PERGAMON

Transportation Research Part A 36 (2002) 453–482

TRANSPORTATION RESEARCH PART A

www.elsevier.com/locate/tra

Bus maintenance systems and maintenance scheduling: model formulations and solutions

Ali Haghani [a,*], Yousef Shafahi [b]

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

[b] *Sharif University of Technology, Tehran, Iran*

Received 3 August 1998; received in revised form 24 January 2001; accepted 28 February 2001

Abstract

This paper deals with the problem of scheduling bus maintenance activities. The scheduling of maintenance activities is an important component in bus transit operations planning process. The other components include network route design, setting timetables, scheduling vehicles, and assignment of drivers. This paper presents a mathematical programming approach to the problem. This approach takes as input a given daily operating schedule for all buses assigned to a depot along with available maintenance resources. It, then, attempts to design daily inspection and maintenance schedules for the buses that are due for inspection so as to minimize the interruptions in the daily bus operating schedule, and maximize the utilization of the maintenance facilities. Three integer programming formulations are presented and different properties of the problem are discussed. Several heuristic methods are presented and tested. Some of these procedures produce very close to optimal solutions very efficiently. In some cases, the computational times required to obtain these solutions are less than 1% of the computational time required for the conventional branch and bound algorithm. Several small examples are offered and the computational results of solving the problem for an actual, 181-bus transit property are reported. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Bus transit; Maintenance; Scheduling; Optimization model; Heuristic algorithm

1. Introduction

The transit operational planning process includes four basic components that are performed in sequence: (1) network route design, (2) setting timetables, (3) scheduling vehicles to trips, and

* Corresponding author. Tel.: +1-301-405-1963; fax: +1-301-405-2585.

E-mail address: haghani@eng.umd.edu (A. Haghani).

PII: S0965-8564(01)00014-3

(4) assignment of drivers. It is desirable for all components to be planned simultaneously in order to optimize productivity and efficiency. However, due to the complexity of the problem, transit planning is performed in a sequential manner in which the outcome of one step is fed as an input to the next step. Fleet maintenance is another major component in bus transit operations and is one of the major problems with which the public transit agencies are faced.

The reliability, safety, and vehicle life in bus transit systems depend on the maintenance system employed by the agencies. Maintenance costs are the second highest expense category after vehicle operations (Bladikas and Papadimitriou, 1986). They account for approximately 21% of the total operating expenses in a typical transit system (Purdy and Wiegmann, 1987). These facts make maintenance one of the most important and substantial elements of transit systems. Considering that maintenance costs have increased 33% faster than vehicle operating costs, and four times faster than general/administrative costs in recent years (Purdy and Wiegmann, 1987), providers of public transportation are being challenged by high costs, dwindling sources of support funds, and pressures to improve services. Therefore, transit agencies make capital investments in maintenance systems with the objective of having a reliable, safe and attractive transit system.

There are numerous papers, articles, and scientific reports that document the results of studies in the bus maintenance area. Some of these studies try to identify different maintenance systems and their components, some focus on maintenance cost characteristics under different conditions, some attempt to offer maintenance performance indicators, and some evaluate different policies and strategies in maintenance systems. These studies can help maintenance managers in planning, managing, and controlling different aspects of the maintenance system.

In most bus transit agencies' operations, buses are not assigned a fixed daily service schedule. To meet the daily route service requirements, buses are assigned to a scheduled route service on a daily basis. This is mainly due to parking space limitations in bus depots which force the depot managers to park the buses, one after the other, in long lines that are sometimes as long as 14 or more buses. In such situations, a bus that is parked in the middle of a line, cannot move until all the buses that are parked in front of it are moved. Since the buses generally arrive randomly, they are parked in the same order as they arrive in the garage. Therefore, there is no guarantee that the parking pattern is the same everyday. Under these circumstances, the buses cannot have a fixed schedule. To meet the daily service schedule, the dispatchers decide which bus must serve a particular line on a daily basis based on the day's parking pattern. In such an operation, scheduling maintenance services is very simple. The entire fleet can be basically viewed as a pool of buses from which buses can be drawn either for daily service or for maintenance.

In most agencies, the routine inspections and maintenance are performed in fixed time or mileage intervals. Buses go on a list as the deadline for their next inspection or routine maintenance approaches, and the list is processed generally in order of the due dates and crew availability for the inspection or maintenance that is needed. When buses have unexpected mechanical anomalies and breakdowns, they are pulled out of service, and depending on the type of repair needed, they are either sent to a major repair facility, or are scheduled for repair along with the buses that need routine inspection and maintenance. Since in most common operations the buses do not have a fixed schedule, scheduling maintenance activities is a trivial problem.

Some agencies have fleet management systems in place that allow Division managers to assign buses to specific schedules, and maintain this assignment for a long period of time. The main rationale for devising such systems is that by assigning a fixed schedule for a bus, the operators drive the same bus everyday and will be more sensitive to mechanical anomalies of the bus and report minor problems before they turn into major ones. When such systems are in place, the transit managers desire that the operators drive the same bus to the extent possible. When the buses are due for routine inspections, it is desirable to perform the inspections during the times that the buses are parked idle in the garage, rather than pulling them out of service and having to dispatch substitute buses to perform the scheduled services.

In this computerized environment, since the buses have fixed schedules, efficient scheduling of maintenance activities around the fixed service schedule of buses is no longer a trivial problem and requires a computerized maintenance scheduling component to minimize the interruption of scheduled service for each bus and maximize the utilization of the maintenance facility. This is a new problem that agencies such as MTA are faced with, and its solution is the underlying motivation for this research. It should be noted that this research has been conducted in close cooperation with the MTA, and all information and data used in this research have been provided by the MTA.

Given the buses' operating schedules, their maintenance and inspection requirements, and maintenance resource and crew availability, the question is how can a maintenance manager schedule buses for maintenance and assign them to limited existing facilities such that each bus is maintained in time, while the amount of time that the buses are pulled out of their scheduled services for maintenance is minimized. Obviously, the main objective of such a scheduling system is to maximize the reliability of the system and efficiently utilize maintenance crew and facility time.

This problem has not been addressed in the literature. Review of the literature related to maintenance systems in other transportation modes also indicates that none of the available models are directly transferable to a bus transit system. Consequently, this is an area of research that has a potential for many fruitful contributions. This research attempts to deal with this issue, and in particular, analyze the problem, present a mathematical formulation, and an efficient solution procedure.

In the rest of this paper we propose three mathematical models and several solution techniques to solve this problem. The results of the application of the proposed model are shown through several small examples and a large scale case study.

2. Problem statement and model formulation

2.1. Problem statement

Andrle (1986) summarized 11 areas in which research in transit bus maintenance would be beneficial. These suggested areas for research included demonstration of the importance of maintenance, quality of current maintenance, information systems, measure and standards, maintenance contracting, personnel incentives, training, programmed component replacement versus fix on failure, computerized smart systems, appropriate scale of maintenance management systems, and information sharing. Over the past few years, some of these topics have been considered by other researchers and there are a few reports about the results and methods (Maze and Cook, 1987). The results of these research efforts are useful for understanding the maintenance problem. Based on these studies the manager of a transit bus maintenance system can conceivably find appropriate equipment, the best operational policy and so on. The literature in bus maintenance scheduling, however, is extremely sparse and virtually non-existent. The main reason for this lack of attention is the current status of the bus operations in many agencies that make the scheduling problem very simple.

We can essentially divide the bus maintenance problems into three categories: daily inspection, preventive (periodic) maintenance, and emergency maintenance. Most bus transit agencies do not have dedicated facilities for daily inspections. This type of inspection is defined as a general check by the operators or other personnel at the beginning or at the end of the daily run, so in this case no scheduling is needed.

Emergency maintenance is required when buses have unexpected breakdowns. There is no way of knowing in advance whether or not a bus will break down on a particular day. Breakdown of a bus during service is the most costly and disruptive situation that results in a road call, and requires that the maintenance manager dispatches a mechanic to the site to determine the cause and the nature of the breakdown, and fix it if possible. This is a significant drain on the maintenance resources because a job in progress must be put on-hold to dispatch a mechanic to the breakdown site. In most cases, the bus is towed back to the facility. Depending on the nature of the breakdown, it is either fixed at the site, or the bus is sent to a central maintenance depot for major repair work. When the bus is to be fixed at the facility, it can be considered as yet another bus that requires maintenance, and this maintenance activity can be scheduled with the other regular maintenance activities. If the bus is sent to another facility, there is no need for it to be considered in scheduling.

Preventive maintenance is the regular inspection of buses in pre-specified mileage or time intervals. During these inspections, certain parts of buses are replaced. If other components are identified that are deemed near failure, they are replaced too. This type of maintenance is the bulk of the maintenance work at the local facilities. In this paper we will focus on the preventive maintenance scheduling which is the most important one from the point of view of programming and scheduling. We assume that we always have a fixed number of maintenance lines or bays, at any given time and we want to schedule the daily preventive maintenance activities for these bays.

2.2. Mathematical formulation of the preventive maintenance scheduling problem

There are several types of preventive maintenance undertaken by an agency depending on its maintenance policies. Generally it can be assumed that each bus transit agency has a certain number of inspection types, and a policy based on mileage and/or time for each type of inspection. In other words, each of the inspections must be performed on each bus periodically based on a pre-specified time or mileage interval.

The time or the mileage interval does not necessarily have to be an exact number. These intervals can be defined as a time or mileage window for each type of inspection for each bus. In other words, a given type of inspection should be performed on a bus within a pre-specified time or mileage window after the last time when a similar inspection was performed on this bus. For example, an agency may have a policy to perform brake inspection every 3000 miles. In practice, this policy may be enforced by allowing future brake inspections within 2800–3200 miles of service since the previous brake inspection. Similar policies can be instituted for inspections that are generally based on time rather that mileage (such as wheelchair lift equipment inspection).

We can safely assume that the agency has accurate information regarding the current mileage of the buses, the date and mileage at which each of the different inspection types was performed on the bus, and the mileage that each bus runs per day. Given this information, we can easily calculate the time and mileage window for each type of inspection for all buses in the fleet. This time window will be extremely accurate if the buses are assigned to particular scheduled services and their actual daily mileage is known. When these time windows are calculated, the time and mileage windows during which the next required inspection type has to be performed on each bus will be known.

Normally every agency has several maintenance lines (bays) in a bus depot. One or more maintenance crews work in each of these maintenance lines who can perform one or more types of maintenance. Sometimes due to physical restrictions certain types of inspections cannot be performed in certain bays. For example, an inspection which requires going under the bus must be performed in a bay which either has a pit or a hoist.

When the buses have assigned service schedules, the ideal situation for the agency is that all buses are inspected during the time when they are idle and parked in the garage and not to have to pull them out of service. The transit agencies are generally interested to schedule their upcoming maintenance activities during a given time period (which may be several days to a week) given the information regarding the bus inspection requirements and crew availability. This time period, which we refer to as the planning period or horizon, can be further broken down into smaller time intervals that represent unit time slots for the maintenance bays. Each type of inspection would require one or more unit time slots in a bay to be completed. The problem, then, is to allocate the maintenance time slots available for all bays, to different maintenance activities so as to minimize service interruption and maximize maintenance resource utilization.

Note that this problem is an operational rather than a tactical problem. The approach to maintenance scheduling in this manner has a short range planning perspective and attempts to use the currently available resources of the agencies to the fullest extent in an optimal manner. It may be determined, as a result of using this approach, that the agency has more maintenance personnel or maintenance lines that are required to accomplish its maintenance objectives and implement its maintenance policies. This would be a by-product of using this approach and in the long run will save the agency from incurring unnecessary cost and spending valuable resources.

The problem as stated above can be formulated as a mathematical optimization problem. We define the following notation:

Notation

B	number of buses
T	duration of the planning period (number of unit maintenance time slots)
M	number of inspection types
ML	total number of maintenance lines (bays)
M_t^m	maximum number of maintenance lines which can be available for maintenance type m at time t
t^m	the time required to perform maintenance type m for each bus
t_1^{bm}	the earliest time slot for starting inspection type m for bus b

t_2^{bm} the latest time slot for starting inspection type m for bus b

S_t^b a binary variable indicating whether or not bus b is scheduled for service in time interval t. $S_t^{b'}$ s can be expressed in a $B'T$ matrix

$$S_t^b \begin{cases} 1 & \text{if bus } b \text{ is busy at time } t \\ 0 & \text{otherwise} \end{cases}$$

$K(t)$ an $ML'M$ matrix for each time period t which indicates whether a particular inspection can be performed in a maintenance bay. The elements of $K(t)$ are defined as follows:

$$K_i^j(t) = \begin{cases} 1 & \text{if inspection type } j \text{ can be accomplished in bay } i \\ 0 & \text{otherwise} \end{cases}$$

M_t^m can be obtained by summation on column m of this matrix

Also, as mentioned before, the current mileage, the average mileage per day for each bus, and the last maintenance dates for all buses and their mileages are known. Therefore, the required earliest and latest time slots for starting a particular inspection on a particular bus, t_1^{bm} and t_2^{bm}, can be computed.

If we define decision variables:

$$Y_t^{bm} = \begin{cases} 1 & \text{if maintenance type } m \text{ is started on bus } b \text{ at time } t, \\ 0 & \text{otherwise} \end{cases}$$

$Z_t^m =$ number of maintenance bays allocated to perform inspection type m during time period t.

The preventive maintenance scheduling problem can be formulated as follows:

$$\text{Maximize} \quad \sum_{b=1}^{B}\sum_{m=1}^{M}\sum_{t=\max(1,t_1^{bm})}^{\min(T,tSUB2^{bm})} w_1 t^m Y_t^{bm} - \sum_{b=1}^{B}\sum_{m=1}^{M}\sum_{t\in s_1}\sum_{k=\max(t_1^{bm},t-t^m+1)}^{t} w_2 Y_k^{bm}, \tag{P-1}$$

$$\text{Subject to} \quad \sum_{t=\max(1,t_1^{bm})}^{\min(T,t_2^{bm})} Y_t^{bm} = 1, \quad b = 1,\ldots,B, \ \ m = 1,\ldots,M, \tag{1}$$

$$\sum_{l\_m}\ \sum_{k=\max(t;t_1^{bl})}^{\min(t+t^m;t_2^{bl};t_2^{bm})} Y_k^{bl} + LY_t^{bm} \leqslant L, \quad b = 1,\ldots,B, \ \ m = 1,\ldots,M,$$

$$t_1^{bm} < t < t_2^{bm}, \tag{2}$$

$$\sum_{b=1}^{B}\sum_{k\in S_2} Y_k^{bm} = Z_t^m \quad t = 1,\ldots,T, \ \ m = 1,\ldots,M, \tag{3}$$

$$Z_t^m \leqslant M_t^m \quad t = 1,\ldots,T, \ \ m = 1,\ldots,M, \tag{4}$$

$$\sum_{m=1}^{M} Z_t^m \leqslant ML, \quad t = 1,\ldots,T, \tag{5}$$

$$Y_t^{bm} = 0,1 \text{ for all } b,m, \text{ and defined } t, \tag{6}$$

$$Z_t^m \geqslant 0 \quad \text{and integer for all } t \text{ and } m, \tag{7}$$

where L is a large number and,

$$S_1 = S_1(b,m) = \{t \,|\, S_t^b = 1 \wedge t_1^{bm} \leqslant t \leqslant t_2^{bm}\}, \tag{8}$$

$$S_2 = S_2(b,m,t) = \{k | \max\left[t_1^{bm}, t - t^m + 1\right] \leqslant k \leqslant \min\left[t, \leqslant t_2^{bm}\right]\}. \tag{9}$$

The objective function is composed of two components. The first is maximizing a weighted total vehicle maintenance hours (maintenance utility) for the buses that are pulled for maintenance during their idle time, and the second is minimizing the weighted total number of maintenance hours for the buses that are pulled out of their scheduled service for inspection. The weights W_1 and W_2 determine the relative importance of the two components. In general, W_1 and W_2 can be different for each bay and each bus, respectively. However, without loss of generality we can assume that W_1 is the same for all bays and W_2 is the same for all buses.

Constraints (1) indicate that type m inspection should be started for each bus during its required time interval for that type of maintenance. Constraints (2) ensure that during the time that a bus is undergoing a particular type of inspection another type of inspection cannot be started. Constraints (3) require that the total number of buses that are undergoing any type of inspection at any time has to be equal to the total number of maintenance lines that are assigned to this type of inspection at that time. Constraints (4) indicate that the number of lines assigned to each type of inspection cannot exceed the total available maintenance lines capable of providing that type of inspection. Constraints (5) require that the total number of maintenance lines assigned to all inspection types has to be less than or equal to the total number of existing maintenance lines.

This formulation is so general that not only it can be used in virtually any bus maintenance system with different characteristics, but also, with minor modifications (if necessary), it can be used in contexts other than just transit bus maintenance. The solution to this problem gives the maintenance schedule for each bus that is due for inspection, as well as the minimum number of maintenance lines that should be allocated for each type of inspection over the scheduled period.

Although we recognize that any real-world size problem formulated as above cannot be solved using conventional methods and off-the-shelf software, for preliminary testing of the formulations, the conventional branch and bound procedure for integer programming is used as a method of solving (P-1). Two C++ programs are developed to set up the problem formulations for LINDO and CPLEX. The inputs to these programs are:

1. Matrix of scheduled service time for buses (matrix S).
2. Maintenance line and inspection type compatibility matrix (matrix K).
3. The matrix that indicates the time window during which a particular type of inspection has to start on each bus.
4. The required time for completing each type of inspection.
5. Other information including the number of buses, the number of inspection types, the number of maintenance lines, and the duration of the planning period.

The outputs of these programs are the IP formulation input files that can be read by LINDO and CPLEX in linear programming modes, respectively. These procedures are shown in examples in the following section.

3. Model characteristics and testing

3.1. Model characteristics

Maintenance operation in a bus transit system is a dynamic operation. Buses are constantly in need of service and inspection as they go through their daily scheduled service and these inspections are periodically repeated. This means that the maintenance manager is constantly scheduling buses for maintenance and revising those schedules based on the most currently available information regarding crew and other resource availabilities. The model formulated in Section 2 (Problem (P-1)) can handle this situation by considering overlapping planning periods. This suggests that if the absolute latest time for a particular inspection for a bus is not within a planning period, then inspection for this bus can be deferred to future planning periods. Problem (P-1) can be revised to account for this by modifying the formulation.

To make the formulation even more efficient, we make a list of buses that require a particular inspection. This list can be compiled for all inspection types. An ordered list of B buses that require different types of inspection may be as follows:

bus number 1 to bus number K_1 need maintenance type 1, $1 \leqslant K_1 \leqslant B$,

bus number $k_1 + 1$ to bus number K_2 need maintenance type 2, $K_1 \leqslant K_2 \leqslant B, \ldots,$ and

bus number $K_{(M-1)} + 1$ to bus number $K_M = B$ need maintenance type M.

A bus that is due for more than one inspection type may appear in several of these lists with different numbers. Using these ordered lists we can further streamline the formulation of the problem. If a particular bus j appears in one of these lists with an index i, we define

$S_i =$ set of all indices of bus j in the other ordered lists.

These modifications result in problem (P-2) which follows:

$$\text{Maximize} \quad \sum_{m=1}^{M} \sum_{b=K_{m-1}+1}^{K_m} \sum_{t=\max(1,t_1^b)}^{\min(T,t_2^b)} \left[w_1 \min(t^m, T-t+1) - \sum_{k=t}^{\min(t+t^m,T)} w_2 s_k^b \right] Y_t^b \tag{P-2}$$

$$\text{Subject to} \quad \sum_{t=\max(1,t_1^{bm})}^{\min(T,t_2^{bm})} Y_t^b = 1 \text{ If } t_2^{bm} \leqslant T, \quad b = 1,\ldots,B, \tag{10}$$

$$\sum_{t=\max(1,t_1^{bm})}^{\min(T,t_2^{bm})} Y_t^b \leqslant 1, \quad b = 1,\ldots,B, \tag{11}$$

$$\sum_{i \in s_b} \sum_{k=\max(t;t_1^i)}^{\min(t+t^m;t_2^i;t_2^b)} Y_k^i + LY_t^b \leqslant L \qquad \begin{array}{l} b = 1,\ldots,B \ \text{ with } S_b \notin \emptyset, \\ t_1^b < t < t_2^b, \end{array} \tag{12}$$

$$\sum_{b=K_{m-1}+1}^{K_m} \sum_{k=\max(t_1^b,t-t^m+1)}^{\min(t,t_2^b)} Y_k^b \leqslant M_t^m \quad t = 1,\ldots,T, \ m = 1,\ldots,M, \tag{13}$$

$$\sum_{b=1}^{B} \sum_{k=\max(t_1^b, t-t^m+1)}^{\min(t, t_2^b)} Y_k^b \leqslant M_t \quad t = 1, \ldots, T, \tag{14}$$

$$Y_t^b = 0, 1 \quad \text{all } b \text{ and defined } t, \tag{15}$$

where t_1^b and t_2^b are the earliest and latest starting times for inspection of bus b, L is a large number, and,

$$Y_t^b = \begin{cases} 1 & \text{if inspection on bus } b \text{ starts at time } t, \\ 0 & \text{otherwise.} \end{cases}$$

Problem (P-2) is basically the same as (P-1). In the formulation of (P-2), however, the variables Z_t^m, that denote the number of maintenance lines assigned for inspection type m during time period t, are deleted because they can be calculated as follows:

$$Z_t^m = \sum_{b=K_{m-1}+1}^{K_m} \sum_{k=\max(t_1^b, t-t^m+1)}^{\min(t, t_2^b)} Y_k^b.$$

The formulation of (P-2) is very close to a network formulation. This means that if we replace constraints (15) with regular linear programming non-negativity constraints, $Y_t^b = 0$ (constraints (15′)), there is a good chance to have integer solution. However, there is no guarantee that if we solve the problem as an LP problem we always get an integer solution.

With the appearance of constraints (10) and (11) in the formulation of (P-2), we can replace constraints (12) with the following set of constraints:

$$Y_t^b + \sum_{k=\max(t;t_1^i)}^{\min(t+t^m;t_2^i;t_2^b)} Y_k^i \leqslant 1, \quad b = 1, \ldots, B \quad \text{with } S_b \notin \emptyset,\ t_1^b < t < t_2^b,\ i \in S_b. \tag{12′}$$

We name the new problem, with new constraints (12′) and (15′), (P-2′). (P-2′) can be written in general form as:

$$\begin{aligned} \text{Maximize} \quad & CY \\ \text{Subject to} \quad & AY = 0, \\ & Y \geqslant 0, \end{aligned}$$

Where matrix A is the coefficient matrix corresponding to all constraints, and vectors Y and C are defined as follows:

$$Y = (Y_t^b : t_1^b < t < t_2^b;\ b = 1, \ldots, B),$$

$$C = (C_t^b) = \left(\left[w_1 \min(t^{m(b)}, T - t + 1) - \sum_{k=t}^{\min(t+t^{m(b)}, T)} w_2 S_k^b \right] : t_1^b \leqslant t \leqslant t_2^b;\ b = 1, 2, \ldots, B \right),$$

where $t^{m(b)}$ is the time required to perform maintenance type m for bus b. Matrix A can be broken down into four submatrices A_1, A_2, A_3, and A_4 corresponding to constraints (10), (11), (12′), (13), and (14), respectively. Each of these submatrices along with the objective function of (P-2′) can be viewed as a subproblem. These subproblems can be solved to find integer solutions very easily. The optimal integer solution for subproblem 1 (related to A_1) can be obtained simply by inspection and without using any formal solution procedure as follows:

If we define:

$$T_{tb} = s : C_s^b = \max\{C_s^b : t_1^b < t < t_2^b\} \quad \text{for } b = 1, 2, \ldots, B.$$

Then the optimal solution for subproblem 1 is

$$Y_t^b = \begin{cases} 1 & \text{If } t = T_{tb} \text{ AND } (b\text{th constraint is an equality constraint OR } \mathrm{C}_t^{b3}0), \\ 0 & \text{otherwise.} \end{cases}$$

Matrix A_2 can be broken down into K submatrices, $A_2^1, A_2^2, \ldots, A_2^K$, where:

$$K = \sum_{b=1}^{B} n(S_b)$$

and $n(S_b)$ is the cardinality of matrix S_b. Now each of the A_2^i matrices with P rows can be changed to a network matrix if we subtract row P from row P-1, row P-1 from row P-2, …, and row 2 from row 1. Each of the two remaining submatrices, A_3 and A_4 can also be changed to a network matrix by subtracting each of their rows from the row above it in the same manner as A_2^i's. This allows us to solve the subproblems 2, 3, and 4 very easily and obtain integer solution. These properties can be used to design a special procedure based on decomposition to find the optimal or near optimal solutions for the original problem (Shafahi, 1997).

In many small and medium size transit agencies, all regular inspections can be performed in all maintenance bays. In this case we can delete constraints (13) from formulation (P-2). We can also safely assume that if a bus is due for more than one inspection in a particular planning period, we can develop the earliest and latest starting times for each of these inspections in such a way that the actual inspection intervals do not overlap. This assumption is not restrictive because in reality there are only a few independent inspection types. With this assumption, and with the deletion of constraints (12′) from (P-2′), the problem can be formulated as a network problem with a set of additional constraints as follows.

If B is the number of buses and T is the planning period, The network contains $2'B + T + 2$ nodes of three types t, s, and e. Nodes s_i and e_i corresponding to bus i and node t_j corresponding to time $j+1$ of the planning period. There are four types of arcs. Arcs s_i^j connect nodes s_i to nodes t_j and have one unit of flow if the inspection of bus i starts at time $j+1$. Arcs e_j^i connect nodes t_j to nodes e_i and have one unit of flow if the inspection of bus i is finished time j. Arcs t_{j-1}^j connect node t_{j-1} to node t_j and have a flow equal to the number of maintenance lines that are used during a maintenance time slot j. Arcs d_i^i connect nodes e_i to nodes s_i and always have one unit of flow. The upper bound, lower bound, and cost per unit flow for each type of arcs are:

Arc	Upper bound	Lower bound	Cost
s_i^j	1	0	C_i^j
e_j^i	1	0	0
t_{j-1}^j	Mi	0	Wi
d_i^i	1	1	0

where

$$C_i^j = -W_2 \sum_{k=j+1}^{J+t^{m(i)}} S_k^i \quad \text{if } j \leqslant T, \text{ and } C_i^T = 0$$

and $t^{m(i)}$, and S_k^i are defined as before. Based on the above notation we can formulate the problem as follows:

$$\text{Maximize} \quad \sum_{t_{i-1}^i} W_1 t_{i-1}^i + \sum_{S_i^j} C_i^j S_i^j \tag{P-3}$$

$$\text{Subject to} \quad \sum_i s_i^0 - t_0^1 = 0, \tag{16}$$

$$\sum_i s_i^j + t_{j-1}^j - t_j^{j+1} - \sum_i e_j^i = 0 \quad 1 \leqslant j \leqslant T, \tag{17}$$

$$t_T^{T+1} - \sum_i e_T^i = 0, \tag{18}$$

$$-\sum_j s_i^j + d_i^i = 0 \quad 1 \leqslant i \leqslant B, \tag{19}$$

$$\sum_j s_j^i - d_i^i = 0 \quad 1 \leqslant i \leqslant B, \tag{20}$$

$$s_i^j - e_{j+t^{m(i)}}^i = 0 \quad \forall j : 2 \leqslant j + t^{m(i)} \leqslant T \text{ and } \forall i, \tag{21}$$

$$\left.\begin{aligned} &0 \leqslant t_{i-1}^i \leqslant M_i \quad \forall i, \\ &0 \leqslant s_i^j \leqslant 1 \qquad \forall i \wedge j, \\ &0 \leqslant e_i^j \leqslant 1 \qquad \forall i \wedge j, \\ &1 \leqslant d_i^i \leqslant 1 \qquad \forall i. \end{aligned}\right. \tag{22}$$

To simplify the formulation of (P-3) we can assume that vectors X, and C, and matrix A are defined as

$$X = (x_i) = (t, s, e, d),$$
$$C = (w, c, 0, 0),$$
$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix},$$

where t, s, e, d, w, c, A_1, and A_2 are defined as:

t	$(t_0^1, t_1^2, \ldots, t_T^{T+1})$
s	$(s_i^j : i = 1, \ldots, B, j = t_1^i - 1, \ldots, \min\{t_2^i - 1, T\})$
e	$(e_j^i : i = 1, \ldots, B, j = \min\{t_1^i + t^{m(i)}, T+1\}, \ldots, \min\{t_2^i + t^{m(i)}, T+1\})$
d	$(d_i^i : i = 1, \ldots, B)$
w	$(W_1, W_1, \ldots, W_1, 0), aT + 1$ vector.
c	$(C_i^j : i = 1, \ldots, B, j = t_1^i - 1, \ldots, \min\{t_2^i - 1, T\})$
A_1	coefficient matrix corresponding to constraints (16)–(20)
A_2	coefficient matrix corresponding to constraints (21)

Problem (P-3) can be rewritten as:

$$\begin{aligned} \text{Maximize} \quad & CX \\ \text{Subject to} \quad & AX = 0, \\ & L \leqslant X \leqslant U, \end{aligned}$$

where L and U are the vectors of the lower and upper bounds corresponding to each element of X.

In this formulation we define a dummy time slot $T + 1$ such that all buses that can start and/or finish their inspection during the next planning period will be allowed to have their inspections during this time slot without any cost or benefit to the system.

It can be observed that matrix A_1 has a network structure and the transpose of matrix A_2 also has a network structure. This property is shown in example 3 and may be used for designing algorithms based on decomposition methods.

3.2. Model testing

Three examples are used to show the implementation of the models described above and their characteristics. CPLEX solver is used in all examples. The first example is used to show the implementation of the model formulated in (P-2). This example is designed for 10 buses, five maintenance types, six maintenance lines, and for a 3-day planning period (each day is 24 h).

Table 1 shows the times during which the ten buses, that are to be scheduled for inspection, have scheduled services in the 3-day (72 h) planning period. This information can be used to determine the idle and busy times for each bus (matrix S). Table 2 contains the earliest and latest maintenance starting times for these buses. Table 3 is the maintenance line – inspection type compatibility table (matrix K). The time needed for performing each type of inspection is shown in Table 4. Finally, the solutions are shown in Tables 5 and 6. Table 5 depicts the time at which the required inspection type should be started, and Table 6 shows the number of maintenance lines that should be assigned to each inspection type.

The second example is used to show the properties described in Section 3.1. Consider formulation (P-2). Assume that we want to schedule three buses for inspection in a planning period which has eight time slots (e.g. hours). There are three types of inspections. The required times for each type of inspection are $t^1 = 2$, $t^2 = 3$, and $t^3 = 5$. We assume that the first bus needs inspection Types 1 and 3; the second bus needs inspection type 2; and the third bus needs inspection type 3.

Table 1
Daily service window for 10 buses for the 3-day planning period (input)

Bus no.	First day				Second day				Third day			
	Service 1		Service 2		Service 1		Service 2		Service 1		Service 2	
	Start	End	Start	End	Start	End	Start	End	Start	End	Start	End
1	6	22	–	–	30	34	39	45	54	67	–	–
2	6	14	–	–	38	47	–	–	54	67	–	–
3	11	19	–	–	35	34	–	–	62	71	–	–
4	6	18	–	–	30	34	39	45	59	67	–	–
5	6	22	–	–	30	34	39	45	54	70	–	–
6	6	19	–	–	30	38	–	–	62	71	–	–
7	11	19	–	–	38	47	–	–	54	58	63	69
8	6	14	–	–	30	46	–	–	54	67	–	–
9	11	19	–	–	30	34	39	45	54	67	–	–
10	14	22	–	–	35	43	–	–	54	70	–	–

Table 2
Earliest and latest starting time period for different inspection types for 10 buses (input)

Bus no.	Inspection type 1		Inspection type 2		Inspection type 3		Inspection type 4		Inspection type 5	
	Min time	Max time	Min Time	Max time	Min time	Max time	Min time	Max time	Min time	Max Time
1	5	20	10	25	7	30	1	30	10	100
2	15	40	3	27	50	70	4	80	2	100
3	1	10	5	30	2	20	1	40	7	60
4	50	70	40	80	30	90	1	20	1	70
5	40	60	30	50	4	24	5	50	40	90
6	40	60	30	50	4	24	5	50	40	90
7	70	80	2	20	15	29	14	45	50	100
8	25	50	23	41	12	38	1	20	20	50
9	2	20	10	25	7	30	1	30	10	100
10	15	40	3	27	50	70	4	80	2	100

Table 3
Maintenance line-inspection type compatibility table (input)

Maintenance line	Inspection type 1	Inspection type 2	Inspection type 3	Inspection type 4	Inspection type 5
1	1	1	0	0	0
2	0	1	0	1	0
3	1	1	1	0	0
4	0	0	0	0	1
5	0	0	1	1	1
6	1	1	1	1	1

So the ordered list of buses will have numbers 1–4 that correspond to the first, second, third, and again the first bus, respectively.

Table 4
The required time (in number of time slots) for different types of inspection (input)

Inspection type	1	2	3	4	5
Maintenance time	2	3	2	4	5

Table 5
The time period at which a particular inspection should start on each bus (output)

Bus no.	Inspection type 1	Inspection type 2	Inspection type 3	Inspection type 4	Inspection type 5
1	20	25	23	2	68
2	33	19	50	68	23
3	10	23	20	6	29
4	68	56	35	1	46
5	46	35	4	25	49
6	56	50	22	41	45
7	70	20	26	22	72
8	50	26	18	1	20
9	5	20	24	7	71
10	30	3	52	7	44

For this problem, matrix $S = [S_t^b]$, that is extracted from the daily service window is shown in Table 7. Table 8 shows the earliest and latest starting times for the required inspections for the buses. The number of available lines for each type of inspection, and the total number of available maintenance lines that are extracted from the hourly maintenance line-maintenance type compatibility tables are Shown in Table 9.

Table 10a, shows the matrix of coefficients, A. The submatrices A_1 to A_4, which were discussed in Section 3.1, for this example are as follows: A_1 is the submatrix corresponding to constraints (1)–(4). A_2 is the submatrix corresponding to constraints (5)–(9). A_2 contains A_2^1, which corresponds to constraints (5)–(7), and A_2^2, which corresponds to constraints (8) and (9) in Table 10. Submatrix A_3 corresponds to constraints (10)–(24) and finally submatrix A_4 is the matrix corresponding to constraints (25)–(32). The network-shaped transformed matrix is shown in Table 10b.

The third example shows the network structure that can be created for the formulation presented in problem (P-3). Assume we want to schedule three buses for inspections that require 2, 3, and 5 time slots to be completed, respectively, during an 8-time slot planning period. Matrix A for this problem is shown in Table 11 and the underlying network for this problem is shown in Fig. 1.

4. Solution approaches

Due to the large scale nature of the maintenance scheduling problem, finding an optimal solution can be very time consuming. Therefore, we have developed and tested several heuristic solution methods. Fortunately, the bus maintenance scheduling problem has an inherent structure that can be exploited to devise heuristic solution procedures that are relatively quick and provide near optimal solutions.

Four different heuristic methods are tested: a gap acceptance method, a rounding method, a two-phases method, and a network-based method. The formulation of problem (P-2) is the basis

Table 6
Number of maintenance lines that should be assigned to each inspection type in different time slots for each day (output)

First day time of day inspection						Second day time of day inspection						Third day time of day inspection					
	1	2	3	4	5		1	2	3	4	5		1	2	3	4	5
1				2		1 (25)[a]		2	1	2	1	1 (49)					1
2				3		2 (26)		2	1	1	1	2 (50)	1	1	1		1
3		1		3		3 (27)		2	1	1	1	3 (51)	1	1	1		1
4		1	1	3		4 (28)		1		1		4 (52)		1	1		1
5	1	1	1	1		5 (29)					1	5 (53)			1		1
6	1			1		6 (30)	1				1	6 (54)					
7				3		7 (31)	1				1	7 (55)					
8				3		8 (32)					1	8 (56)	1	1			
9				3		9 (33)	1				1	9 (57)	1	1			
10	1			2		10 (34)	1					10 (58)		1			
11	1					11 (35)		1	1			11 (59)					
12						12 (36)		1	1			12 (60)					
13						13 (37)		1				13 (61)					
14						14 (38)						14 (62)					
15						15 (39)						15 (63)					
16						16 (40)						16 (64)					
17						17 (41)				1		17 (65)					
18			1			18 (42)				1		18 (66)					
19		1	1			19 (43)				1		19 (67)					
20	1	3	1		1	20 (44)				1		20 (68)	1			1	1
21	1	3	1		1	21 (45)					1	21 (69)	1			1	1
22		2	1	1	1	22 (46)	1				1	22 (70)	1			1	1
23		1	2	1	2	23 (47)	1				1	23 (71)	1			1	2
24		1	2	1	2	24 (48)					1	24 (72)					3

[a] The numbers in parentheses indicate the consecutive time period number.

Table 7
Matrix S, daily service windows for buses in the second example

Bus no.	Time							
	1	2	3	4	5	6	7	8
1	0	1	1	1	0	0	0	0
2	1	1	1	1	0	0	0	0
3	0	1	1	1	1	0	0	1
4	0	1	1	1	0	0	0	0

for designing and implementing the first three heuristic approaches. The network-based method, on the other hand, is based on the formulation of problem (P-3).

4.1. The gap acceptance algorithm

The basis for the gap acceptance algorithm (GAA) is to decide upon an acceptable gap between the current and the optimal solutions and terminate the branch and bound procedure when a

Table 8
The earliest and latest time slots for starting each type of inspection for buses in the second example

Bus no.	Maintenance type	Minimum time start	Maximum time start
1	1	1	6
2	2	2	5
3	3	4	7
4	3	5	10

Table 9
Number of available maintenance lines in example 2

Inspection type	Time							
	1	2	3	4	5	6	7	8
1	1	1	2	2	2	1	1	1
2	1	1	2	2	2	1	1	1
3	1	1	2	2	2	1	1	1
Total	1	1	2	2	2	2	2	2

feasible solution is found where its objective function value is less than this acceptable gap from the upper bound (for a maximization problem). This upper bound is the solution to the linear programming relaxation. This causes the branch and bound procedure to be more efficient in terms of the solution time. The acceptable gap can be an absolute distance or a relative distance as

$$\text{best node-best integer}/(1 + \text{best node}).$$

4.2. The rounding algorithm

As a result of using the gap acceptance algorithm, the authors noted that the solution is not very sensitive to the value of the gap that is accepted. This means that an integer solution for the maintenance scheduling problem can be obtained by appropriately rounding the linear programming relaxation solution. This integer solution can have an objective function value that is close to the optimal solution. This fact was used to design a quicker algorithm that can find an integer solution in the proximity of the solution of the linear programming relaxation of the problem. The algorithm that is called the rounding algorithm simply rounds the optimal solution of the relaxed linear programming problem, accepts an infinity gap, and tries to find a feasible integer solution based on the rounded numbers.

4.3. The two-phase algorithm

Looking at the results of the relaxed linear problems that are obtained from the relaxation of the integrality constraints, we can see that less than two percent of the variables have positive values. The two-phase algorithm is designed to take advantage of this situation. In the two-phase heuristic method we set all variables that have zero values in the solution of the relaxed linear

Table 10
Matrix of coefficients, A, for example 2

Constraint number	Decision variable																		Right side
	Y_1^1	Y_2^1	Y_3^1	Y_4^1	Y_5^1	Y_6^1	Y_2^2	Y_3^2	Y_4^2	Y_5^2	Y_4^3	Y_5^3	Y_6^3	Y_7^3	Y_5^4	Y_6^4	Y_7^4	Y_8^4	
(a) *Before transformation*																			
1	1	1	1	1	1	1													$=1$
2							1	1	1	1									$=1$
3											1	1	1	1					$=1$
4															1	1	1	1	$\leqslant 1$
5				1											1				$\leqslant 1$
6					1										1	1			$\leqslant 1$
7						1										1	1		$\leqslant 1$
8					1	1									1				$\leqslant 1$
9						1										1			$\leqslant 1$
10	1																		$\leqslant 1$
11		1																	$\leqslant 1$
12		1	1																$\leqslant 2$
13			1	1															$\leqslant 2$
14				1	1														$\leqslant 2$
15					1	1													$\leqslant 2$
16							1												$\leqslant 1$
17							1	1											$\leqslant 2$
18							1	1	1										$\leqslant 2$
19								1	1	1									$\leqslant 2$
20											1								$\leqslant 2$
21											1	1			1				$\leqslant 2$
22											1	1	1		1	1			$\leqslant 1$
23											1	1	1	1	1	1	1		$\leqslant 1$
24											1	1	1	1	1	1	1	1	$\leqslant 1$
25	1																		$\leqslant 1$
26	1	1					1												$\leqslant 1$
27		1	1				1	1											$\leqslant 2$
28			1	1			1	1	1		1								$\leqslant 2$
29				1	1			1	1	1	1	1			1				$\leqslant 2$
30					1	1			1	1	1	1	1		1	1			$\leqslant 2$
31						1				1	1	1	1	1	1	1	1		$\leqslant 2$
32											1	1	1	1	1	1	1	1	$\leqslant 2$

(*continued on next page*)

Table 10 (*continued*)

Constraint number	Decision variable																		Right side
	Y_1^1	Y_2^1	Y_3^1	Y_4^1	Y_5^1	Y_6^1	Y_2^2	Y_3^2	Y_4^2	Y_5^2	Y_4^3	Y_5^3	Y_6^3	Y_7^3	Y_5^4	Y_6^4	Y_7^4	Y_8^4	
(b) *After transformation*																			
1	1	1	1	1	1	1													= 1
2							1	1	1	1									= 1
3											1	1	1	1					= 1
4															1	1	1	1	⩽ 1
5				1	−1											−1			⩽ 1
6			1	−1										1	−1				⩽ 1
7						1										1	1		⩽ 1
8					1										1	−1			⩽ 1
9						1										1			⩽ 1
10	1	−1																	⩽ 1
11			−1																⩽ 1
12		1		−1															⩽ 2
13			1		−1														⩽ 2
14				1		−1													⩽ 2
15					1	1	−1												⩽ 2
16								−1											⩽ 1
17									−1										⩽ 2
18							1			−1									⩽ 2
19								1	1	1	−1								⩽ 2
20												−1			−1				⩽ 2
21													−1			−1			⩽ 2
22														−1			−1		⩽ 1
23																		−1	⩽ 1
24											1	1	1	1	1	1	1	1	⩽ 1
25		−1					−1												⩽ 1
26	1		−1					−1											⩽ 1
27		1		−1					−1		−1								⩽ 2
28			1		−1		1			−1		−1			−1				⩽ 2
29				1		−1		1					−1			−1			⩽ 2
30					1				1					−1			−1		⩽ 2
31						1				1								−1	⩽ 2
32											1	1	1	1	1	1	1	1	⩽ 2

Table 11
Coefficient matrix A, for example 3

A_1	−1								1																									=0
	1	−1								1																								=0
		1	−1								1						−1																	=0
			1	−1								1						−1																=0
				1	−1								1	1					−1															=0
					1	−1									1	1				−1														=0
						1	−1										1	1			−1													=0
							1	−1											1			−1	−1											=0
								1	−1											1				−1										=0
									1																−1	−1								=0
										−1	−1	−1	−1	−1	−1												1							=0
																−1	−1	−1										1						=0
																			−1	−1	−1								1					=0
																							1	1	1	1	1	1			−1			=0
																													1	1		−1		=0
																														1			−1	=0
A_2								1												−1														=0
									1												−1													=0
										1												−1												=0
											1												−1											=0
												1												−1										=0
													1												−1									=0
														1												−1								=0
															1												−1							=0
																1												−1						=0

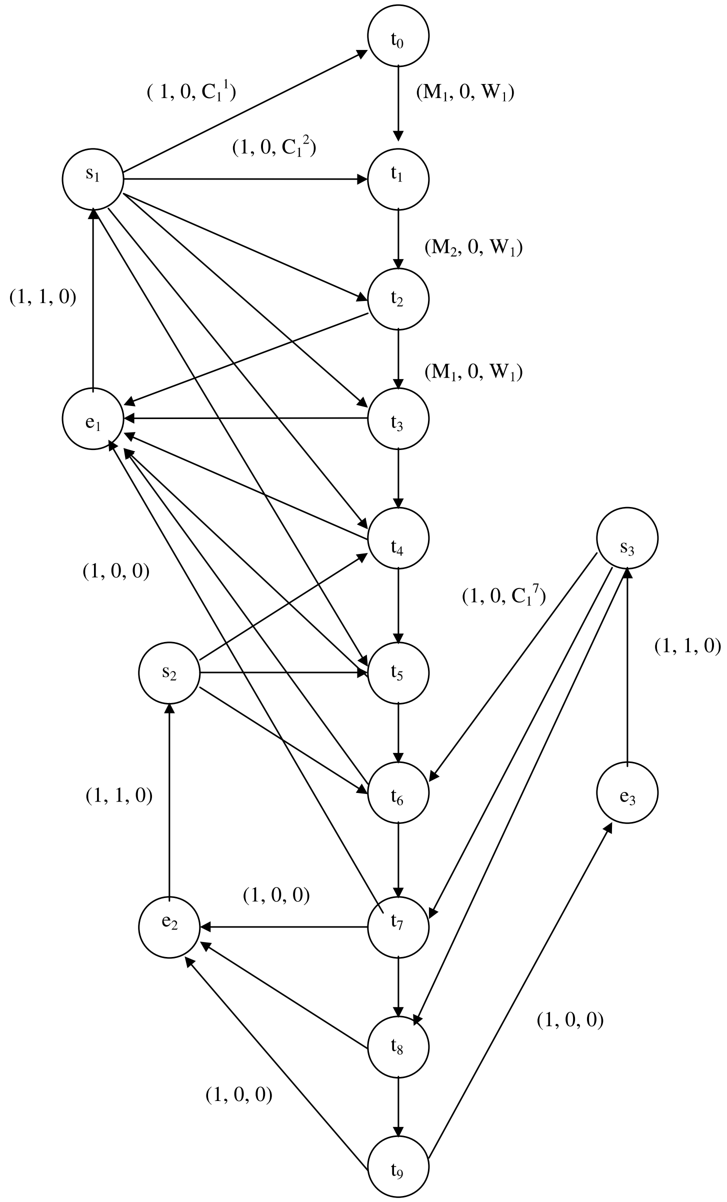

Fig. 1. The underlying network for example 3.

programming problem, equal to zero and solve a very small integer problem that contains only variables with positive values. The algorithm proceeds with the following three general steps:

Step 1. Solve the linear programming relaxation of problem (P-2).
Step 2. Set the variables with 0 values equal to 0.
Step 3. Solve the revised formulation for non-zero variables as an integer program.

4.4. The network-based algorithm

The network-based method uses the formulation of problem (P-3) to solve the maintenance scheduling problems. The steps of the algorithm can be summarized as follows:

Step 1. Sort the type s links (the links that bring the flow into the network) based on their benefits. If two links S_i^j and S_k^l have the same benefits, give priority to the link with smaller superscript. The benefit for link S_i^j can be computed as follows:

$$B_i^j = W_1 \min[t^{m(i)}, T - j + 1] - W_2 \sum_{k=j+1}^{\min[T, J+t^{m(i)}]} S_k^i.$$

Step 2. If there are no type S_i^j links with positive benefits stop.
Step 3. Check for the possibility of adding link S_i^j with maximum benefit to the system. If it is not possible to add the link to the system, delete the link from the positive benefit list and go to Step 2.
Step 4. Add the link to both networks and balance the flow in both networks A_1 and A_2.
Step 5. If there is a type t link t_{i-1}^i with a flow equal to its capacity, from both networks delete all type s links that have not been chosen so far; and if they are chosen, their addition to the network requires increasing the flow in link t_{i-1}^i; repeat this step for all link t_{i-1}^i that have flows equal to their capacity.
Step 6. Delete all S_i^m links for all m and go to Step 2.

Addition and deletion of links from the networks have the following meaning. Adding a link to a network is equivalent to assigning a flow of 1 to this link. Deleting a link from the list is equivalent to setting the flow of this link equal to zero.

5. Model implementation

Formulation (P-2) was used to schedule maintenance for a real-word size example. The site for the case study for this research is the Eastern Division of Mass Transit Administration (MTA), Baltimore. The case study was performed by conducting several site visits, interviewing the managers and supervisors of the maintenance shop, and a literature search and review of the maintenance reports.

The Eastern Division is one of the four operating divisions in Baltimore, Maryland. The active revenue fleet assigned to Eastern Division includes 181 of the 814 total transit vehicles of the MTA. All buses are equipped with air-conditioning systems and approximately half are equipped with wheel-chair lift equipment. The vehicles are categorized into four types by their age and whether or not they are equipped with wheelchair lift equipment. These categories are regular old, wheelchair old, regular new, and wheelchair new. The number of buses in each of the above groups is 64, 15, 34, and 68, respectively.

The division has facilities (lines) for repair or inspection of 12 buses simultaneously in the repair shop. Four of these 12 lines are the bays where major repair can be performed. This facility is sufficient for current operation. However, if necessary, the open area in front of the shop can be used for non-major inspections.

Five types of inspections are performed in the division. Inspection Type 1 is a brake and safety inspection and is done every 3000 miles. Inspection type 2 is a major engine inspection and is done every 9000 miles. Inspection type 3 is the same as inspection type 2 plus transmission inspection and is done every 27,000 miles. When type 3 inspection is performed type 2 and type 1 inspections are also completed. Also, type 2 inspection contains type 1 as well.

A type 4 inspection targets air conditioning equipment. This inspection is based on time instead of mileage. An air conditioning equipment inspection is performed every 120 days. Type 5 inspection is wheelchair lift equipment inspection. There are two types of wheelchair lift equipment. The wheelchair lift equipments that fail most have screw drive lifts. The buses with these equipments are brought for inspection once a month. The second type of wheelchair lift equipment is more reliable and is called chain drive wheelchair. The buses equipped with this type of equipment are inspected once every two months.

Each type 1 inspection on the average takes about 1.5 h to be completed. The average time for inspection type 2 is 8 h and for inspection type 3 is 16 h. An A/C inspection takes about 8 h and a wheelchair lift equipment inspection takes about 2 h to be completed. During the inspections if it is determined that a bus has a problem, it is kept in the shop for repair.

A computer program that tracks the bus mileage generates the list of all buses that need different types of maintenance every day. This program works based on the mileage at the last inspections of each type and the current mileages of the buses, that are computed based on the buses' daily schedules. When the mileage of a bus reaches within 300 miles of the mileage at which an inspection is due, its ID number appears on the daily inspection list with an indication that the bus is -300 miles overdue for inspection. This means that the bus is close to inspection time. Every day the daily operational mileage is added to this number until it passes the inspection point.

The list of buses that need inspection is sorted based on the mileage by which an inspection is overdue for the buses. This list is updated every day because the average daily mileage of the buses varies and a bus that is at the bottom of the list today may be on the top of the list tomorrow.

If a bus needs inspection type 1 but is not inspected by the time inspection type 2 is nearly due, it automatically disappears from the list of inspection type 1 and is added to list of buses that need inspection type 2. This procedure also is done for inspection Types 2 and 3. The inspection and maintenance shop in the Division works 24 h a day for 7 days a week to maintain the buses.

Unscheduled maintenance and repair is another task that is performed in the Division. This task stems from road calls or problems that are found in buses during different types of inspections. When a bus has a problem and has to be repaired, depending on the nature of the problem and the failed component, the foreman and the supervisor of each shift decide whether the bus can be repaired in the Division repair shop or has to be sent to the MTAs main repair shop. Usually, major repairs and engine works are done at the MTAs repair shop and component replacements are done in the Division. The discipline for choosing a bus in the repair queue for repair is based on the shortest repair time. This means that a bus that can be repaired faster has priority for repair.

The buses that have to be sent for major repair to the MTAs main shop are parked in an area called failed bus parking. The list of these failed buses and their problems is sent to the MTAs main shop and the buses remain in this parking area until they are called for repair from the MTAs main shop.

Within the Division, buses are classified into two groups: active buses and failed buses. The active buses in the Division may be categorized as scheduled buses and spare buses, and the failed buses are those that are out of service and are in the MTA central repair shop, or are waiting to be sent there. When a bus completes its assigned daily service, it comes back to the garage for cleaning and fueling. After the driver fills in and signs the daily report, the fueler is required to bump test the tires for flats, perform a fast walk around inspection that is called daily inspection, and record actual miles operated. The fueler is also responsible for fueling the bus, and adding any engine oil or coolant if needed. The fluids and fuel usage are recorded on the fuel sheet. The fueler may send a bus to inspection or the repair shop for one of the following reasons:

1. The bus is in the list for inspection and the fueler has been asked to send it for inspection.
2. The fueler finds a problem with the bus during daily inspection.
3. A problem has been reported by the driver.

Since detailed scheduling of maintenance activities is not currently practiced in transit agencies and to produce enough examples for testing, a simulation model was built to mimic the current practices of the transit agencies such as those of the Eastern Division when they use the proposed maintenance scheduling models to schedule the buses for inspections.

This simulation model is a general model and with changes in the parameters and data can be used for any transit agency. The main purpose of the model is to ensure that the maintenance scheduling system is run daily over a period of six months while uncertainties in real-world bus transit operations are taken into account via the simulation of the operations. This approach allows us to propagate the effects of any disturbances that happen in daily operations to the following days during the six-month simulated period. For example, if buses break down and remain out of service for several days, then the shortage of these buses is accounted for in the following days operations. In this paper, the simulation of the transit operations for six months can be viewed as repeated runs of the maintenance scheduling algorithms under various daily operational conditions. This paper focuses on the performance of the maintenance scheduling algorithms. Therefore, the details of the simulation model are beyond the scope of this paper. For the interested reader, the details can be found in Shafahi (1997).

The simulation model is used to test and compare the heuristic methods. To insure that the conclusions are reliable and significant, a 182-day (6-month) simulation is run for each of the methods. In other words, the analysis is based on the results of applying each of the heuristic methods 182 times for maintenance scheduling.

Two measures of effectiveness are used to evaluate the performance of the heuristic methods. These are utilization of the maintenance shop, and the total time that spare buses are used as substitutes for buses that are under inspection. The total simulation time for 182 days is used as a measure to compare the total time that is required for each method to find the solutions (execution, I/O, etc.). We can use this time because the simulation time for all cases is almost the same. The average, minimum, maximum, and standard deviation of the execution times can be used as a measure of reliability of the solution procedures. An IBM compatible personal computer with a 200 MHZ processor and 64 MB memory is used in these evaluation procedures.

5.1. Using a conventional branch and bound procedure

To have a base to compare the heuristic methods we tried to find the optimal solutions to the maintenance scheduling problems for all of the 182 simulation days, by using the conventional branch and bound algorithm. The CPLEX Mixed Integer Programming solver (MIP) was used to solve these integer problems. To find the optimal solution of the maintenance scheduling problems for all 182 days and simulate all garage activities during these days, it takes a total of about 13.5 days (13 days, 11 h, and 59 min). Considering that only a few minutes are needed for simulating the system during these days, we conclude that almost all of this time is used by CPLEX to solve the maintenance scheduling problems during these 182 simulation days.

The average solution time for each scheduling problem (for one day) was 1.78 h and the standard deviation was 3.40 h. Among these 182 attempts to solve the problem, the minimum time for solving one problem was 18 min while the maximum time was 31.57 h. During these 182 days of simulation, CPLEX found the optimal solution for the maintenance scheduling problem in only 90 cases. In 85 cases because of memory limitation, the branching procedures were terminated and the best solutions at the time of termination were accepted as final solutions. Considering the minimum possible increment to the objective function we concluded that in 68 out of these 85 cases, the best solutions at termination times were in fact the optimal solutions. In 14 cases the solutions were at most only one incremental unit away from the optimal solutions and in three cases the best solutions were at most two incremental units away from the optimal solutions. In 7 cases the procedures were interrupted because of the high number of iterations (500,000 iterations). However, the best solutions in each case were the optimum solutions.

The minimum incremental unit for the objective function of problem (P-2) can be computed as:

$$\text{minimum increment} = \min\{w_1, w_2, d\},$$

where

$$d = \begin{cases} w_1 - w_2 & \text{if } w_1 w_2, \\ w_1 & \text{if } w_1 = w_2. \end{cases}$$

In these analyses we have chosen $w_1 = w_2 = 100$, therefore, the minimum objective function increment is 100 units.

An overall review of the results of this experiment shows that the conventional branch and bound algorithm cannot be used as a reliable procedure. The scheduler should run the procedure for solving the maintenance scheduling problem at least every day, while we have solution times that are larger than 24 h. Moreover, the real average and maximum times are larger than what was reported because, as mentioned above, for only 90 cases (less than 50% of the cases) we obtained the exact optimum solution and for the others the procedures were terminated for one reason or the other, and CPLEX could not verify the optimality of the solutions. Of course, by using more memory or using a hard drive to store the branching activities we can solve the memory problem and by setting CPLEX limitation on the number of iterations at a larger number we can find an optimal solution for each day of simulation. However this increases the solution times significantly. Therefore having a quick and reliable algorithm for solving the maintenance scheduling is a must for practical applications.

5.2. The gap acceptance algorithm

In the last section we introduced a minimum incremental value of the objective function that is 100 for our case. Here each increment represents 1 h idle time of a maintenance line or 1 h of using a spare bus as a substitute for a bus that is under inspection during the schedule planning period of one week. Therefore, if we choose an acceptable gap of less than 100, for example 99, we can make sure that we would have an optimal solution in the branch and bound procedure. If we accept an absolute gap of less than 200, for example 199, the obtained solution has at most a value equal to 1 h of idle maintenance line, or 1 h of spare bus use, away from the optimum point and so on.

This method was used with the absolute acceptance gaps equal to 99, 199, 299, 399, 499, 599, 699, and 799. For the absolute gap of 99, as it was expected, the results of the simulation were exactly the same as those of the branch and bound method while the total simulation time for 182 simulation days was 2 days, 12 h, and 10 min. This is less than 19% of the regular branch and bound method. The average, standard deviation, minimum, and maximum solution times of the maintenance scheduling problems using this gap were 19.57, 72.77, 0.3, and 830.58 min that are 18%, 36%, 100%, and 44% of the similar statistics with the branch and bound method, respectively.

The results of using other values as absolute accepted gaps are reported in Shafahi (1997), but generally, it seems that the objectives, utilization and total spare bus-hours used are not so sensitive to the accepted gap value while the solution times and their standard deviations reduce dramatically when we choose a relatively large value for the absolute accepted gap. For example if we accept an absolute gap equal to 799, this means that the solution can be away from the optimal solution at most equal to the value of 1 h of maintenance line or 1 h of spare bus use per day (7 h per week), we have results as good as the optimal solution while the average solution time is less than 1 min and the standard deviation of the solution time is equal to 0.35 min. These values are 0.8%, and 0.2% of the same statistics for the branch and bound algorithm. The same results can be seen when we use relative accepted gaps. The test results for relative accepted gaps to 0.005, 0.01, 0.05, 0.1, and 0.5 are also reported in Shafahi (1997).

5.3. The rounding algorithm

The results of using the rounding method in the 182-day simulation show that the total simulation time is reduced to 2 h and 38 min. The solution time in this method has an average of 35 and a standard deviation of 16 s. The maximum time for solving one problem by this method was about 1 min.

5.4. The two-phase algorithm

Using the two-phase algorithm for solving the maintenance scheduling problem during the 182-day simulation a record of 1 h and 19 min of total simulation time was obtained. For the linear part the average, standard deviation, minimum, and maximum solution times were 1.6, 1.6, 0.05, and 11.31 second, respectively. These statistics for the integer part were 1.78, 4.41, 0.16, and 56.85 s. The solutions obtained from using this method are also comparable to other methods.

5.5. The network-based algorithm

As discussed before, the network formulations for bus maintenance scheduling are correct if there are not any buses that require two, or more, types of maintenance. To solve this problem two methods can be employed:

1. During a schedule period if a bus needs more than one type of maintenance, choose the maintenance with the higher overdue time and let the bus have the other inspection in the next schedule period.
2. Combine the inspections as one new inspection type with an inspection time equal to the sum of the required time for both inspections. In this case the new combined inspection types can be defined as follows:

Inspection type	Component inspection	Inspection time
6	1, 4	9
7	1, 5	3
8	1, 4, 5	11
9	2, 4	16
10	2, 5	10
11	2, 4, 5	18
12	3, 4	24
13	3, 5	18
14	3, 4, 5	28
15	4, 5	10

The results from implementing these assumption that are called network-based 1 and 2 formulation, respectively, are summarized in detail in Shafahi (1997). The results show that the network-based 1 formulation can find better solutions. This is expected because the network-based 2 formulation is more constrained. The total simulation times using both cases are equal to 2 h and 22 min. An average of 40 s is needed for total solution procedure including solution time and I/O. The standard deviation of the solution times for this method is about 3 seconds that shows the reliability of the algorithms for finding a solution. The other advantages of this method are its independence of other software such as CPLEX and also, the program is not limited by the size of the problems.

5.6. Comparison of the solution methods

Table 12 shows the results of using different heuristic methods for solving the maintenance scheduling problems during the 182 days simulation of the maintenance system. The first three columns of the table show the different methods under different assumptions that were discussed before. The fourth column shows the total simulation times for the 182 days simulation of the maintenance system when using different methods to solve the maintenance scheduling problems. The fifth column shows the number of times that CPLEX needed more than 64 MB memory to

Table 12
Results of using different heuristic models for solving maintenance scheduling problem in 182-days simulation of the maintenance system

Methods			Total simulation time (h)	No. of IMP[a]	No. of ILN[b]	Average solution time (S)	SD solution time (S)	Minimum solution time (S)	Maximum solution time (S)	Utilisation of mainte-nance	Hours spare used
Branch and bound			323:59	85	7	6392	12,241	18	113646	67.46	465
Gap acceptance	Absolute gap	99	60:10	18		1174	4366	18	49,835	67.46	465
		199	15:15	3		283	1363	1	14576	66.98	487
		299	7:48	1		136	683	16	8812	68.1	455
		399	5:14			85	205	22	2588	67.99	470
		499	4:24			69	84	1	898	68.13	443
		599	3:57			63	92	19	1095	68.85	443
		699	3:53			59	43	20	425	67.46	441
		799	3:34			52	21	1	117	68.44	499
	Relative gap	0.005	45:46			889	2471	1	15,405	67.27	467
		0.01	9:29			171	536	1	4796	68.09	433
		0.05	3:32			53	21	21	125	68.39	498
		0.1	3:28			53	21	20	124	68.39	498
		0.5	3:33			53	21	20	124	68.39	498
Rounding			2:38			35	16	16	139	68.33	477
Two-phases			1:19			3.4	4.7	0.2	60	67.36	490
Net-base	1		2:22			40	3	30	49	65.18	399
	2		2:22			40	4	32	52	63.55	750

[a] Number of times that the procedures were terminated because of memory limitation.
[b] Number of times that the procedures were terminated because of high number of iterations.

continue the branch and bound procedure (in these cases the best solutions were accepted). The sixth column shows the number of times that the branch and bound procedure was terminated because of the very large number of branching nodes (as discussed before in all these cases the best solutions were optimum). Columns 7, 8, 9, and 10 show average, standard deviation, minimum, and maximum solution times for the different methods. Except for the network-based methods, for which these numbers represent the total procedure time (solution time, I/O time, etc.), for other methods these numbers only represent the integer programming solution times. Therefore a few seconds for problem formulation, I/Os, and CPLEX calling time should be added to these numbers. Finally, columns 11 and 12 show the utilization of the maintenance systems and the total bus-hours in which the spare buses are used as substitute for the buses that are under inspection. These two columns represent the two parts of the objective function in the maintenance scheduling problem.

As can be seen from Table 12 in terms of optimality all methods provide acceptable solutions and except for the case of network-based 2 formulation, that is different because of the related constraints, there is not a significant difference between the solutions. In terms of the average solution time the two-phase algorithm is superior. In terms of the reliability of finding a solution the network-based 1 algorithm is superior. As mentioned in the last section, the network-based method has two more advantages over the other methods. First, the program is independent of other software, and second, there is no limitation on the size of the problems to be solved.

6. Conclusions and future research directions

6.1. Conclusions

In recent years computerized fleet management systems have been developed which allow the transit agencies to assign buses to a specific schedule, and maintain that assignment for a long period of time. When such systems are in place, the transit managers prefer that the operators drive the same bus to the extent possible.

Since the buses have fixed schedules, one can predict when they are available in the garage, and therefore, any inspection and maintenance can be accomplished when the buses are parked in the garage. Consequently, one need not pull a bus out of service for routine inspection and maintenance and the amount of service interruptions can be kept to a minimum.

In this research the maintenance systems in bus transit operations were studied and several formulations for scheduling preventive maintenance in a bus transit system were presented. All of these formulations are based on integer programming and some of them possess special structures that can be exploited for devising quick and near optimal heuristic solution algorithms. Some properties of the maintenance scheduling problem formulations were discussed. Based on these properties several procedures for solving the preventive maintenance scheduling problems were also presented.

A case study was undertaken based on real-word data and a simulation program was used to compare the efficiency and the effectiveness of the proposed solution procedures. The efficiency and reliability of the proposed heuristic methods were studied in depth by implementing them in the simulation program and using them for scheduling maintenance for a 182-day simulation of the maintenance system. The results showed that almost all of the proposed methods can find solutions for the maintenance scheduling problem that are very close to the optimal solutions while the solution times for these methods are generally less than 1% of the solution times for the conventional branch and bound algorithm.

Model testing, sensitivity analyses, and the case study results lead to the following conclusions:

(1) Scheduling maintenance activities can have significant impacts on the efficiency and effectiveness of the maintenance operations, and provision of the scheduled services. The results of this research indicate that maintenance scheduling results in significant reduction in the amount of time the buses have to be pulled out of their scheduled services for preventive maintenance. This in turn means a significant reduction in the number of substitutions that need to be made when buses are not available for their scheduled services.

(2) As a result of maintenance scheduling since the buses that have scheduled services are not pulled out of service to be inspected as often, the need for use of spare buses as substitutes declines. This, in the long run, results in reduction of the spare bus fleet size, and can amount to significant operational and capital cost savings. More efficient use of the personnel time in the long run can result in reduction of the maintenance crew size and translate into significant labor cost savings.

(3) As a result of maintenance scheduling, more buses are inspected on time and this reduces the negative impacts of overdue maintenance. Performing on-time maintenance in the long run will results in reduction of road calls and breakdown maintenance costs.

(4) The sensitivity analysis on different inspection factors that was done as part of this research showed that the maintenance systems are very sensitive with respect to some of the inspections. It means more detailed inspections of these types are provided (by for example better diagnostic system), or if the frequency of these inspections are increased the agencies can have fewer road calls and a more reliable system. This observation was especially extreme for pre-run and daily inspections. Performing very detailed inspections on a daily basis is not applicable and realistic. However, a bit more concentration on this type of inspection (by for example training drivers and fuelers to conduct better pre-run and post-run inspections) can increase the reliability of the system significantly.

In the long run, the maintenance scheduling models proposed in this research can be embedded in an automatic system that ties into an advance database system containing up-to-date information regarding all of the buses as well as maintenance bays and manpower information. The scheduling models can extract necessary information for bus maintenance scheduling from the database and schedule buses for maintenance by solving the maintenance scheduling problem. In the short run, however, any one of the proposed optimization models can be used as an off-line maintenance scheduling system to develop and update schedules for maintenance activities.

6.2. Recommendations for further research

There are several potential areas for future research in developing bus maintenance systems. The following areas are the immediate opportunities for extension of the current research.

1. Modify, extend, and use the simulation model to evaluate different proposed and practiced policies and strategies in transit bus maintenance systems.
2. Develop other heuristic or exact solution procedures based on decomposition methods for solving the maintenance scheduling problems.
3. Develop a method to gather the information that is needed to estimate the parameters that are used in the simulation models and also in the maintenance scheduling formulation such as, maintenance factors, w_1, and w_2.
4. Develop a model for assignment of maintenance crew to various work shifts under various regulations and other constraints to maximize the efficiency of maintenance system.
5. Study the potential of implementing different automatic diagnostics systems in bus maintenance to improve the current practice in daily and preventive inspection and examine the existing tradeoff between increased maintenance cost due to more involved inspections and reduced number of road calls (more reliable service).

Acknowledgements

The authors wish to express their gratitude toward the operations and management personnel at the Eastern Division of the Mass Transit Administration in Baltimore, MD, for their efforts in providing the data used in this research, and their invaluable comments and suggestions throughout the course of the research. Special thanks go to Reid B. Kreider, Chief of Operations, and John Smolenski, Mobility Superintendent.

References

Andrle, S.J., 1986. Summary: research needs in transit bus maintenance. Transportation Research Record 1066.

Bladikas, A.K., Papadimitriou, C., 1986. Analysis of bus transit's maintenance efficiency using section 15 data. Transportation Research Record 1066.

Maze, T.H., Cook, A.R., 1987. Theory and practice of transit bus maintenance performance measurement. Transportation Research Record 1140.

Purdy, J.E., Wiegmann, J.D., 1987. Vehicle maintenance: cost relationship and estimating methodology. Transportation Research Record 1140.

Shafahi, Y., 1997. Bus maintenance systems and maintenance scheduling. Ph.D. Dissertation, Department of Civil Engineering, University of Maryland, College Park, MD.