevent the construction of the same set of non-integer variables with the same values in the next iteration. This constraint is built from the list of the variables included in the non-integer blocks. There is an exception if rule 4 from the above rules is used. In this case, the values of the non-integer variables included in the non-integer block, which were set to 1 would be considered 1 and not the original value.

Table 6 shows the results corresponding to the test problems that are quite satisfactory especially regarding the gap.

This approach relies strictly on the LP module of the optimization software and its advantages over the first heuristic are that the solution time at each iteration is less, and at each iteration we can use the previous solution using the dual simplex algorithm. This procedure also has a disadvantage over the first heuristic. There might be some occasions that the problem becomes infeasible while the original problem is feasible. For these occasions, we can eliminate the constraints that were added in the last iteration and set the variable(s) that make the problem infeasible, to 0. This happened once in all of our test runs.

Table 6
Test results for the heuristic procedure 2

Number of trips	(Problem number)	Additional iterations	H3 solution	Total solution time (min)	MDVS solution	Gap with the MDVS solution (%)
300	(1)	–	38902	14	38092	0.00
	(2)	–	31529	10	31529	0.00
	(3)	6	40279	30	40823	0.00
	(4)	–	38798	14	38798	0.00
	(5)	12	37401	26	37396	0.00
400	(1)	15	47172	104	46961	0.42
	(2)	19	45516	134	45503	0.03
	(3)	–	50496	105	50496	0.00
	(4)	10	43176	81	43141	0.08
	(5)	–	46549	47	46549	0.00
500	(1)	–	60228	89	60228	0.00
	(2)	–	50795	108	50795	0.00
	(3)	17	59727	245	59547	0.30
	(4)	20	59019	893	59000	0.03
	(5)	13	56347	183	56317	0.05

7. Reducing the size of large-scale problems

The real-world MDVSRTC problems are much larger than the problems that can be solved optimally using existing computer capabilities. The number of trips in a medium-size city may be 6000 or even more. A problem with this number of trips may have more than 20 million variables. We may be able to solve the corresponding MDVS problem using existing approaches such as the one used by Löbel (1997). But adding the route time constraints to the problem and building the MDVSRTC problem changes the nature of the problem. The results of the exact solution procedure show that adding even a few route time constraints to the problem makes the problem much more complicated and increases the solution time significantly. This complexity makes the large MDVSRTC problems practically unmanageable.

In this section, two techniques for decreasing the size of very large problems are presented. These techniques are: (1) decreasing the number of trips by joining trips to make one trip from two or more trips, and (2) decreasing the number of variables by eliminating the ones that have little chance of being in an acceptable solution.

7.1. Decreasing the number of trips by joining trips

The number of trips in a problem can be decreased by using some criteria to join two or more trips a priori and considering them as a single trip. Considering a "maximum layover time", two or more trips can be considered as one trip if the following conditions exist:

1. All trips are pair-wise compatible;
2. In all compatible pairs of trips, the ending point of the first trip matches the starting point of the second;
3. The total time from starting the first trip to ending the last one, including the time required for deadhead trips from and to depots (for all depots), should be no more than the maximum allowed time;
4. For all compatible pairs of trips, the time difference between the starting time of the second trip in the trip-pair and the ending time of the first trip in the trip-pair is not greater than the maximum layover time.

We applied this method to four sample problems, a 300-trip, a 400-trip, a 500-trip, and a 600-trip problem, with the maximum layover time of 15 min. Table 7 shows the results.

Each new trip created from joining several trips has the following characteristics:

(a) its starting point and time would be the starting point and time of the first trip;

(b) its ending point and time would be the ending point and time of the last trip;

Table 7
Maximum layover method for joining the trips

Number of trips (Problem number)		New number of trips	Maximum layover solution	Gap with the MDVS solution (%)
300	(1)	260	38167	0.2
400	(1)	350	47015	0.1
500	(1)	428	60395	0.3
600	(1)	507	56154	0.3

(c) its duration would be the period between the starting time of the first trip and the ending time of the last one.

7.2. Decreasing the number of variables

The number of variables can be reduced in a preprocessing step. The compatibility variables $X_{i,j,d}$ are the best candidates for this variable reduction because more than 90% of the variables are compatibility variables and the percentage increases with the increase in the size of the problem. There is a cost associated with each compatibility variable. This cost is the summation of two costs, the cost of the deadhead travel between the two compatible trips, and the cost of the first trip. Based on analysis of the test problems, we concluded that the 95th percentile of the costs

Table 8
Decreasing the number of variables

Number of trips (Problem number)		(%) Decrease in the no. of variabless	Solution	Solution time (min)	Gap with the MDVS solution		
					Total gap	Vehicle cost	Operating cost
300	(1)	81.0	38980	12	2.3	1.6	0.7
	(2)	76.6	32546	12	3.2	2.9	0.3
	(3)	76.9	42645	11	5.9	5.2	0.7
	(4)	77.0	39424	13	1.6	1.6	0.0
	(5)	76.7	37424	15	0.1	0.0	0.1
400	(1)	77.7	48837	36	4.0	3.8	0.2
	(2)	77.2	47157	25	4.3	2.0	2.3
	(3)	77.0	50919	27	0.8	0.6	0.2
	(4)	77.8	43542	35	0.9	0.7	0.2
	(5)	77.4	48813	28	4.9	3.9	1.0
500	(1)	77.9	60986	28	1.3	1.0	0.3
	(2)	78.3	51837	32	2.1	1.2	0.9
	(3)	78.0	60074	73	0.9	0.5	0.4
	(4)	80.0	59029	88	0.1	0.0	0.1
	(5)	77.9	56366	45	0.1	0.0	0.1
600	(1)	78.7	57776	67	3.2	1.6	1.6
	(2)	78.7	61419	83	3.5	3.0	0.5
	(3)	78.4	70083	63	0.8	0.4	0.4
	(4)	78.2	63802	88	4.5	3.9	0.6
	(5)	78.9	78392	144	0.6	0.4	0.2
700	(1)	78.2	79142	140	4.9	4.4	0.5
800	(1)	78.7	86256	209	3.0	2.2	0.8
900	(1)	78.2	107082	207	9.3	8.0	1.3
Average					2.7	2.1	0.6

associated with the deadhead travel of compatibility variables chosen in the optimal solution of the MDVS problem is at about the 20th percentile of the cost of the deadhead travel of all compatibility variables of the problem. This means that eliminating 80% of the compatibility variables with higher deadhead travel costs only affects 5% of the compatibility variables in the optimal solution.

For all test problems the new problems were constructed by eliminating the compatibility variables with the deadhead travel costs higher than the 20th percentile of the deadhead travel costs of all compatibility variables. The average reduction in problem size was 78%. The solution results are shown in Table 8. We can choose some other percentiles for this reduction depending on the problem size. The solution times in the table include the time required for preparing the new problem, but not the time required for preparing the original problem and finding the 20th percentiles.

8. Sensitivity analysis

The most important factors affecting the solutions of the MDVS and MDVSRTC problems are the fixed cost of one vehicle, the number of vehicles assigned to each depot, and the maximum allowed route time. The results of the analysis of the sensitivity of the model with respect to these parameters and some other parameters are presented in detail in Banihashemi (1998). In this paper, we provide a summary of this analysis.

The first parameter tested was the fixed cost for the vehicles. Starting with the fixed cost equal to zero, we experimented by increasing the fixed cost. Increasing the fixed cost causes a dramatic decrease in the number of vehicles. When the fixed cost equals 50, the number of vehicles reaches the minimum number needed for servicing the trips. Fig. 5 shows the variation of total cost, vehicle cost and operational cost versus fixed cost and Fig. 6 shows the variation of the number of vehicles versus fixed cost for one of the test problems.

The next parameter tested was the vehicle allocation. In an MDVS problem an appropriate vehicle allocation (the number of vehicles available in different depots) may decrease the total cost significantly, but it does not change the vehicle cost or the total number of vehicles used in the solution. The change in the total cost is due to the change in the operational cost. Assuming that all depots have sufficient vehicles to serve all trips, we found the best allocation for the vehicles for some test problems. We then varied the vehicle allocations among the depots and made some test runs. The results show that properly allocating vehicles among depots can result in significant savings for a transit agency. In our test problems, on average, there was a 12.6% difference in total cost between the best and worst vehicle allocation.

The third parameter tested was the basic parameter distinguishing the MDVSRTC problem from the MDVS problem. This is the "maximum allowed block-time", namely, $T_{\max}$. We tested the problems for $T_{\max}$ equal to 960, 1020, 1080, 1200, 1320 and 1440. As the results indicate as the $T_{\max}$ increases fewer block violations are found in the MDVS solution and as a result, fewer iterations are needed to obtain the MDVSRTC solution. Of course the total cost decreases as $T_{\max}$ increases. Fig. 7 shows the total cost as a function of $T_{\max}$ for one of the test problems.

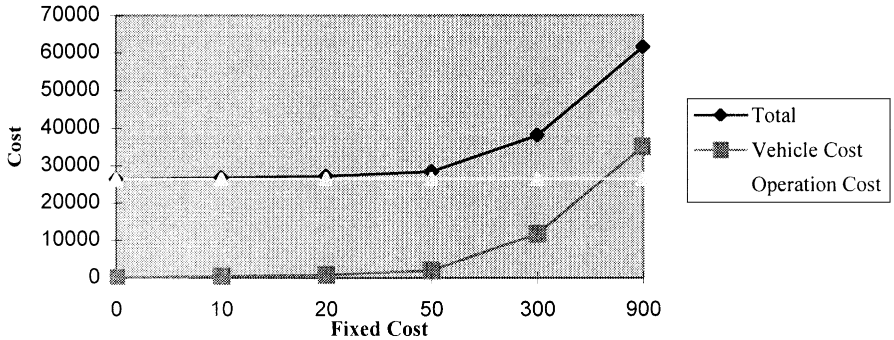

Fig. 5. Total cost, vehicle cost, and operational cost vs. fixed cost for a 300 trip problem.

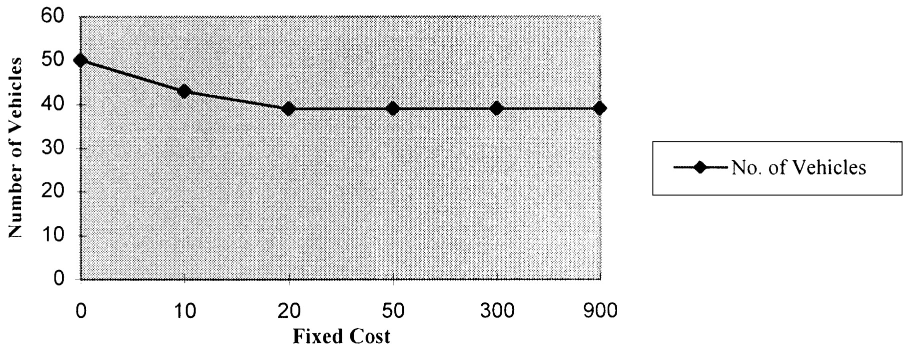

Fig. 6. No. of vehicles vs. fixed cost for a 300 trip problem.

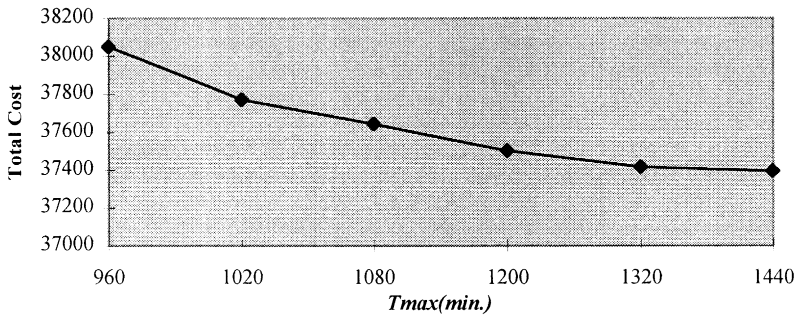

Fig. 7. Total cost vs. $T_{\max}$.

9. Case study

In this section, we apply a solution strategy to the real-world large-scale MDVSRTC problem of the MTA in Baltimore City. As discussed before, this problem has 5650 trips running from four depots.

9.1. Solution strategy

In this solution strategy, we use the new formulation of the MDVSRTC problem and heuristic procedure 1. We also use the two methods presented for decreasing the size of the original problem. The solution strategy has the following steps.

Step 1. Choose 15 min as the maximum layover time to combine the trips in the original trip set. Build the MDVS problem from this new trip set by relaxing the route time constraints. Name this problem "Revision 1" problem.

Step 2. Apply the method for decreasing the number of variables using the 7th percentile of the deadhead costs of the compatibility variables. This percentile is chosen to make the size of the problem more manageable and solvable, considering our computer capability. Name the new problem "Revision 2" problem.

Step 3. Solve the Revision 2 MDVSRTC problem using the heuristic procedure number 1.

Step 4. Partition the Revision 1 problem into 4 SDVSRTC sub-problems, one for each depot, using the solution found in *Step* 3. Solve each of the sub-problems separately using the heuristic procedure number 1. Combine the solutions. This is the solution to the Revision 1 problem and also to the original MDVSRTC problem.

Applying this procedure to the problem has the following results. By applying *Step* 1 to the original trip set, a trip set containing 2218 trips is built. This Revision 1 problem has 5,168,108 variables and 11,098 constraints. The required time for the preparation process is about 720 min (12 h) but the trip combination process is very fast taking no more than 5 min.

Applying *Step* 2 we build the Revision 2 problem with 276,265 variables and 11,098 constraints. This step takes about 60 min. The solution of the Revision 2 MDVSRTC problem requires 524,446 min of operation and includes the fixed cost of 605 buses used in the solution. The total solution time for this step is about 520 min.

The Revision 2 problem is partitioned into four SDVSRTC sub-problems with 585, 450, 757 and 426 trips at depots 1, 2, 3 and 4, respectively. By applying the heuristic procedure 1 to these four problems, we find the final solution that is 510,939 min of operation with 571 buses used in the solution. This step takes about 120 min.

9.2. Comparing the solution to the MTA solution

The comparison of our solution to the solution of the MTA, Baltimore is shown in Table 9. The total times required for applying this strategy to the problem of the MTA, Baltimore is 1425 min. As this table indicates significant savings in terms of number of vehicles used and the total operational costs can be realized using the model proposed in this research. These results were presented to the MTA managers and currently they are in the process of a more detailed evaluation and possible test implementation of the model.

9.3. Comparison of single depot problem solutions with the MTA solution

The previous solution of the MTA problem was based on the multiple depot formulation. This solution produced a different assignment of trips to depots compared to the MTA allocation. We

Table 9
Comparing the solution to the MTA solution

Solution	Vehicle cost (min)	Operating cost (min)	Total cost (min)	# of vehicles used	Solution time (min)	Saving over MTA solution (%)		
						Vehicle cost	Operating cost	Total cost
MTA	186000	356243	542243	620				
Proposed strategy	171300	339639	510939	571	1425	7.90	4.66	5.77

Table 10
Comparing the single depot solutions to the MTA solution

Solution	Vehicle cost (min)	Operating cost (min)	Total cost (min)	# of vehicles used	Saving over the solution of MTA (%)		
					Vehicle cost	Operating cost	Total cost
MTA	186000	356243	542243	620			
Proposed strategy	174300	344865	519165	581	6.29	3.19	4.26

also solved the MTA single depot problems with the exact trip sets as in the MTA problem. In other words, we solved the problem using our approach for individual MTA depots based on the MTA trip allocations for those depots. As was expected, the solution times for these experiments were much less than the time required for the multiple depot case. The results indicate that the single depot approach also provides significant savings in all respects over the MTA solutions. Table 10 shows a summary of the results.

10. Conclusion, and future research directions

New formulations for the MDVS and MDVSRTC problems were presented. An exact solution procedure and two heuristic procedures for solving the MDVSRTC problem were proposed. Two methods were proposed to decrease the size of very large problems. These techniques can be applied to the same problem in a sequence.

The MTA operation was analyzed and the required data for constructing a case study was extracted from the existing MTA data. Any missing data were generated using other sources. A solution strategy was used to solve the real-world MDVSRTC problem of the MTA. The solutions obtained using this strategy were presented and compared with the MTA operation statistics. The existing solution of the problem was improved by 5.77%.

From the results on the test problems and especially from the results achieved by solving the MTA operation problem, we can conclude that the model, the heuristic procedures, and the reduction techniques are quite successful in solving the MDVSRTC problems for medium to large cities.

The existing MTA operation is based on the single depot vehicle scheduling analysis, and the pre-assigning of the lines to the depots similar to most transit agencies. In most agencies drivers are used to operate on specific routes. Multiple depot scheduling may result in construction of blocks containing trips from several lines and is more efficient than the single depot scheduling, but there are some costs associated with issues such as driver training that should be compared to the savings achieved by MDVS.

The number of route time constraints can be very high and all of these constraints cannot be generated and added to the problem. As our experience on the test problems and also on the real-world problem shows, during a scheduling process, the number of block violations that result are not large. We can conclude that the constraint generation approach that builds the constraints only if they are needed, is a very practical approach to solving the MDVSRTC problem.

A long-term research opportunity is to look into the modeling of other steps in the transit planning process. One of the more important steps in this process is crew scheduling or run cutting. Since most of the transit operational costs are attributed to labor, efficient run cutting can result in significant cost saving for transit agencies.

The new formulation, the heuristic solution procedures, the techniques for decreasing the size of the MDVSRTC problems, and the strategies for solving the large problems can be used as tools to develop a decision support system to help solve real-world MDVSRTC problems. The adjustment factors built into these procedures let us adjust any problem to fit the existing computer capabilities. These factors make these procedures useful not only for today but also for the future, as long as the new formulation can be used to describe the problem.

References

Ball, M., Bodin, L., 1983. A matching based heuristic for scheduling mass transit crews and vehicles. Transportation Science 17, 4–31.

Banihashemi, M., 1998. Multiple depot transit scheduling problem considering time restriction constraints. Ph.D. dissertation, Department of Civil Engineering, University of Maryland, College Park, MD.

Bertossi, A.A., Carraresi, P., Gallo, G., 1987. On some matching problems arising in vehicle scheduling models. Networks 17, 271–281.

Bodin, L., Golden, B., Assad, A., Ball, M., 1983. Routing and scheduling of vehicles and crews the state of the art. Computer and Operations Research 10, 63–211.

Branco, I.M., 1989. Algorithms para modelos matemtio de quasi-afectacao e extenses. Ph.D. thesis, DEIOC, FCUL, Universidade de Lisboa, Portugal.

Carraresi, P., Gallo, G., 1984. Network models for vehicle and crew scheduling. European Journal of Operations Research 16, 139–151.

Forbes, M.A., Hotts, J.N., Watts, A.M., 1994. An exact algorithm for multiple depot vehicle scheduling. European Journal of Operational Research 72, 115–124.

Freling, R., Paixão, J., 1993. Vehicle scheduling with time constraints. In: Proceedings of the Sixth International Workshop on Computer-aided Scheduling of Public Transit, Lisbon, Portugal.

Gavish, B., Shifler, E., 1978. An approach for solving a class of transportation scheduling problems. European Journal of Operations Research 3, 12–134.

Lamatsch, A., 1990. An approach to vehicle scheduling with depot capacity constraints. In: Proceedings of the Fifth International Workshop on Computer-aided Scheduling of Public Transit, Montreal, Canada.

Löbel, A., 1997. Solving large-scale multiple-depot scheduling problems. In: Proceedings of the Seventh International Conference on Computer-aided Scheduling of Public Transit, Cambridge, MA.

Mesquita, M., Paixão, J., 1990. Multiple depot vehicle scheduling problem: a new heuristic based on quasi-assignment algorithm. In: Proceedings of the Fifth International Workshop on Computer-aided Scheduling of Public Transit, Montreal, Canada.

Mesquita, M., Paixão, J., 1997. Exact algorithms for the multiple-depot vehicle scheduling problem based on multicommodity network flow type formulations. In: Proceedings of the Seventh International Conference on Computer-aided Scheduling of Public Transit, Cambridge, MA.

Ribeiro, C.C., Soumis, F., 1994. A column generation approach to the multiple-depot vehicle scheduling problem. Operations Research 42, 41–52.