ORIGINAL RESEARCH   Open Access

An employee transporting problem

Ümit Yüceer

Abstract

An employee transporting problem is described and a set partitioning model is developed. An investigation of the model leads to a knapsack problem as a surrogate problem. Finding a partition corresponding to the knapsack problem provides a solution to the problem. An exact algorithm is proposed to obtain a partition (subset-vehicle combination) corresponding to the knapsack solution. It requires testing and matching too many alternatives to obtain a partition. The sweep algorithm is implemented in obtaining a partition (subset-vehicle combination) in an efficient manner. Illustrations are provided to show how the algorithms obtain solutions.

Keywords: Employee transportation; Set partitioning; Knapsack problem; Sweep algorithm

Background

A number of employees will be picked up from various places (bus-stops) within a city and brought to the plant. At the end of the work day, they will be returned back to where they were picked up. A similar situation holds for school students. The students are picked up from their homes and returned back to their homes after school. Transporting the passengers needs to be accomplished within a given time period. It is obvious that work at a plant starts at a specified time (8:30 in the morning for instance), and the first class starting time at some schools is 8:40 in the morning. The workers or the students must reach the plant or the school on time. We call this problem an *employee transporting* problem.

A set of different types of vehicles is used during the transportation of those passengers. Vehicles may differ in capacity, operating cost, and speed. Cruising within a city has speed limits, and transporting passengers within a city requires strict obedience to these speed limits. Subsequently, all the vehicles are assumed to have the same speed while traveling in the city streets.

Even though the vehicles may have different operating costs, they are chartered according to their capacities and paid accordingly. Every vehicle will be assigned to a subset of bus stops according to its capacity and the distance (time) required for traveling to this subset. Clearly, the number of passengers picked up or delivered cannot exceed the capacity of the vehicle. Further, the distance

(time) required to travel through this subset of destinations (bus stops) via a route must remain less than a predetermined distance (time) amount. Any route is acceptable as long as the traveling distance (time) does not exceed the predetermined time, and thus, it is not necessarily the traveling salesman (TSP) route.

The vehicle routing problem (VRP) or vehicle scheduling problems concern with the distribution of goods between the center (depot) and the customers (destinations-final users). The objective is to determine an optimal set of routes by a fleet of identical vehicles to serve these customers. The total demand of the customers on a route cannot exceed the vehicle capacity. This type of problem is called capacitated vehicle routing problem (CVRP). The fleet of the vehicles may differ in size, capacity, and the operating costs. This gives rise to heterogeneous vehicle routing problem (HVRP) or fleet mix problem in the literature. All these problems are the extensions and/or generalizations of the vehicle routing problems. Toth and Vigo (2002) provide an extensive discussion on VRP and solution methods with some applications. There are also numerous excellent articles on those problems and solution methods in the literature. A general approach to modeling this class of problems is based on the integer formulation of Miller et al. (1960). Various researchers have modified and/or extended their formulation to handle different situations.

The latest developments and the challenges in terms of modeling and the solution techniques for VRP and its

Correspondence: umit.yuceer@toros.edu.tr
Department of Industrial Engineering, Toros University, Mersin 33140, Turkey

Springer

variants are presented in Bruce et al. (2008). Yaman (2006) presents different formulations for HVRP and one formulation with flow variables and also develops strong bounds. Baldacci et al. (2011a) developed an exact method for solving VRP after modeling it based on a partitioning problem. Again in another attempt, Baldacci et al. (2011b) describe an effective exact method for solving CVRP based on the set partitioning formulation.

One possible approach for dealing with the existence of binary and integer variables in those models is to rely on the heuristic solution methods for CVRP and HVRP. Desrochers and Verhoog (1991) propose a new savings heuristic based on successive route fusion. Gheysens et al. (1986) developed another heuristic based on a lower-bound procedure. Numerous others have developed various heuristic methods to solve the problem of determining fleet size and the mix vehicle routing problems in the literature. There have also been attempts to apply metaheuristic methods too. Önder (2007) attempts using metaheuristics to solve HRVP without an investigation of the structure of the model but fails unfortunately. Evidently, the application of heuristics and even the metaheuristics requires a thorough investigation of the structure of the problem and the model.

The sweep algorithm described in Toth and Vigo (2002) is a heuristic for a possible solution method for VRP and some other problems. Nurchanyo et al. (2002) investigates the capability of the sweep algorithm in solving the VRP for public transport. After trying various approaches, they conclude that it is capable of solving VRP for public transport under certain conditions. Renaud and Boctor (2002) apply the sweep algorithm for the fleet size and the mix vehicle routing problem. Two kinds of decisions are considered: selecting a mix of vehicles and the routing of the selected fleet. Their algorithm generates a large number of routes to be serviced by one or two vehicles. Then the selection of the vehicles is accomplished by solving a set partitioning problem with a special structure.

The employee transporting problem is a variant of mix vehicle problem without routing. This article takes a different view from the literature and presents a set partitioning model for the employee transporting problem. The thrust of this article lies in the thorough investigation of the structure of the model. Subsequently, a surrogate problem is developed from this model which turns out to be a knapsack problem. The assignments of the vehicles are accomplished by matching subsets. For large-size problems, the sweep algorithm is proposed to obtain the assignments of the vehicles and consequently a partition in an efficient manner. The next section presents a set partitioning model, then the section 'A surrogate constraint' develops the surrogate problem. The solution methods are described in the section 'A solution method,' and some illustrations are presented in the section

'An illustration'. Finally, conclusions and findings of this article are summarized in the 'Conclusions' section.

A set partitioning model for employee transporting problem

Let A be the set of n destinations (bus stops) in an area, for instance, vicinity of a city. Each destination has a demand: the number of employees to be picked up and/or delivered. Let $s(a)$ denote the number of employees at destination $a \in A$. Naturally, $s(a) \geq 1$ and an integer. Let V be the set of the vehicles available with different capacities. Then $A_v \subset A$ represents a number of destinations to be serviced by the vehicle v of capacity Q_v. If m vehicles are required for servicing those n destinations, then each vehicle v will serve a subset A_v, and none of the subsets have any intersection. More specifically, a partition of the set A is obtained in the assignment of m vehicles. Let Ω be the class of all partitions of the set A. A partition $P = (A_1, A_2 \ldots, A_m)$ is a collection of subsets of A where $1 \leq m \leq n$, $A_i \cap A_j = \emptyset$, for $i \neq j$, and $\cup_{j=1}^m A_j = A$.

The cost per trip of assigning a vehicle $v \in V$ with capacity Q_v to a subset $A_j \in P \subset \Omega$ is c_v. This cost depends on the capacity of the vehicle only, and it is independent of the subset assigned. However, the total cost of the assignment depends on the partition $P \in \Omega$. Consequently, the decision of assigning a vehicle to a subset is not independent of the partition $P \in \Omega$. Therefore, the decision variable of assigning a vehicle to a subset is defined as follows:

$$y_{vj} = \begin{cases} 1 \text{ if the vehicle } v \in V \text{ is assigned to the subset } A_j \in P \\ 0 \text{ otherwise} \end{cases}$$

(1)

Thus, the employee transporting model is given by the following:

$$\min_{P \in \Omega} \sum_{j \in J^P} \sum_{v \in V} c_v y_{vj} \qquad (2)$$

subject to

$$y_{vj} \sum_{a \in A_j} s(a) \leq Q_v \qquad \text{for all } v \in V, A_j \in P, P \in \Omega \quad (3)$$

$$y_{vj} \delta(A_j) \leq T_{\max} \quad \text{for all } v \in V, A_j \in P, P \in \Omega \quad (4)$$

$$y_{vj} \in \{0, 1\} \qquad (5)$$

where J^P is the index set of the subsets in the partition $P \in \Omega$, $\delta(A_j)$ is the traveling distance (time) to the destinations in a subset A_j via some route for $j \in J^P$, and $T_{\max}$ is the maximum allowed total traveling distance (time) for any subset in any partition. The sum $s(A_j) = \sum_{a \in A_j} s(a)$ rep-

resents the total number of passengers to be transported in a subset A_j, and it reduces constraint 3 to $y_{vj}S(A_j) \leq Q_v$. This is a linear constraint. It simply expresses the fact that the total number of passengers to be transported from any subset cannot exceed the capacity of the vehicle.

The set of constraint 4 states that the traveling distance (time) to a subset $A_j \in P$ for any partition $P \in \Omega$ cannot exceed the maximum allowed distance (time). It is tacitly assumed that the vehicles start from the center and return back to the center after visiting every destination in their routes. Traveling distance (time) to a subset is independent of the vehicle; therefore, $\delta(A_j) \leq T_{\max}$ for all $A_j \in P$, $P \in \Omega$. This is not a linear constraint. The total distance (time) needs to be calculated for each subset A_j. Any route to the subset A_j is acceptable as long as the distance (time) does not exceed $T_{\max}$. This means that it is not necessary to solve the TSP to determine the least-distance (time) route for each subset.

The set of constraint 4 can also be treated as a set of soft constraints. The distance (time) traveled to any subset can be increased by a small amount ($\epsilon > 0$) to make the right-hand side of the constraint $T_{\max} + \epsilon$ at an added cost of opportunity loss of being late to work (school). Consequently, the objective function will be increased by $\eta(\delta(A_j))$ if the vehicle v traverses the subset A_j in $T_{\max} + \epsilon$. In this expression, $\eta(.) > 0$ is the cost of opportunity loss of being late by an amount of ϵ for the subset A_j. For a given value of ϵ, this problem is solved by setting the right-hand side of the set of constraint 4 to $T_{\max} + \epsilon$.

The objective function is then the total cost of all assignments of vehicles to all subsets of any partition. The problem is then to find a partition $P \in \Omega$ and a corresponding set of assignments of the vehicles to those subsets of this partition such that the total cost of such partition-vehicle assignments is minimum. This expression clearly defines the employee transporting problem as a heterogenous fleet mix problem without routing.

A surrogate constraint

Multiplying the set of constraint 3 by y_{vj} then summing the capacity constraints over $v \in V$ and $j \in J^P$ yields $\sum_{j \in J^P} \sum_{v \in V} y_{vj}^2 s(A_j) \leq \sum_{j \in J^P} \sum_{v \in V} y_{vj} Q_v$. The variables y_{vj} are binary variables, and thus, $y_{vj}^2 = y_{vj}$. On the other hand, every subset $A_j \in P$ must be assigned to a vehicle $v \in V$, but not every vehicle $v \in V$ may be assigned to a subset. Consequently, $\sum_{v \in V} y_{vj} = 1$ for all $j \in J^P$, and $\sum_{j \in J^P} y_{vj} \leq 1$ for all $v \in V$. In fact, a new variable can be defined now representing whether a vehicle is assigned or not as follows:

$$y_v = \sum_{j \in J^P} y_{vj} = \begin{cases} 1 & \text{if the vehicle } v \in V \text{ is assigned} \\ 0 & \text{otherwise} \end{cases}$$

(6)

Then a sequence of algebraic manipulations yields the right-hand side of the inequality 3 to be the total number of passengers to be transported from all the destinations.

$$\sum_{j \in J^P} \sum_{v \in V} y_{vj}^2 s(A_j) = \sum_{j \in J^P} \sum_{v \in V} y_{vj} s(A_j) \qquad (7)$$

$$= \sum_{j \in J^P} s(A_j) \sum_{v \in V} y_{vj} \qquad (8)$$

$$= \sum_{j \in J^P} s(A_j) = s(A) \qquad (9)$$

On the other side of the inequality, $\sum_{v \in V} Q_v \sum_{j \in J^P} y_{vj} = \sum_{v \in V} Q_v y_v$. Then constraint 3 becomes $\sum_{v \in V} Q_v y_v \geq S(A)$. A simple modification in the objective function gives $\sum_{v \in V} \sum_{j \in J^P} c_v y_{vj} = \sum_{v \in V} c_v \sum_{j \in J^P} y_{vj} = \sum_{v \in V} c_v y_v$. Consequently, a knapsack problem is obtained for a partition $P \in \Omega$ as follows:

$$\min \sum_{v \in V} c_v y_v \qquad (10)$$

subject to

$$\sum_{v \in V} Q_v y_v \geq s(A) \qquad (11)$$

where $y_v \in \{0, 1\}$ for all $v \in V$ and the additional set of constraints $\delta(A_j) \leq T_{\max}$ for all $j \in J^P$, $P \in \Omega$.

For any partition $P \in \Omega$, this surrogate problem needs to be solved. The Stirling number of the second kind ($\mathcal{S}_n^{(k)}$) gives the number of partitions with exactly k nonempty subsets of n elements. For instance, the number of partitions of 15 elements with exactly six nonempty subsets is $\mathcal{S}_{15}^{(6)} = 420{,}693{,}273$. Consequently, the total number of all partitions of a set of n elements is given by $\sum_{k=1}^{n} \mathcal{S}_n^{(k)}$. In fact, the total number of all partitions of a set of 15 elements is $\sum_{k=1}^{n} \mathcal{S}_{15}^{(k)} = 1{,}382{,}858{,}545$.

Let $P^* \in \Omega$ be an optimal partition with a corresponding vehicle assignment ($V^* = \{1, 2, \ldots, v_p\}$). Hence, $\sum_{v \in V^*} c_v y_v$ is the minimum cost and the total transporting capacity is given by $\sum_{v \in V^*} Q_v y_v = b \geq s(A)$. The right-hand side of the constraint can be greater than or equal to the total number of passengers ($s(A)$) to be carried due to the discreteness of the number of passengers at different locations and the traveling time constraint. For each subset $A_j \in P^*$, the distance (time) constraint $\delta(A_j) \leq T_{\max}$ holds.

The following knapsack problem is independent of any partition. If a partition (vehicle-subset assignment satisfying the capacity and the distance (time) constraints) can

be obtained for this solution, a solution to the original problem is obtained.

$$\min \sum_{v \in V} c_v y_v \tag{12}$$

subject to

$$\sum_{v \in V} Q_v y_v \geq r \tag{13}$$

$$y_v \in \{0, 1\} \qquad \text{for all } v \in V \tag{14}$$

Theorem 1. *Let $Y^*(r = b)$ be an optimal solution to the knapsack problem above. If there exists a partition (satisfying the capacity and the distance constraints) corresponding to $Y^*(b)$, then this solution is optimal to the problems (2), (3), and (4).*

Proof. Let $P* \in \Omega$ be an optimal partition and V^* be the corresponding vehicle assignment. This solution satisfies the surrogate problem necessarily. Clearly, $\sum_{v \in V*} c_v y_v$ is the minimum cost solution for the optimal partition $P*$, and further, $\sum_{v \in V*} c_v y_v \leq \sum_{v \in V} c_v y_v$ for any set V and any partition $P \in \Omega$. The knapsack problem with the right-hand side $s(A) \leq r < b$ can have a solution in $Y(r)$, but there cannot be any corresponding vehicle-subset assignments and/or partitions available satisfying the distance (time) constraint. If this were possible, then there would be a partition P' and V' such that $\sum_{v \in V'} c_v y_v < \sum_{v \in V*} c_v y_v$. This is a contradiction to the hypothesis. Therefore, a knapsack problem with $r = b$ needs to be considered.

Let $Y(r = b) = (y_{v1}, y_{v2}, \ldots, y_{vm})$ be a solution to this knapsack problem for some $1 \leq m \leq n$ together with $V(r) = \{v_1, v_2, \ldots, v_m\}$ corresponding to a vehicle-subset assignment $(A_1, A_2, \ldots, A_m)$ satisfying the distance (time) constraints. Then $\sum_{v \in V(r)} c_v y_v$ is the minimum cost satisfying the constraint $\sum_{v \in V(r)} Q_v y_v \geq b > s(A)$. Subsequently, $\sum_{v \in V(r)} c_v y_v \leq \sum_{v \in V*} c_v y_v$ implies that they must be equal. The sets V^* and $V(r)$ may not be equal because of the existence of alternating optimal vehicle-partition combinations. Thus, an optimal partition (vehicle-subset combination) exists for the solution $Y(r)$. $\qquad \square$

The argument until now assumes that there is only one type of each vehicle; hence, y_v is a binary variable, and the surrogate problem is a (0,1) knapsack problem. However, if the number of vehicles of each capacity is more than one, then the surrogate problem becomes a bounded integer knapsack problem. All the arguments up to now apply equally well to the case of bounded integer knapsack problem. In this form, the employee transporting problem is reduced to the common sense problem of determining the number of vehicles of each capacity to transport all the passengers at a minimum cost provided that the distance (time) of any route to a subset of destinations does not exceed a predetermined limit without any consideration of routing and vehicle-subset assignments.

A solution method

The knapsack problem can be solved by any method in the literature, see for instance Kellerer et al. (2004). It contains $|V|$ variables, binary or integer. Fortunately, in this class of problems, $|V|$ is a rather small number possibly not exceeding 10. This fact simplifies the numerical computations somehow while solving this knapsack problem. After obtaining a solution to the knapsack problem, a partition of the set A is sought, satisfying the capacity and the distance constraints. If such a partition is obtained, then an optimal solution to the original problem is found. Otherwise, the right-hand side value of the knapsack problem is increased to the next range, and the procedure is repeated. Before describing the algorithms, first, how to determine the range on the right-hand side of the knapsack problem is presented. The knapsack problem

$$\min z = \sum_{j=1}^{m} c_j y_j$$

subject to

$$\sum_{j=1}^{m} a_j y_j \geq b$$

where $y_j \geq 0$, and the integer for $j = 1, 2, \ldots, m$ can be converted to a maximization knapsack problem by setting $y_j = w_j - x_j$ where $w_j = \lceil \frac{b}{a_j} \rceil$ is the smallest integer greater than or equal to b/a_j for $j = 1, 2, \ldots, m$. The new knapsack problem is given as follows:

$$\max z' = \sum_{j=1}^{m} c_j x_j - \sum_{j=1}^{m} c_j w_j$$

subject to

$$\sum_{j=1}^{m} a_j x_j \leq \sum_{j=1}^{m} a_j w_j - b$$

where $0 \leq x_j \leq w_j$, and the integer for $j = 1, 2, \ldots, m$. Let $(x_1^*, x_2^*, \ldots, x_m^*)$ be an optimal solution to this knapsack problem. Then $\sum_{j=1}^{m} a_j x_j^* = b_* \leq \sum_{j=1}^{m} a_j w_j - b$. Consequently, the upper bound on r is $r_u = \sum_{j=1}^{m} a_j (w_j - x_j^*)$ and the lower bound on r is $r_\ell = b$.

Intuitively, having a sufficient number of vehicles will provide a partition (vehicle-subset assignment). Obviously, the algorithm will converge to a solution after a finite number iterations. A main algorithm to solve the knapsack problem consecutively and a subalgorithm to obtain a partition will be described next. Let $Y^t(r) = (y_{v1}^t, y_{v2}^t, \ldots, y_{vm}^t)$ represent the number of each type of vehicle in the knapsack solution in the iteration t.

The main algorithm is as follows:

Step 0. Initialize: Set $t = 1$, $r = s(A)$, go to Step 1.
Step 1. Solve the knapsack problem to obtain a solution $Y^t(r)$, and go to Step 2.
Step 2. Use the subalgorithm to obtain a partition corresponding to the solution $Y^t(r)$. If such a partition is obtained, go to Step 4, otherwise go to Step 3.
Step 3. Increase the right-hand side range to the next level for which r is the lower bound. Set $t := t + 1$ and go to Step 1.
Step 4. Terminate: An optimal partition (vehicle-subset assignment) is obtained.

The subalgorithm to find a partition corresponding to $Y^t(r)$ will determine all the possible subsets for each vehicle in the solution $Y^t(r)$, and subsequently, it will check whether a match is possible. The total capacity of the vehicles used in the solution $Y^t(r)$ is $\sum_{v \in V(r)} Q_v = r \geq s(A)$. Let $s_0 = s(A) - r$ be the slack representing the unused capacity of all the vehicles initially. Let R_v represent the set of not-yet-assigned destinations for the vehicle v. Initially, $R_1 = A$. P_v represents the incomplete partition for the vehicles $1, 2, \ldots, v$, and $P_0 = \emptyset$.

An exact algorithm to find a partition is as follows:

Step 0. Initialize: Set $v = 1$, $R_1 = A$, and $P_0 = \emptyset$. Go to Step 1.
Step 1. $L_1 = \{A_x | A_x \subset A, Q_1 - s_0 \leq s(A_x) \leq Q_1, \delta(A_x) \leq T_{\max}\}$. If $L_1 = \emptyset$, go to Step 5, otherwise go to Step 2.
Step 2. Let $A_v \in L_v$ be the first subset in this list. Set $P_v := P_{v-1} + A_v$, $s_v := s_{v-1} - (Q_v - s(A_v))$, and $v := v + 1$. Go to Step 3.
Step 3. If $(v > v_m)$, go to Step 5. Otherwise, $R_v := R_{v-1} - A_v$, and $L_v = \{A_x \subset R_v \mid Q_v - s_{v-1} \leq s(A_x) \leq Q_v\}$. If $L_v = \emptyset$, go to Step 4, otherwise go to Step 2.
Step 4. $v := v - 1$, $P_{v-1} := P_v - A_v$, $L_v := L_v - A_v$, $s_{v-1} := s_v + (Q_v - s(A_v))$, and $R_v := R_v + A_v$. If $v = 1$, go to Step 1, otherwise go to Step 2.
Step 5. Terminate by declaring that either a feasible partition is obtained if $R_v = \emptyset$ or there are no feasible partitions if $R_v \neq \emptyset$ or $L_1 = \emptyset$.

This algorithm finds a partition for the solution $Y^t(r)$ if it exists by checking all the possibilities. An example is provided to illustrate how this algorithm works. Unfortunately, it requires checking so many alternatives; it is not practical for solving large-size problems. Consequently, an adaptation of the sweep method (Toth and Vigo 2002) is proposed as an easier and efficient approach for finding the subset-vehicle assignments. This sweep algorithm is also a subalgorithm to

be called by the main algorithm in Step 2 to obtain a partition.

The sweep algorithm is as follows:

Step 0. Initialize: Convert the Cartesian coordinates to the polar coordinates for each destination $a_j(x, y) \rightarrow a_j(r, \theta)$ for $j = 1, 2, \ldots, n$. The destinations are called nodes from now on. Set $v := 1$. Start with the vehicle v in the solution $Y^t(r)$ with the capacity Q_v. Go to Step 1.
Step 1. Set $B := \emptyset$, $A_v := \emptyset$, $s(A_v) = 0$, $k = 0$.
Step 1a. Start rotating from the x-axis and find a node a_{vk} such that $\min_\theta a_j(r, \theta) = a_{vk}(r, \theta)$ where $\theta_k \leq \theta_j$ for all $a_j \in A$. If a_{vk} exists, then go to Step 1b, otherwise go to Step 3.
Step 1b. If $s(A_v) + s(a_{vk}) \leq Q_v$, then go to Step 1c, otherwise go to Step 1e.
Step 1c. Calculate $\delta(A_v + \{a_{vk}\}) = \delta$. If $\delta \leq T_{\max}$, then go to Step 1d, otherwise go to Step 1e.
Step 1d. $A_v := A_v + \{a_{vk}\}$, and $A := A - \{a_{vk}\}$. Set $k := k + 1$ and go to Step 1a.
Step 1e. Set $B := B + \{a_{vk}\}$ and $A := A - \{a_{vk}\}$. If $A = \emptyset$, go to Step 2, otherwise go to Step 1a.
Step 2. Set $A := A \cup B$, and $v := v + 1$. If $v \leq v_m$, then go to Step 1, otherwise go to Step 3.
Step 3. Terminate by declaring either that a partition is obtained if $A = \emptyset$ or a partition is not obtained if $A \neq \emptyset$.

An illustration

An implementation of the main algorithm and the subalgorithms is essential to observe how the employee transporting problem can be solved to optimality in an efficient manner. A small example with ten destinations and 91 passengers to be transported is used in testing the knapsack solution and the exact algorithm. A larger-size example with 79 destinations and 694 passengers is used to implement the sweep method. Both examples are randomly generated.

A small town with ten destinations is considered. The coordinates and the number of passengers to be transported from/to each destination is given in Table 1. The distance matrix is given in Table 2. There are three types of vehicles available for transportation: mini-busses with 15 seats, midi-busses with 30 seats, and coaches with 50 seats with costs of 35, 55, and 105 TL per trip, respectively. There are altogether 91 passengers to be transported. How many vehicles of each type are needed and to which subsets will they be assigned to accomplish the task at a minimum cost?

The initial knapsack problem to be solved is given as follows:

$$\min z = 35y_1 + 55y_2 + 105y_3$$

Table 1 The coordinates and the number of passengers

a	x_a	y_a	r_a	θ_a	$s(a)$
1	0.90	7.10	7.16	0.13	10
2	0.70	10.20	10.22	0.07	8
3	1.50	12.20	12.29	0.12	13
4	0.20	10.90	10.90	0.02	6
5	0.60	14.30	14.31	0.04	6
6	2.20	14.50	14.67	0.15	4
7	0.10	14.60	14.60	0.01	10
8	0.50	16.50	16.51	0.03	11
9	1.80	18.00	18.09	0.10	8
10	2.50	15.80	16.00	0.16	15

subject to

$$15y_1 + 30y_2 + 50y_3 \geq 91$$

where $y_i \geq 0$, and the integer for $i = 1, 2, 3$. In this problem, y_1 is the number of mini-busses, y_2 is the number of midi-busses, and y_3 is the number of coaches to employ. The solution to this knapsack problem yields $Y^1 = (1, 1, 1)$ with a cost of 195 TL. The following argument shows that this solution is valid for $91 \leq r \leq 95$. A simple computation shows $w_1 = 7$, $w_2 = 4$, and $w_3 = 2$. The transformation $y_j = w_j - x_j$ for $j = 1, 2, 3$ gives the following knapsack problem:

$$\max z' = 35x_1 + 55x_2 + 105x_3 - 675$$

subject to

$$15x_1 + 30x_2 + 50x_3 \leq 325 - 91 = 234$$

where $0 \leq x_1 \leq 7$, $0 \leq x_2 \leq 4$, and $0 \leq x_3 \leq 2$ are all integers. The optimal solution to this knapsack is $X = (6, 3, 1)$ and $Y = (1, 1, 1)$ with $z' = 480 - 675 = -195 = -z$ and $15x_1 + 30x_2 + 50x_3 = 230$. Therefore, $r_u = $

Table 2 The distances between the destinations

a	0	1	2	3	4	5	6	7	8	9	10
0	-	75	103	126	111	144	152	146	167	185	166
10	166	91	67	42	68	40	14	47	40	26	
9	185	110	82	59	79	45	35	47	30		
8	167	94	64	48	57	23	40	20			
7	146	75	45	35	38	8	40				
6	152	78	53	27	55	32					
5	144	71	41	28	35						
4	111	40	10	30							
3	126	52	26								
2	103	30									
1	75										

$325 - 230 = 95$ and $r_\ell = 91$. Obviously, if the right-hand side of this knapsack is less than 230, a new solution is required.

The next range is $96 \leq r \leq 105$ with an optimal solution $Y^2 = (1, 3, 0)$ with a cost of 200 TL. Further, the next range is $106 \leq r \leq 110$ with an optimal solution of $Y^3 = (0, 2, 1)$ with a cost of 215 TL. $T_{\max} = 370$ distance (time) units for this town. First, the exact algorithm is implemented for finding a partition corresponding to the solution $Y^t(r)$. There are no partitions for the solution $Y^1(91) = (1, 1, 1)$ with $T_{\max} = 370$. Then the next best solution (next range) is attempted. $Y^2(96) = (1, 3, 0)$. The total capacity of the vehicles now is 105. Therefore, $s_0 = 105 - 91 = 14$. The list L_1 of all subsets satisfying the condition $30 - 14 \leq s(A_x) \leq 30$ and the distance constraint has 118 subsets of A. First, the subset $A_1 = \{4, 7, 8\}$ with $s(A_1) = 27$ and $\delta(A_1) = 336$ is chosen. The set of remaining destinations is $\{1, 2, 3, 5, 6, 9, 10\}$ for the vehicle $v = 2$ with $Q_2 = 30$. Now $s_1 = 14 - (30 - 27) = 11$. The list L_2 of all subsets satisfying the condition $30 - 11 = 19 \leq s(A_x) \leq 30$ contains 22 subsets. First one in this list, $A_2 = \{6, 10\}$ with $s(A_2) = 19$ and $\delta(A_2) = 332$ yields $s_2 = 11 - (30 - 19) = 0$, and there are no subsets for $v = 3$ satisfying the condition $30 \leq s(A_x) \leq 30$. Subsequently, the subset $A_2 = \{6, 10\}$ is deleted from L_2. Next, the subset $A_2 = \{3, 9\}$ is tested. $s(A_2) = 21$ and $\delta(A_2) = 370$. Now $s_2 = 11 - (30 - 21) = 2$. The set of remaining destinations now is $\{1, 2, 5, 6, 10\}$. The set L_3 of all subsets satisfying the condition $30 - 2 \leq s(A_x) \leq 30$ and $\delta(A_x) \leq 370$ for the vehicle $v = 3$ with $Q_3 = 30$ contains three elements only. Taking $A_3 = \{2, 5, 10\}$ with $s(A_3) = 29$ and $\delta(A_3) = 350$ yields $L_4 = A_4 = \{1, 6\}$ with $s(A_4) = 14$ and $\delta(A_4) = 305$ for the vehicle $v = 4$ with $Q_v = 15$. Hence, an optimal partition is obtained. There are also alternating optimal partitions: $A_3 = \{1, 2, 5, 6\}$ with $s(A_3) = 28$ and $\delta(A_3) = 330$ for $Q_3 = 30$ and $A_4 = \{10\}$ with $s(A_4) = 15$ and $\delta(A_4) = 332$ for $Q_4 = 15$ with a cost of 200 TL.

Solving the same problem by the sweep algorithm yields $A_1 = \{2, 4, 5, 7, 8\}$ with $s(A_1) = 41$, $\delta(A_1) = 346$ for the vehicle of the capacity $Q_1 = 50$; $A_2 = \{3, 9\}$ with $s(A_2) = 21$, $\delta(A_2) = 370$ for the vehicle of the capacity $Q_2 = 30$; and $A_3 = \{1, 6, 10\}$ with $s(A_3) = 29$, $\delta(A_3) = 333$ for the vehicle of capacity $Q_3 = 30$ corresponding to the knapsack solution $Y^3 = (0, 2, 1)$ with a cost of 215 TL and $T_{\max} = 370$. This is a good example that the sweep algorithm may miss the optimal partition.

Treating the constraint $\delta(A_x) \leq T_{\max}$ as a soft constraint is also considered. $T_{\max} = 370$ distance time units in this example. Now it is increased by 1% to 373.7 at an additional cost of opportunity loss of 2% of the total cost of the solution. Then $Y^1 = (1, 1, 1)$ yields the partition $A_1 = \{1, 4, 5, 6, 7, 8\}$ with $s(A_1) = 47$, $\delta(A_1) = 370$; $A_2 = \{2, 3, 9\}$ with $s(A_2) = 29$, $\delta(A_2) = 373$; and $A_3 = $

Table 3 The locations and the number of passengers to be transported

a	y_a	x_a	r_a	θ_a	$s(a)$	a	y_a	x_a	r_a	θ_a	$s(a)$
1	0.50	1.50	1.58	1.25	7	41	12.20	1.50	12.29	0.12	13
2	1.00	2.40	2.60	1.18	8	42	10.80	2.60	11.11	0.24	14
3	1.80	3.30	3.76	1.07	9	43	10.90	0.20	10.90	0.02	6
4	2.60	4.40	5.11	1.04	9	44	13.40	11.40	17.59	0.70	5
5	3.50	3.50	4.95	0.79	6	45	13.50	7.50	15.44	0.51	5
6	5.00	3.60	6.16	0.62	13	46	13.40	5.80	14.60	0.41	8
7	4.70	4.60	6.58	0.77	3	47	14.00	5.50	15.04	0.37	12
8	4.40	5.60	7.12	0.90	11	48	19.40	4.30	19.87	0.22	4
9	4.60	8.10	9.32	1.05	7	49	13.90	3.50	14.33	0.25	7
10	5.70	5.80	8.13	0.79	8	50	13.10	2.50	13.34	0.19	3
11	5.50	9.00	10.55	1.02	8	51	14.30	0.60	14.31	0.04	6
12	6.60	7.50	9.99	0.85	2	52	14.80	12.50	19.37	0.70	6
13	7.00	2.50	7.43	0.34	8	53	14.20	11.00	17.96	0.66	6
14	7.10	0.90	7.16	0.13	10	54	14.40	9.80	17.42	0.60	3
15	8.10	11.00	13.66	0.94	6	55	14.50	8.10	16.61	0.51	8
16	8.00	8.50	11.67	0.82	2	56	14.50	2.20	14.67	0.15	4
17	7.90	6.20	10.04	0.67	6	57	14.60	0.10	14.60	0.01	10
18	7.80	2.60	8.22	0.32	10	58	15.90	11.10	19.39	0.61	9
19	8.00	1.40	8.12	0.17	11	59	15.80	10.50	18.97	0.59	13
20	9.00	1.00	9.06	0.11	8	60	15.10	9.10	17.63	0.54	11
21	8.70	7.00	11.17	0.68	13	61	15.50	7.00	17.01	0.42	10
22	9.20	2.50	9.53	0.27	6	62	15.10	4.30	15.70	0.28	10
23	10.50	10.40	14.78	0.78	8	63	15.80	2.50	16.00	0.16	15
24	10.20	8.30	13.15	0.68	5	64	16.10	13.00	20.69	0.68	15
25	10.30	7.30	12.62	0.62	13	65	16.70	12.40	20.80	0.64	10
26	10.00	6.40	11.87	0.57	12	66	16.20	9.70	18.88	0.54	10
27	10.10	4.50	11.06	0.42	14	67	16.80	8.80	18.97	0.48	10
28	10.00	2.40	10.28	0.24	13	68	16.20	8.10	18.11	0.46	5
29	10.20	0.70	10.22	0.07	8	69	16.10	6.20	17.25	0.37	3
30	11.70	11.50	16.41	0.78	12	70	16.60	5.30	17.43	0.31	3
31	11.50	9.60	14.98	0.70	11	71	16.50	3.30	16.83	0.20	11
32	11.40	6.70	13.22	0.53	15	72	16.50	0.50	16.51	0.03	11
33	11.50	3.50	12.02	0.30	13	73	18.00	11.20	21.20	0.56	5
34	12.70	11.00	16.80	0.71	13	74	18.00	7.50	19.50	0.39	13
35	12.60	9.70	15.90	0.66	8	75	18.00	5.00	18.68	0.27	12
36	12.50	8.30	15.00	0.59	7	76	17.80	3.10	18.07	0.17	12
37	13.00	6.50	14.53	0.46	8	77	18.00	1.80	18.09	0.10	8
38	12.60	3.50	13.08	0.27	6	78	18.50	11.40	21.73	0.55	8
39	12.00	3.30	12.45	0.27	14	79	19.00	9.00	21.02	0.44	11
40	12.10	2.20	12.30	0.18	8		Total				694

{10} with $s(A_3) = 15$, $\delta(A_3) = 332$ at a total cost of $195 + 2\%(195) = 198.90$ TL. This is another alternative to consider for the fleet mix problem.

Now a larger-size problem shows how efficient the sweep algorithm is in obtaining a partition if it exists or not for the given fleet of vehicles. This example has 79

destinations located in a metropolitan area, and the total distance to be traveled should not exceed 50 km. Table 3 provides the locations in (x, y) and (r, θ) polar coordinates and the number of passengers to be transported from those 79 destinations to the center in the morning and back later in the evening in this metropolitan area. There are 694 passengers to be transported; therefore, the knapsack problem to be solved is given as follows:

$$\min \ z = 35y_1 + 55y_2 + 105y_3$$

subject to

$$15y_1 + 30y_2 + 50y_3 \geq 694$$

where $y_i \geq 0$ and the integer for $i = 1, 2, 3$. The solution is $Y^1 = (1, 21, 1)$ with a cost of 1,295 in the range $694 \leq r \leq 695$. The next best solution is $Y^2 = (1, 23, 0)$ with a cost of 1,300 in the range $696 \leq r \leq 710$. Further, the next best solution is $Y^3 = (0, 22, 1)$ with a cost of 1,315 in the range $711 \leq r \leq 720$. Finally, the next solution is $Y^4 = (2, 23, 0)$ with a cost of 1,335 in the range $721 \leq r \leq 725$. The sweep algorithm does not obtain any partitions for the solutions $Y^1 = (1, 21, 1)$, $Y^2 = (1, 23, 0)$, and $Y^3 = (0, 22, 1)$, but finds a partition for the solution $Y^4 = (2, 23, 0)$ given in Table 4. Therefore, the relative error is less than $(1, 335 - 1, 295)/1, 295 = 3.09\%$. As an indication of the efficiency of the sweep algorithm, one should point out that a small laptop computer (netbook) with a processor chip of 1.60 GHz solves the last problem in 1.73 s of CPU time.

Conclusions

The employee transporting problem is actually a set partitioning problem for the assignment of the vehicles. The capacities of the vehicles vary in size. Therefore, it is a heterogenous fleet problem. These facts make the employee transporting problem belong to a different class of problems than the HVRP in the literature. Although in both problems the aim is to determine the fleet mix and composition, the employee transporting problem strives for this aim without routing. A set partitioning model is developed, deviating from the integer formulations in the literature. An important aspect of this article is the thorough investigation of the model structure. Consequently, this investigation leads to a knapsack problem from the model structure. This knapsack problem is the common sense problem of determining the number of vehicles of each capacity to transport all the passengers at a minimum cost by ignoring the subset-vehicle assignments.

The partition obtained for the knapsack solution is an optimal solution. There is an exact algorithm to find the partition for a given knapsack solution. However, it requires matching too many subsets; it is impractical for large-size problems. The sweep algorithm is suitable for large-size problems. It is a heuristic method which obtains a partition in a very efficient manner computationally

Table 4 Solution of the large-size problem for $Y = (2, 23, 0)$

	v	$s(A_v)$	Q_v	$\delta(A_v)$	A_v
Vehicle	1	30	30	41.98	Part: 50 57 43 72
Vehicle	2	30	30	42.78	Part: 51 56 77 48 29
Vehicle	3	30	30	29.35	Part: 41 49 14
Vehicle	4	30	30	38.02	Part: 63 76 70
Vehicle	5	30	30	33.66	Part: 19 40 71
Vehicle	6	30	30	34.77	Part: 28 42 69
Vehicle	7	29	30	39.96	Part: 22 39 38 54
Vehicle	8	30	30	37.40	Part: 13 62 75
Vehicle	9	30	30	48.37	Part: 16 18 33 68
Vehicle	10	30	30	41.10	Part: 47 45 74
Vehicle	11	30	30	29.94	Part: 37 46 27
Vehicle	12	29	30	43.15	Part: 55 61 79
Vehicle	13	30	30	42.86	Part: 67 73 32
Vehicle	14	29	30	45.97	Part: 66 60 78
Vehicle	15	30	30	43.31	Part: 12 26 36 58
Vehicle	16	29	30	38.13	Part: 7 25 59
Vehicle	17	29	30	41.63	Part: 6 53 65
Vehicle	18	27	30	34.06	Part: 21 17 35
Vehicle	19	26	30	41.46	Part: 52 64 24
Vehicle	20	29	30	35.22	Part: 31 34 44
Vehicle	21	28	30	32.81	Part: 20 23 30
Vehicle	22	25	30	16.68	Part: 5 8 10
Vehicle	23	30	30	27.69	Part: 4 9 15 11
Vehicle	24	9	15	7.52	Part: 3
Vehicle	25	15	15	5.21	Part: 2 1
Total		694	720		

but may miss the optimal partition. A few examples are provided to illustrate how the main algorithm and each subalgorithm works. It is worth mentioning at this point that other heuristics and/or metaheuristics can be implemented in obtaining the partition after deciding the mix of the fleet. The employee transporting problem differs in nature from HVRP by not considering the routing, thus a computational comparison with other methods in the literature will not serve any purpose and such a task was not undertaken.

The model for the employee transporting problem and its solution can provide useful insights to practitioners in order to decide the mix of vehicles in the fleet and subsequently the fleet size to transport the passengers from/to the destinations. Furthermore, vehicle drivers are human beings; hence, they may have preferences and/or restrictions where to go. The approach of this article allows including the preferences and/or restrictions of the drivers in the choice of destination subsets.

References

Baldacci R, Mingozzi A, Roberti R (2011a) New route relaxation and pricing strategies for the vehicle routing problem. Oper Res 59: 1269–1283

Baldacci R, Bartolini E, Mingozzi A (2011b) An exact algorithm for the pickup and delivery problem with time windows. Oper Res 59: 414–426

Bruce GL, Raghavan S, Wasil EA (eds) (2008) The vehicle routing problem: latest advances and new challenges. Operations research/computer science interface series, vol. 43. Springer, Heidelberg

Desrochers M, Verhoog TW (1991) A new heuristic for the fleet size and the mix vehicle routing problem. Comput Oper Res 18: 263–274

Gheysens F, Golden B, Assad A (1986) A new heuristic for determining the fleet size and composition. Math Programming Study 26: 233–236

Kellerer H, Pferschy U, Pisinger D (2004) Knapsack problems. Springer, Heidelberg

Miller C, Tucker A, Zemlin R (1960) Integer programming formulations and traveling salesman problem. J ACM 7: 326–329

Nurchanyo GW, Alias A, Shamsuddin SMM, MNMd Sap (2002) Sweep algorithm in vehicle routing problem for public transport. J Antarabanga (Teknologi Maklumat) 2: 51–64

Önder I (2007) An employee transporting problem and its heuristic solution. M.S Thesis. Çankaya University, Ankara

Renaud J, Boctor FF (2002) A sweep based algorithm for the fleet size and the mix vehicle routing problem. Eur J Oper Res 140: 618–628

Toth P, Vigo D (2002) The vehicle routing problem. Monographs on discrete mathematics and applications, Vol. 9. SIAM, Philadelphia

Yaman H (2006) Formulations and valid inequalities for the heterogeneous vehicle routing problem. Math Programming A 106: 365–390