Transit Vehicle-Type Scheduling Problem

AVISHAI CEDER

This work addressed the problem of how to allocate vehicles efficiently for carrying out all of the trips in a given transit timetable, in which each vehicle is assigned a chain of trips, some of which may be deadhead (empty) trips. The methodology presented takes into account the association between the characteristics of each trip (urban, peripheral, intercity, etc.) and its required vehicle type. The problem is based on given sets of trips and vehicle types, in which the categories are arranged in decreasing order of vehicle cost. Therefore, each trip can be carried out by its vehicle type, or by other types listed in priority order. This problem can be formularized as a cost-flow network problem with a nonpolynomial-hard complexity level. Thus, a heuristic algorithm was developed in this work on the basis of the deficit function theory. A real-life example is presented at the end of the paper to assess the methodology's effectiveness.

The transit scheduling process in general comprises three main components. First, the timetables are established based on passenger demand. Second, chains of trips are created; each is associated with one vehicle. Third, the driver duties are constructed (crew scheduling), based on various constraints and work rules. The component dealt with in this study is how to allocate vehicles efficiently for carrying out all of the trips in a given timetable, in which each vehicle is assigned a chain of trips, some of which may be deadhead (DH) (empty) trips.

In the vehicle scheduling component, the major objective is to minimize the cost of the vehicle assignment task. This usually coincides with minimizing the number of vehicles required to carry out all of the timetables, provided that the cost of the DH trips is less than the cost involved in employing additional vehicles. Several procedures and techniques are reported in the literature for handling the vehicle scheduling component efficiently (not entirely optimally because of the complexity of the problem). Most of the reported studies, as well as the procedures commonly carried out by practitioners, consider one type of transit vehicle. However, because of recently proposed and implemented deregulation and privatization strategies, more than one type of vehicle is being used in operation (e.g., minibuses, articulated and double-decker buses, and standard buses with varying degrees of comfort and different number of seats).

It is the purpose of this study to address the vehicle scheduling problem while taking into account the association between the characteristics of each trip (urban, peripheral, intercity, etc.) and its required vehicle type. This is to comply with a certain level of service required by each trip in terms of comfort, seat availability, and other operational features.

This work comprises seven sections. Section 1 is the introduction. Section 2 provides the background on the deficit function methodology that is used extensively in the proposed solution. Section 3 interprets the problem as an optimization problem. Section 4 provides the various definitions used. Section 5 describes the algorithm developed. Section 6 presents a real-life example, and Section 7 contains some concluding remarks.

Civil Engineering Department, Technion-Israel Institute of Technology, Transportation Research Institute, Haifa 32000, Israel.

BACKGROUND OF DEFICIT FUNCTION

Following is a description of the deficit function approach for assigning the minimum number of vehicles to allocate for a given timetable. A deficit function is simply a step function that increases by one at the time of each trip departure and decreases by one at the time of each trip arrival. Such a function may be constructed for each terminal in a multiterminal bus system. To construct a set of deficit functions, the only information needed is a timetable of required trips. The main advantage of the deficit function is its visual nature. Let $d(k, t, S)$ denote the deficit function for terminal k at time t for the schedule S. The value of $d(k, t, S)$ represents the total number of departures minus the total number of trip arrivals at terminal k, up to and including time t. The maximal value of $d(k, t, S)$ over the schedule horizon $[T_1, T_2]$ is designated $D(k, S)$.

Let t_s^i and t_e^i denote the start and end times of trip i, $i \epsilon S$. It is possible to partition the schedule horizon of $d(k, t, S)$ into a sequence of alternating hollow and maximal intervals. The maximal intervals $[s_i^k, e_i^k]$, $i = 1, \ldots n(k)$ define the interval of time over which $d(k,t)$ takes on its maximum value. Note that the S will be deleted when it is clear which underlying schedule is being considered. Index i represents the ith maximal intervals from the left, and $n(k)$ represents the total number of maximal intervals in $d(k, t)$. A hollow interval is defined as the interval between two maximal intervals. Hollows may consist of only one point, and if this case is not on the schedule horizon boundaries (T_1 or T_2), the graphical representation of $d(k, t)$ is emphasized by a clear dot.

If the set of all terminals is denoted as T, the sum of $D(k)$ $\forall$ $k \epsilon T$ is equal to the minimum number of vehicles required to service the set T. This is known as the fleet size formula. Mathematically, for a given fixed schedule S:

$$N = \sum_{k \epsilon T} d(k) = \sum_{k \epsilon T} \max_{t \epsilon [T_1, T_2]} d(k, t) \tag{1}$$

where N is the minimum number of buses to service the set T.

When DH trips are allowed, the fleet size may be reduced below the level described in Equation 1. Ceder and Stern (*1*) describe a procedure based on the construction of a unit reduction DH chain (URDHC), which, when inserted into the schedule, allows a unit reduction in the fleet size. The procedure continues inserting URDHCs until no more can be included, or a lower boundary on the minimum fleet is reached. The lower boundary is determined from the overall deficit function defined as $g(t, S)_{k \epsilon T} = \Sigma d(k, t, S)$. This function represents the number of trips simultaneously in operation.

Initially, the lower bound was determined to be the maximum number of trips in a given timetable that are in simultaneous operation over the schedule horizon. Stern and Ceder (*2*) improved this lower bound, based on the construction of a temporary timetable in which each trip's arrival time is extended to the time of the first trip that may feasibly follow it in S.

The deficit function theory was extended by Ceder and Stern (*3*) to include possible shifting in departure times within bounded tolerances. Basically, the shifting criteria is based on a defined tolerance time $[t_s^i - \Delta_a^i, t_s^i + \Delta_d^i]$, where Δ_a^i is the maximum advance of the trip scheduled departure time (early departure), and Δ_d^i is the maximum delay allowed (late departure). The maximum interval is then compared with the appropriate tolerance time elements for establishing conditions in which it is possible to reduce the fleet size by one via certain shifts.

The algorithms of the deficit function theory are described in detail by Ceder and Stern (*1,3*). However, it is worth mentioning the next terminal (NT) selection rule and the URDHC routines. The selection of the NT in attempting to reduce its maximal deficit function may rely on the basis of garage capacity violation, or on a terminal whose first hollow is the longest. The rationale here is to try to open up the greatest opportunity for the insertion of the DH trip.

Once a terminal k is selected, the algorithm searches to reduce $D(k)$ by shifting departure times (if allowed). Then all of the $d(k, t)$ values are updated and the NT rule is again applied. When no more shiftings are possible, the algorithm searches for a URDHC from the selected terminal while considering possible blending between DH insertion and shiftings in departure times. In the URDHC routines there are four rules: $R = 0$ for inserting the DH trip manually in a conversational mode, $R = 1$ for inserting the candidate DH trip that has the minimum travel time, R = 2 for inserting a candidate DH trip whose hollow starts farthest to the right, and $R = 3$ for inserting a candidate DH trip whose hollow ends farthest to the right. In the automatic mode ($R = 1, 2, 3$), if a DH trip cannot be inserted and the completion of a URDHC is blocked, the algorithm backs up to a DH candidate list and selects the next DH candidate on that list.

In the fixed schedule problem, the algorithm also terminates when the improved lower bound (*3*) is equal to $D(S)$. In the variable schedule problem (when shiftings are allowed), the algorithm also uses this comparison, and if the improved lower bound is equal to $D(S)$, the URDHC procedure (with shiftings) ceases and the shifting-only mode is applied. If the latter results in reducing $D(S)$, the URDHC procedure is again activated. The process terminates when $D(S)$ cannot be further reduced.

Finally, all of the trips, including those that were shifted and the DH trips, are chained together for constructing the vehicle schedules (blocks). Two rules can be applied for creating the chains: first in-first out (FIFO), and a chain-extraction procedure described by Gertsbach and Gurevich (*4*). The FIFO rule simply links the arrival time of a trip to the nearest departure time of another trip (at the same location), and continues to create a schedule until no connection can be made. The trips considered are deleted and the process continues. The chain-extraction procedure allows an arrival-departure connection for any pair within a given hollow (on each deficit function). The pairs considered are deleted and the procedure continues. Both methods end with the minimum derived number of vehicles (chains).

OPTIMIZATION FRAMEWORK

The problem is based on given sets S of trips and M of vehicle types. The set M is arranged in decreasing order of vehicle cost so that if $u \epsilon M$ is listed above $v \epsilon M$, it means that $c_u > c_v$, where c_u, c_v are the costs involved in employing vehicle of type m and n, respectively. Each trip $i \epsilon S$ can be carried out by vehicle type $u \epsilon M$ or by other types listed before u in the above-mentioned order of M.

The problem can be formularized as a cost-flow network problem, in which each trip is a node, and there is an arc connecting between two trips if, and only if, it is possible to link them in time sequence with and without DH connections. On each arc (i, j), there is a capacity of one unit and an assigned cost C_{ij}. If the cost of the lower-level vehicle type associated with trip i is higher than the cost of the vehicle type (lower level possible) required for trip j, then $C_{ij} = c_i$. That is, $C_{ij} = max\ (c_i, c_j)$. The use of such formulation was implemented by Costa et al. (*5*), while employing three categories of solutions: (a) a multicommodity network flow, (b) a multidepot vehicle scheduling problem, and (c) a set partitioning problem with side constraints. The mixed-integer programming of these problems is known as nonpolynomial-hard (NP-hard): as, for example, in Bertossi et al. (*6*).

Because of the complexity involved in reaching an optimal solution for a large number of elements (trips) in S, a heuristic method has been realized as a more practical approach. This article develops a heuristic procedure based on the deficit function theory for transit vehicle scheduling.

The heuristic algorithm developed in this article is titled the vehicle-type scheduling problem (VTSP) algorithm. It begins by establishing lower and upper bounds on the fleet size. The upper bound is attained by creating different deficit functions, each associated with a certain vehicle type $u \epsilon M$, in which it includes only the trips whose lower-level required vehicle type is u. Certainly, this scheduling solution reflects high cost, caused by the large number of vehicles demanded. The lower bound on the fleet size is attained by using only one vehicle type: the most luxurious one with the highest cost that can clearly carry out any trip in the timetables. In addition, for the lower-bound case, the cost required is high. Between these bounds on fleet size, the procedure searches for the best solution based on the properties and characteristics of the deficit function theory.

This optimization framework is presented in Figure 1, with (C_1, N_1) and (C_2, N_2) representing the lower- and upper-boundary solutions, respectively. Following are the definitions of the VSTP algorithm that are based mainly on the definitions of the deficit function theory.

DEFINITIONS

S = the set of required trips in the fixed trip schedule;

T = the set of all terminals (or start and end points) in the trip schedule;

M = the set of all vehicle types;

$\{'kq('kq)\}$ = the matrix of DH trip times from terminal k to q and its associated possible service trip times (in parentheses);

$[T_1, T_2]$ = the span of the schedule horizon;

$d(k, t, S)$ = the deficit function for terminal k, at time t, for the schedule S;

$D(k, S) = \max_t d(k, t, S)$ = the maximum deficit over all $t \epsilon [T_1, T_2]$ at terminal k, for schedule S;

$D(S) = \Sigma_{k \epsilon T} D(k, S)$ = the total deficit;

$[s_i^k, e_i^k]$ = start and end time, respectively, of the ith max interval of the deficit function for terminal k, i = 1, 2, . . . $n(k)$;

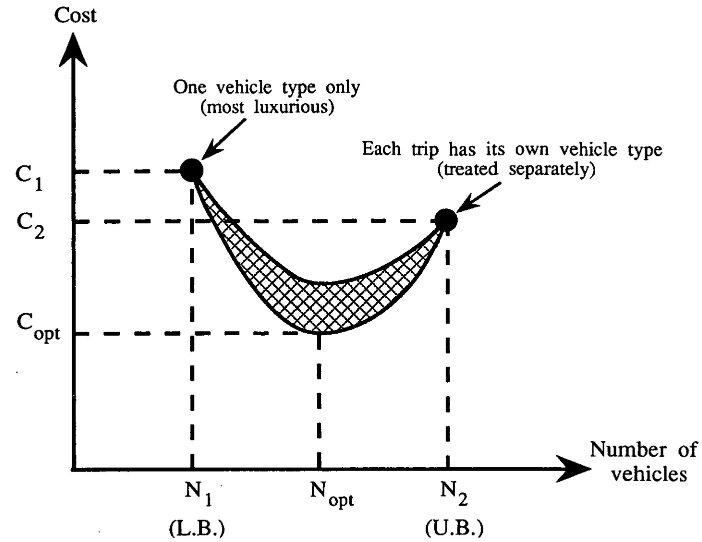

FIGURE 1 Trade-off situation between total cost and total number of vehicles.

$[e_i^k, s_i^k + 1]$ = end and start times, respectively, of the ith hollow of the deficit function for terminal k, $i = 0, 1, 2, \ldots n(k)$;

URDHC = abbreviation for unit reduction deadhead chain; that is, a chain of DH trips that reduces by one $D(k, s)$ for a given $k \in T$;

t_{su}^i = start time of trip $i \in S$ of type $u \in M$;

t_{eu}^i = end time of trip $i \in S$ of type $u \in M$;

Δ_{du}^i = maximum allowed delay from the scheduled departure time of trip $i \in S$ of type $u \in M$;

Δ_{au}^i = maximum allowed advance of the trip scheduled departure time of trip $i \in S$ of type $u \in M$;

$d_u(k, t, S)$, $D_u(k, S)$, $D_u(S)$, $[s_{iu}^k, e_{iu}^k]$, $[e_{iu}^k, s_{(i+1)u}]$ refer to these definitions with respect only to trips of type $u \in M$;

C_u = the cost involved in employing vehicle of type $u \in M$;

N_1 = the minimum number of vehicles required to service all trips in S for a single vehicle type;

N_{2u} = the minimum number of vehicles required to service all trips of type $u \in M$ in S;

$N_2 = \sum_{u=1} N_{2u}$ = the sum of all minimum numbers of vehicles required when treating each type separately;

$C_1 = N_1 c_1$ = the total cost involved in employing N_1 vehicles of type 1 (most luxurious);

$C_2 = \sum_{u=1} N_{2u} c_u$ = the total cost involved in employing N_2 vehicles (for each type separately);

C = the objective function of minimal total cost required to service all of the trips in S while complying with the vehicle type constraints.

VTSP ALGORITHM

The VTSP algorithm developed is heuristic in nature while incorporating all of the components of the deficit function methodology. Because of the graphical features associated with the deficit function theory, the algorithm can be applied in an interactive manner or in an automatic mode, along with the possibility to examine its intermediate steps.

The following is a general description of the VTSP algorithm in a stepwise manner.

Step 0 Arrange the set of vehicle types M in decreasing order of vehicle cost (so that if $m \in M$ is listed above $n \in M$, it means $c_m > c_n$).

Step 1 Solve the problem as a single vehicle type problem using the deficit function theory, including the DH and shifting procedures (*1,2,3*), to obtain N_1 vehicles considered as type 1 with a total cost of C_1, where $C_1 = N_1 c_1$.

Step 2 Partition the trips by their associated type and apply the deficit function methodology with the DH and shifting procedures (*1,2,3*), for each type separately. Sum the number of vehicles derived to obtain a total of N_2 vehicles with a total cost, where $C_2 = \sum_{u=1} N_{2u} c_u$ and $N_2 = \sum_{u=1} N_{2u}$.

Step 3 If $N_1 = N_2$, STOP. Use the solution of Step 2.

Step 4 Consider $d_u(k, t)$ as in Step 2 for all $k \in T$ and $u \in M$.

Step 5 Perform the shifting-only procedures for shifting departure times within their tolerances (*3*).

Step 6 Find a URDHC (*1,2,3*), such that a DH trip (with possible shiftings) can link between trip i of type u and trip j of type v if and only if one or more of the following conditions are fulfilled: (a) $u < V$; (b) the URDHC aims at saving a vehicle of type w and $w < v$; or (c) the URDHC aims at saving a vehicle of type w and $c_v - (c_w + c_u) < 0$. If no URDHC can be found, stop.

Step 7 Examine whether the total cost of the URDHC (DH cost) is less than the cost of saving one vehicle (of the type considered). If not, delete this possibility and go to Step 6. Otherwise, update $d_u(k,t)$ for all $k \in T$ and $u \in M$.

Step 8 Apply the improved lower-bound check (*2*). If it equals to $D(S)$, go to Step 5.

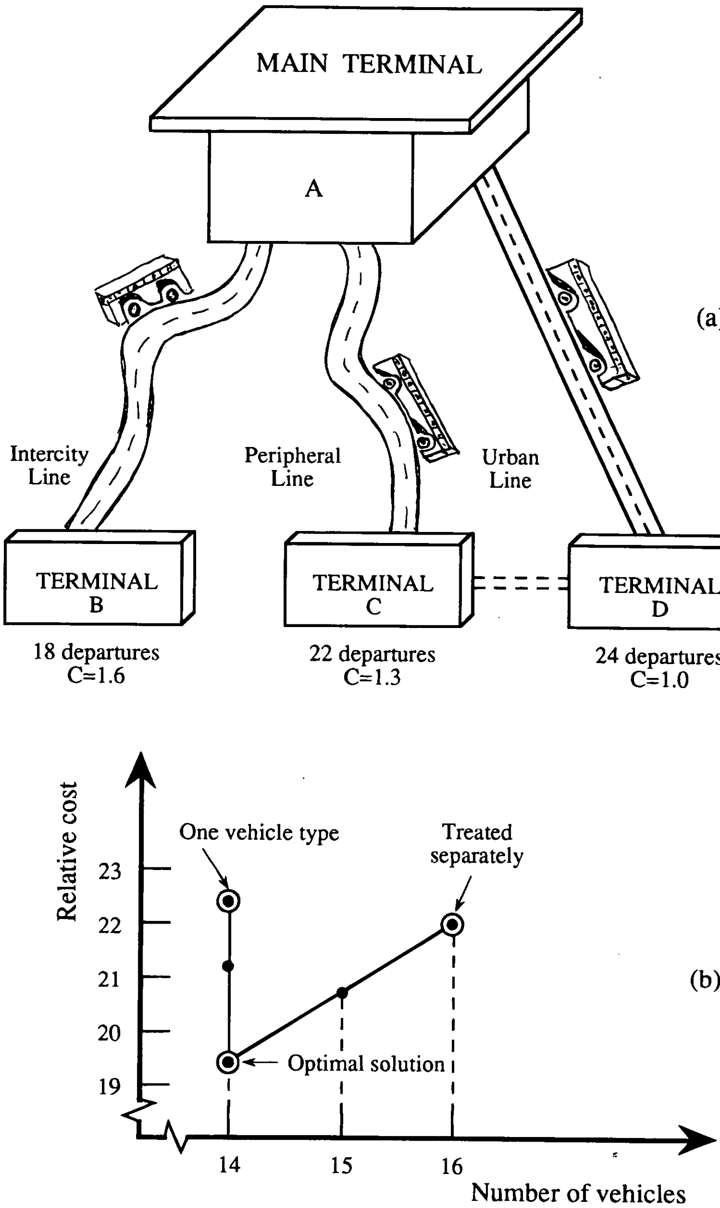

FIGURE 2 **(*a*) Schematic description of real-life scheduling problem with three types of vehicles and relative costs; (*b*) Trade-off situation between total number of vehicles and cost associated with five disparate solutions.**

Among the eight steps of the VTSP algorithm, Step 3 and particularly Step 6 deserve further attention concerning the conditions specified in these steps. The following four propositions clarify and interpret these conditions.

Proposition 1 (for Step 3) If $N_1 = N_2$, then $C_2 < C_1$.

Proof: Given $M > 2$ and $c_1 > c_2 \ldots > c_m$, the proof is straightforward because $C_1 = N_1c_1$. $C_2 = \sum_{u=1} N_{2u}c_u$, $N_2 = \sum_{u=1} N_{2u}$, and $N_1 = N_2$.

Proposition 2 [for Step 6(*a*)] Any DH trip connection between an arrival of a trip of type u with a departure of a trip of type v such that $u < v$ within any URDHC does not increase C.

Proof: Because $c_u > c_v$, this DH trip connection cannot lead to an upgrade of the vehicle type, thus cannot increase the objection function C.

Proposition 3 [for Step 6(*b*)] Any DH trip connection between an arrival of a trip of type u with a departure of a trip of type v, such that $u > v$ within a URDHC aims at saving a vehicle of type w, such that $w < v$ does not increase C.

Proof: This DH trip connection may upgrade the vehicle type of a trip of type u (from u to v). That is, the result may be a saving of a vehicle of type w along with an upgrade of one vehicle from type u to v. As $c_w > c_v$ and $c_v > c_u$, then the net saving is always negative: $-c_w + (c_v - c_u) < 0$; thus, in any of these instances, C can only decrease.

Proposition 4 (for Step 6(*c*)) Any DH trip connection between an arrival of a trip of type u and a departure of a trip of type v, such that $u > v$ within a URDHC aims at saving a vehicle of type w, such that $w > v$ does not increase C if $c_v - (c_w + c_u) < 0$.

Proof: In this case $c_v > c_w$; hence Proposition 3 cannot be applied here. Therefore, the condition for a negative net saving is set to $-c_w + (c_v - c_u) = c_v - (c_w + c_u) > 0, w, u, v, \epsilon M$.

A REAL-LIFE EXAMPLE

The VTSP algorithm was used to examine a real-life scheduling problem. The EGGED bus company in Israel was selected, which has three different bus lines departing from a main terminal in Haifa. In Figure 2*a*, the three lines are shown schematically: intercity, peripheral, and urban. The intercity line is characterized by 18 departures in its daily timetable, with 120 min average travel time and 105 min average DH time between *A* and *B*. The peripheral line has 22 daily departures, with 45 min average travel time between *A* and *C*, average DH time of 24 min between *A* and *C*, and 36 min between *C* and *D*. The urban line has 24 daily departures, with 30 min average travel time and 15 min average DH time between *A* and *D*. The relative costs of the intercity, peripheral, and urban vehicles are 1.6, 1.3, and 1.0, respectively. The allowed shiftings $\Delta^i du = \Delta^i au = 3$ min for all trips i and vehicle type u.

The VTSP algorithm results in the optimal solution shown in Figure 2*b* with $C = 19.4$, and 14 vehicles are required: 7, 4, and 3, intercity, peripheral, and urban vehicles, respectively. Steps 1 and 2 of the algorithm result in $C = 22.4$ (14 intercity vehicles) and $C = 22$ (7 intercity, 6 peripheral, and 3 urban vehicles), respectively. This outcome of the algorithm is circled in Figure 2*b*. In addition, this figure contains two more solutions, with $C = 21.2$ (11 intercity, 2 peripheral, and 1 urban vehicles) and $C = 20.7$ (7 intercity, 5 peripheral, and 3 urban vehicles). These solutions are based on the deficit function's shifting and URDHC procedures, excluding the three conditions of Step 6 of the VTSP algorithm.

CONCLUDING REMARKS

The results of the heuristic method suggest that the VTSP algorithm also can be used for large transit agencies, ensuring efficient allocation of different vehicles to trips while reducing costs involved to a minimum level. It is worth mentioning that for further understanding of the algorithm presented, a detailed example is presented by Ceder (*7*).

Because the algorithm is based on the deficit function theory, it is recommended that the various rules contained in this theory be applied. That is, the algorithm may commence searching for the optimal solution based on different criteria: (a) shifting (departure times) first DH (trip insertion) after; (b) only DH insertions without shifting, and (c) DH with shifting simultaneously. It is worth mentioning that one of the main advantages of the deficit function is its visual nature. Consequently, one can observe, even in an automatic mode, intermediate results and evaluate them while the algorithm executes further procedures. The inevitable interaction between the setting timetable, vehicle scheduling, and crew assignment components emphasizes the importance of allowing the scheduler to understand the solution process and be able to interfere whenever he thinks it is justifiable.

REFERENCES

1. Ceder, A., and H. I. Stern. Deficit Function Bus Scheduling with Deadheading Trip Insertions for Fleet Size Reduction. *Transportation Science*, Vol. 15, 1981, pp. 338–363.
2. Stern, H. I., and A. Ceder. An Improved Lower Bound to the Minimum Fleet Size Problem. *Transportation Science*, Vol. 17, 1983.
3. Ceder, A., and H. I. Stern. Graphical Person-Machine Interactive Approach for Bus Scheduling. In *Transportation Research Record 857*, TRB, National Research Council, Washington, D.C., 1982, pp. 69–72.
4. Gertsbach, I., and Y. Gurevich. Constructing an Optimal Fleet for a Transportation Schedule. *Transportation Science*, Vol. 11, 1977, pp. 20–36.
5. Costa, A., I. Branco, and J. Paixao. Vehicle Scheduling Problem with Multiple Types of Vehicles. In *International Workshop on Computer-Aided Scheduling of Public Transport*, Springer-Verlag, Lisbon, July 1993.
6. Bertossi, A., P. Carraresi, and G. Gallo. On Some Matching Problems Arising in Vehicle Scheduling. *Networks*, Vol. 17, 1987, pp. 271–281.
7. Ceder, A. *Minimum Cost Vehicle Scheduling with Different Types of Transit Vehicles.* In *International Workshop on Computer-Aided Scheduling of Public Transport*, Springer-Verlag, Lisbon, July 1993.

Publication of this paper sponsored by Committee on Bus Transit Systems.