Mathematical Programming 72 (1996) 83–100

On the nucleolus of the basic vehicle routing game [1]

Maud Göthe-Lundgren [a,*], Kurt Jörnsten [b], Peter Värbrand [a]

[a] *Department of Mathematics, Division of Optimization, Linköping Institute of Technology, S-581 83 Linköping, Sweden*

[b] *Norwegian Business School, Helleveien 30, N-5035 Bergen-Sandviken, Norway*

Received 15 June 1992; revised manuscript received 13 March 1995

Abstract

In the vehicle routing cost allocation problem the aim is to find a good cost allocation method, i.e., a method that according to specified criteria allocates the cost of an optimal route configuration among the customers. We formulate this problem as a co-operative game in characteristic function form and give conditions for when the core of the vehicle routing game is nonempty.

One specific solution concept to the cost allocation problem is the nucleolus, which minimizes maximum discontent among the players in a co-operative game. The class of games we study is such that the values of the characteristic function are obtained from the solution of a set of mathematical programming problems. We do not require an explicit description of the characteristic function for all coalitions. Instead, by applying a constraint generation approach, we evaluate information about the function only when it is needed for the computation of the nucleolus.

Keywords: Cost allocation; Co-operative game theory; Vehicle routing problem; Constraint generation

1. Introduction

In the vehicle routing problem (VRP) we consider a set of customers, each with a specific demand, and a central depot from which the customers are supplied by a

* Corresponding author. e-mail: mabre@math.liu.se.

[1] This research has been financed by grants from the Swedish Transportation Research Board (TFB), Dnr 92-97-43. It was performed in part while the first author was a guest researcher at GERAD, École des Hautes Études Commerciales in Montréal, Canada.

SSDI 0025-5610(95)00039-9

number of vehicles. The vehicles have fixed capacities and it is assumed that each vehicle travels along one route at the most. A route is defined as a path that starts at the central depot, passes at least one customer and returns to the depot. If all vehicle capacities are equal, the vehicle fleet is said to be homogen, otherwise it is inhomogen. The basic VRP is the problem of determining a route configuration that minimizes the total transportation cost, given that the demand of each customer is satisfied, no vehicle capacity is exceeded and that each customer is visited by exactly one vehicle. Further, it is assumed that the vehicle fleet is homogen. The VRP is an extensively studied problem (see e.g., [3,10]) and it is a well-known fact that it belongs to the class of NP-hard problems.

The vehicle routing cost allocation problem is the problem of finding, by some means, a good cost allocation method for the VRP. By means of the cost allocation method we specify the criteria of how to allocate the total cost of the route configuration among the customers. The problem can occur when different companies together own or hire transportation services and want to share the total cost. We formulate this problem as a co-operative game in characteristic function form. We are concerned with the computation of one specific solution concept of the basic VRP cost allocation problem: the nucleolus [16]. The nucleolus is a cost allocation that minimizes maximum discontent among the players in a co-operative game. In the class of games we study, the values of the characteristic function are obtained from the solution of a set of mathematical programming problems. We do not require an explicit description of the characteristic function for all coalitions. Instead, by applying a constraint generation approach, we evaluate information about the function only when it is needed for the computation of the nucleolus.

We are not aware of any published results concerning the basic VRP cost allocation problem. However, cost allocation for the well known traveling salesman problem (TSP), which is a relaxation of VRP, has been studied by Tamir [17] and Dror [7]. In [17], where the TSP with a home city is considered, results concerning emptiness/non-emptiness of the core are presented for specific noncomplete graphs and subgraphs. Dror [7] shows that the core of the TSP game which includes no home city is empty, and he designs schemes for allocating costs among the nodes in the TSP with a home city.

The outline of this paper is the following. In Section 2, a game theory formulation of the basic VRP cost allocation problem is presented. It is shown that the number of inequalities defining the core can be reduced, possibly by a large number. Section 3 includes some further results concerning the core and the nucleolus. It is shown that the core of the basic VRP game can be empty and conditions are established for when this is the case. In Section 4, a constraint generation scheme for the computation of the nucleolus is presented and Section 5 contains a mathematical formulation of the subproblem of the constraint generation scheme and an implicit enumeration procedure for solving the subproblem. Finally, in Section 6, numerical examples are presented and in Section 7 computational experiences are reported and some conclusions are given.

2. A game theory formulation of the basic VRP

Consider a customer area and a central depot from which the customers are supplied. We denote the demand of customer i by d_i and the capacity of a vehicle by Q. The set of customers is denoted by N and the set of vehicles by M. If a vehicle travels directly from customer i to customer j, $i, j \in N \cup \{0\}$ ($0 =$ depot), the transportation cost c_{ij} ($= c_{ji}$) occurs and it is assumed that the transportation cost matrix, $\{c_{ij}\}$, satisfies the triangel inequality, i.e., $c_{ij} \leqslant c_{ik} + c_{kj}$, $i, j, k \in N \cup \{0\}$. Further we assume that $d_i \leqslant Q$, $i \in N$, and that $\sum_{i \in N} d_i > Q$. The latter assumption implies that at least two routes are used in any feasible solution to the basic VRP.

In order to give a mathematical programming formulation of the basic VRP, consider the subsets of customers for which the total demand does not exceed the vehicle capacity. Assume that, for each such customer subset, the minimum cost route is known. Denote the cost of such a route by c_r and the set of the minimum cost routes by R. Furthermore, let $a_{ir} = 1$ if customer i is covered by route r and 0 otherwise. The formulation of the basic VRP is then

$$
\begin{aligned}
(\mathrm{VRP}) \quad z = \quad & \min \quad \sum_{r \in R} c_r x_r \\
& \text{s.t.} \quad \sum_{r \in R} a_{ir} x_r = 1, && i \in N, && (1.1) \\
& \qquad\qquad x_r \geqslant 0, && r \in R, && (1.2) \\
& \qquad\qquad x_r \text{ integer}, && r \in R, && (1.3)
\end{aligned}
$$

where $x_r = 1$ if route r chosen and 0 otherwise. The constraints (1.1) assure that each customer is visited exactly once. Observe that for a specified vector a_{ir} the objective function coefficient c_r is evaluated by identifying the minimum cost route covering the customers for which $a_{ir} = 1$, i.e. by the solution of a traveling salesman problem, which belongs to the class of NP-hard problems. Further, in this formulation of the basic VRP there is no limitation on the number of vehicles used.

From a practical point of view this formulation – due to its huge number of columns — is mainly interesting in a column generation setting. Originally, this approach was presented by Balinski and Quandt [2], who focused only on a subset of all feasible routes, where the feasibility of a route was given in terms of a restricted number of deliveries on a route. The most recent and very successful adoption of this formulation is presented in [5], where the VRP with time windows is considered. Here, new columns are generated by a dynamic programming model which solves a relaxed version of the shortest path problem with time windows.

In the terminology of game theory, we denote each customer, $i \in N$, by a *player* and each subset of customers, $S \subseteq N$, by a *coalition*. The *characteristic cost function* $c(S)$ of the basic VRP game $(N; c)$ is interpreted as the total transportation cost occuring if coalition $S \subseteq N$ is formed. A coalition is described by the binary vector s by

$$
s_i = \begin{cases} 1, & \text{if customer } i \text{ is a member of the coalition,} \\ 0, & \text{otherwise,} \end{cases} \qquad i \in N.
$$

We define c as the objective function value of a specific mathematical programming problem. For all coalitions $S \subseteq N$, $S \neq \emptyset$, $c(S)$ is defined as

$$\begin{aligned} c(S) = \quad \min \quad & \sum_{r \in R} c_r x_r \\ \text{s.t.} \quad & \sum_{r \in R} a_{ir} x_r = s_i, && i \in N, \\ & x_r \geqslant 0, && r \in R, \\ & x_r \text{ integer}, && r \in R. \end{aligned} \tag{2}$$

If $S = \emptyset$, then $c(S)$ takes the value 0. That is, $c(S)$ is the cost of an optimal route configuration covering the customers in S, i.e., the customers for which $s_i = 1$. Moreover, we have that $c(N) = z$. Under the assumption that the cost matrix $\{c_{ij}\}$ satisfies the triangle inequality, the characteristic cost function is monotone, i.e., $c(S) \leqslant c(T)$, $S \subset T \subset N$. In addition, we know that the basic VRP game is proper, i.e. the characteristic cost function is subadditive ($c(S) + c(T) \geqslant c(S \cup T)$, $S, T \subset N$, $S \cap T = \emptyset$).

For later use, denote a coalition S such that $\sum_{i \in S} d_i \leqslant Q$ by a *feasible coalition* and otherwise by an *infeasible coalition*. Thus, a feasible coalition is a feasible subset of customers. Let $\mathcal{S}$ be the set of feasible coalitions. Then we have that

$$c(S) = c_r, \quad \text{for all } r \in R \text{ and } S \in \mathcal{S} \text{ such that } a_{ir} = s_i,\ i \in N.$$

Therefore, if in (2) S is a feasible coalition, the evaluation of $c(S)$ reduces to the solution of a traveling salesman problem over the customers for which $s_i = 1$.

Denote the cost to be allocated to customer i by y_i, $i \in N$. The *core* C of the game $(N; c)$ is the set of cost allocations y, which satisfy

$$\sum_{i \in N} y_i = c(N), \tag{3.1}$$

$$\sum_{i \in S} y_i \leqslant c(S), \quad S \subset N. \tag{3.2}$$

Due to the monotonicity of the characteristic cost function c it follows that each vector $y \in C$ is nonnegative since $y_i = c(N) - \sum_{j \in N \setminus \{i\}} y_j \geqslant c(N) - c(N \setminus \{i\}) \geqslant 0$, $i \in N$. The interpretation of system (3) is that no single customer or coalition of customers should be charged more than their "stand-alone" cost $c(S)$. A cost allocation which is in the core is a fair allocation in the sense that, each customer or coalition of customers have an incentive to agree to the proposed allocation. We denote an inequality of type (3.2) by a *core defining inequality* (CDI).

When we study the core of a basic VRP game it is of great interest to — if possible — reduce the number of CDIs. In the following mathematical statement we show that this is possible.

Proposition 1. *Any CDI that sums over an infeasible coalition S, $S \neq N$, is redundant in* (3).

Proof. Consider any infeasible coalition $\hat{S}$, $\hat{S} \neq N$, and a corresponding optimal route configuration with routes $\{r_1, \ldots, r_m\}$. Denote by $\{S_1, \ldots, S_m\}$ the feasible disjoint coalitions corresponding to the optimal routes. Since $\sum_{j=1}^{m} \sum_{i \in S_j} y_i = \sum_{i \in \hat{S}} y_i$ and $\sum_{j=1}^{m} c(S_j) = c(\hat{S})$ it follows that $\sum_{i \in S_j} y_i \leqslant c(S_j)$, $j = 1, \ldots, m$, $\Rightarrow \sum_{i \in \hat{S}} y_i \leqslant c(\hat{S})$. $\square$

Thus,

$$C = \left\{ y \,\middle|\, \sum_{i \in S} y_i \leqslant c(S),\ S \in \mathcal{S};\ \sum_{i \in N} y_i = c(N) \right\}.$$

Due to Proposition 1, the only values of c that are of interest are those corresponding to feasible coalitions. Recall that the evaluation of $c(S)$ for a given coalition S in this case reduces to the solution of a traveling salesman problem over the customers for which $s_i = 1$.

3. The core and the nucleolus of the basic VRP game

In order to define the nucleolus of the basic VRP game, we first introduce the *excess* of S with respect to y as $c(S) - \sum_{i \in S} y_i$. The excess of S with respect to y "reflects the attitude of coalition S to the ... vector y" [16]. In our context this quote can be interpreted as, given a cost allocation vector $y \in Y = \{y \mid \sum_{i \in N} y_i = c(N)\}$, the coalition with the smallest excess is the one that disagrees most with the proposed allocation. If $\min_{S \in \mathcal{S}}\{c(S) - \sum_{i \in S} y_i^1\} > \min_{S \in \mathcal{S}}\{c(S) - \sum_{i \in S} y_i^2\}$, the vector y^1 is more acceptable than y^2. If there is an equality between the minima, we compare the next smallest excess with respect to y^1 and y^2, respectively.

For a vector $y \in Y$ let $\theta(y)$ be the *excess vector* in $R^{|\mathcal{S}|}$, with elements $c(S) - \sum_{i \in S} y_i$ arranged in increasing order, i.e., $k < l$ implies that $\theta_k(y) \leqslant \theta_l(y)$. Then, the *nucleolus* is the unique vector $y \in \bar{Y} = \{y \mid \sum_{i \in N} y_i = c(N),\ y_i \leqslant c(\{i\}),\ i \in N\}$ that has the lexicographically greatest associated excess vector, i.e., $\theta(y) \succ \theta(y')$, $y' \in \bar{Y} \setminus \{y\}$. If the core is nonempty, the nucleolus is, intuitively, the "centre" of the core. The *pre-nucleolus* is the unique vector $y \in Y$ that has the lexicographically greatest associated excess vector. Since the definitions of the nucleolus and the pre-nucleolus are closely related to the concept of the core, i.e. to the CDIs, it is of interest to investigate the question of non-emptiness of the core further. The example in Fig. 1 illustrates the fact that the core of the basic VRP game is not guaranteed to be nonempty.

The location of three customers and a depot, denoted by 0, is shown in Fig. 1. The transportation costs are also given in the figure. Each customer has a demand of one unit and there are three vehicles with a capacity of two units respectively, available. The characteristic cost function values for the singleton coalitions are $c(\{1\}) = c(\{2\}) = c(\{3\}) = 2$. Further, $c(\{1, 2\}) = c(\{1, 3\}) = c(\{2, 3\}) = 3.7$ and for the grand coalition we have that $c(\{1, 2, 3\}) = 5.7$ (e.g., the routes 0–1–2–0 and 0–3–0). Here, $C = \emptyset$ and the nucleolus is $y = (1.9, 1.9, 1.9)$ due to the symmetrical roles of the three customers. It is possible to obtain a game with a nonempty core by increasing the demand of any one of

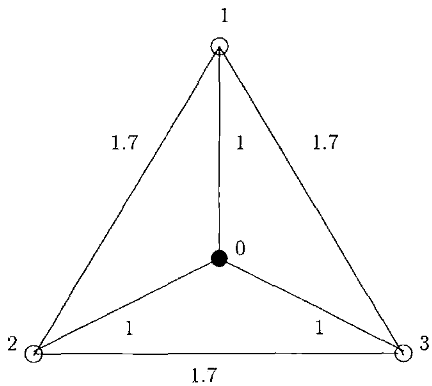

Fig. 1. 3-customer example.

the customers from one to two, e.g., customer 1 demands two units instead of one. The characteristic cost function values for the singleton coalition and for the grand coalition remain the same, but for the 2-coalitions we now have $c(\{1, 2\}) = c(\{1, 3\}) = 4$, $c(\{2, 3\}) = 3.7$. The optimal route configuration is 0–1–0 and 0–2–3–0. The core for this basic VRP game is nonempty.

Observe that, in the first case, there is an incentive for co-operation in the sense that, in an optimal route configuration two customers must be included in a route in order to minimize the overall cost. However, by randomly picking any pair of customers gives no specific reason for why they in particular should co-operate; any one of the coalitions $\{1, 2\}$, $\{1, 3\}$ or $\{2, 3\}$ would do. In the second case, there is an obvious reason for customers 2 and 3 to co-operate, since none of them can be combined in a route with customer 1 but both can reduce their common cost by sharing a vehicle. The clustering of the customers ($\{1\}$ and $\{2, 3\}$) is reflected by all the solutions in the core: In fact, any cost allocation in the core is such that $y_1 = 2$ and $y_2 + y_3 = 3.7$. Since the nucleolus is included in the core and the roles of customers 2 and 3 are symmetrical the nucleolus is $y = (2.0, 1.85, 1.85)$.

We generalize these observations below.

Proposition 2. *Consider any optimal route configuration of N, say that it is composed of routes covering the customers of the feasible disjoint coalitions $S_1, S_2, \dots, S_m$. Then,*

$$\sum_{j \in S_r} y_j = c(S_r), \quad \textit{for all } y \in C \textit{ and all } 1 \leqslant r \leqslant m.$$

Proof: As in the proof of Proposition 1, we have that $c(N) = \sum_{r=1}^{m} c(S_r)$ as well as $\sum_{r=1}^{m} \sum_{j \in S_r} y_j = \sum_{j \in N} y_j$, for all $y \in R^n$. Any $y \in C$ satisfies $\sum_{j \in N} y_j = c(N)$ and $bs_{j \in S_r} y_j \leqslant c(S_r)$, for all $1 \leqslant r \leqslant m$, so it follows that $c(N) = \sum_{j \in N} y_j = \sum_{r=1}^{m} \sum_{j \in S_r} y_j \leqslant \sum_{r=1}^{m} c(S_r) = c(N)$. We observe that all inequalities must be equalities and in particular, $\sum_{j \in S_r} y_j = c(S_r)$, for all $y \in C$ and all $1 \leqslant r \leqslant m$. □

Hence, if the core is nonempty, then the cost of a route of any optimal route configuration should be allocated among only the customers covered by that route.

From the example above, we know that the core of the basic VRP game can be empty or non-empty, depending on the data of the problem. In the following statement a sufficient and necessary condition is established for when the core is nonempty. Denote by $(\overline{\text{VRP}})$ the LP-relaxation of (VRP) and by $\bar{z}$ the optimal objective function value of $(\overline{\text{VRP}})$.

Proposition 3. *The core of the basic* VRP *game* $(N;\ c)$ *is nonempty if and only if* $\bar{z} = z$.

Proof: We know that the core of the basic VRP game, $(N;\ c)$, is nonempty if and only if $\omega \geqslant c(N)$, where

$$(\mathrm{P}_0) \qquad \omega = \max \quad \sum_{i \in N} y_i$$

$$\text{s.t.} \quad \sum_{i \in S} y_i \leqslant c(S) \quad S \in \mathcal{S}.$$

Since (P_0) is the dual problem to $(\overline{\text{VRP}})$ it follows that $\omega = \bar{z} \leqslant z = c(N)$. Thus, $\omega \not> c(N)$ and $\omega = c(N)$ exactly when $\bar{z} = z$. □

For computational purposes, if $c(N)$ is computed by solving (VRP) by the use of a LP-based integer method in conjunction with column generation, then information about the duality gap of (VRP) is directly obtained. Such a procedure also provides us with a set of possibly "good" coalitions and corresponding values of $c(S)$ for initiating the cost allocation procedure, as we describe in the following section.

4. Computation of the nucleolus by constraint generation

For the situation when *all relevant* CDIs for a game $(N;\ c)$ are known, Kopelowitz [12] has presented a procedure for computing the nucleolus of a co-operative game by the solution of a sequence of linear programs (see also [6,15]). By applying this procedure on the basic VRP game, the initial linear program to be solved is

$$(\mathrm{P}_1) \quad \max \quad w$$

$$\text{s.t.} \quad w \leqslant c(S) - \sum_{i \in S} y_i, \quad S \in \mathcal{S}, \tag{4.1}$$

$$\sum_{i \in N} y_i = c(N). \tag{4.2}$$

Observe that problem (P_1) concerns the computation of the pre-nucleolus. If the nucleolus is to be computed the constraints (4.1) corresponding to the singleton coalitions should be formulated as $y_i \leqslant c(\{i\})$, $i \in N$.

If it is known that the core is nonempty, then the constraints (4.1) for the coalitions corresponding to the routes of an optimal route configuration can be formulated as equations. Thus, the constraint (4.2) can be omitted.

If (P_1) has a unique optimal solution, (y^*, w^*), i.e. $\theta_1(y^*) > \theta_1(y')$, $y' \in Y \setminus \{y^*\}$, y^* is the nucleolus of the game. Otherwise we have to continue to find the greatest $\theta_2(y)$ of those $y \in Y$ with $\theta_1(y) = w^*$. Let $\Gamma_\tau = \{S \in \mathcal{S} \mid w_\tau = c(S) - \sum_{i \in S} y_i$ for all y that are optimal to $P_\tau\}$. At stage τ the linear programming problem to be solved is

$$
\begin{aligned}
(\mathrm{P}_\tau) \quad \max \quad & w_\tau \\
\text{s.t.} \quad & w_\tau \leqslant c(S) - \sum_{i \in S} y_i, \quad S \in \mathcal{S} \setminus \left(\bigcup_{l=1}^{\tau-1} \Gamma_l \right), \\
& w_l = c(S) - \sum_{i \in S} y_i, \quad S \in \Gamma_l, \; l = 1, 2, \ldots, \tau - 1, \\
& \sum_{i \in N} y_i = c(N).
\end{aligned}
$$

The process continues until P_τ has a unique solution. The nucleolus is the solution to the last problem in the sequence.

The linear problem for computing the nucleolus of the basic VRP game has $|N| + 1$ variables and $|\mathcal{S}| + 1$ constraints. Although the number of CDIs was reduced according to proposition 1, the remaining number of CDIs can still be large. It would be practically impossible to compute the value of $c(S)$ for all those constraints, since one such evaluation involves the solution of a traveling salesman problem. Therefore we suggest a constraint generation procedure [9] for the computation of the nucleolus. Such an approach has earlier been suggested for linear programming games [11].

When a constraint generation procedure is applied on the procedure described above, we divide the solution process into different stages, one corresponding to each problem (P_τ). At stage 1, the initial problem to be solved is a relaxed version of (P_1). The constraint generation procedure should be initiated by at least $|N| + 1$ constraints.

Since at least one optimal route configuration is known, we suggest using the coalitions corresponding to these routes $(r_1^*, \ldots, r_m^*)$ and a subset of at least $|N| - m$ other coalitions for which the values of $c(S)$ are easy to compute, e.g. the singleton coalitions. If $c(N)$ was computed by a column generation approach, the coalitions corresponding to the final columns of that masterproblem can be used.

Denote the current set of inequalities by $\Omega \subseteq \mathcal{S}$. The linear master problem of stage 1 is then

$$
\begin{aligned}
(\mathrm{P}_{M1}) \quad \max \quad & w \\
\text{s.t.} \quad & w \leqslant c(S) - \sum_{i \in S} y_i, \quad S \in \Omega, \\
& \sum_{i \in N} y_i = c(N).
\end{aligned}
$$

Denote the solution to (P_{M1}) by $y^* = (y_1^*, \ldots, y_n^*)$ and w^*. The most violated constraint at (y^*, w^*), of those not in Ω, is then identified by solving the subproblem

$$(\mathrm{P}_S) \qquad \min_{\substack{S \in \mathcal{S} \\ S \notin \Omega}} \; c(S) - \sum_{i \in S} y_i^* - w^* .$$

A solution to (P_S) is the coalition that is the "least satisfied" with the allocation proposal y^*. In the next section we design the subproblem and propose a possible procedure for its solution. Here, we simply assume that an optimal solution, i.e., a coalition S^*, to (P_S) is given. Then, the inequality $w \leqslant c(S^*) - \sum_{i \in S^*} y_i$ is added to (P_{M1}) and the master problem is resolved. Stage 1 is terminated when the objective function value of the subproblem is greater than or equal to zero.

At this stage, all constraints are satisfied and we can proceed to stage 2 in the procedure of computing the nucleolus. That is, active constraints are identified and a master problem (P_{M2}), analogous to problem (P_2), is formulated and solved by constraint generation. The process continues with lexicographic optimization by constraint generation until the first solution y^* of a master problem (P_{M_T}) is unique. The nucleolus of the basic VRP game is the solution y^* to the last solved master problem (P_{M_T}) in the sequence.

Observe that, in (P_S), the feasible set of solutions is restricted to $S \in \mathcal{S} \setminus \Omega$. In general, a previously generated coalition will not be optimal in the subproblem and the restriction $S \in \mathcal{S}$ is sufficient. However, since the rows of the master problem are slightly modified between the different stages, it is obvious that a previously generated coalition can be regenerated. To prevent this, we restrict the set of feasible coalitions in the subproblem to those not included in Ω.

5. The subproblem of the constraint generation scheme

By using the definition of the vector s, the restriction $S \notin \Omega$ can be formulated as [11]

$$\sum_{\{i \mid s_i^{(j)} = 0\}} s_i + \sum_{\{i \mid s_i^{(j)} = 1\}} (1 - s_i) \geqslant 1, \quad j \text{ such that } S_j \in \Omega. \tag{5}$$

That is to say that for each coalition that has been previously generated, such an inequality is formulated and added to the subproblem. Hence, the subproblem can be stated as

$$\begin{aligned}
(\mathrm{P}_S^1) \quad \min \quad & c(S) - \sum_{i \in N} y_i^* s_i \; (-w^*) \\
\text{s.t.} \quad & \sum_{\{i \mid s_i^{(j)} = 0\}} s_i + \sum_{\{i \mid s_i^{(j)} = 1\}} (1 - s_i) \geqslant 1, \; j \text{ such that } S_j \in \Omega, \\
& s_i \in \{0, 1\}, \; i \in N, \\
& S \in \mathcal{S}.
\end{aligned}$$

Recall that $c(S)$, $S \in \mathcal{S}$, is the cost of the minimum cost route that starts and ends at the depot and covers the customers for which $s_i = 1$. This leads us to the explicit formulation of the subproblem presented below. Initially, we discuss the subproblem by omitting the restrictions (5). At the end of this section, further comments on this restriction are made in connection with the description of the enumerative scheme.

We denote by $G = (V, E)$ the graph with vertex set $V = N \cup \{0\}$ and edge set $E = \{(i, j) \mid i, j \in V, i < j\}$. The profit $f_i (\equiv y_i^*)$ and the weight d_i are associated to each vertex, $i \in N$, and the cost c_{ij} to each edge, $(i, j) \in E$. The subproblem of the constraint generation scheme consists of finding a tour in G that maximizes the total profit minus the total cost, such that the total weight of the vertices covered by the tour does not exceed a specified amount Q. Vertex 0 is always included in the tour. By denoting the incidence vector of the tour by λ, this problem can be stated as

$$(\mathrm{P}_S^2)\,\pi = \max \quad \sum_{i \in V} y_i^* s_i - \sum_{i \in V} \sum_{j \in V} c_{ij} \lambda_{ij} \; (+w^*)$$

$$\text{s.t.} \quad \sum_{i \in N} d_i s_i \leqslant Q \tag{6.1}$$

$$\sum_{i \in N} \lambda_{0i} = 2 \tag{6.2}$$

$$\sum_{i=1}^{k-1} \lambda_{ik} + \sum_{j=k+1}^{n} \lambda_{kj} = 2 s_k, \qquad k \in N, \tag{6.3}$$

$$\lambda_{0j} \in \{0, 1, 2\}, \qquad j \in N, \tag{6.4}$$

$$\lambda_{ij} \in \{0, 1\}, \qquad i, j \in N,\ i < j, \tag{6.5}$$

$$s_i \in \{0, 1\}, \qquad i \in N, \tag{6.6}$$

$$\lambda \text{ represents exactly one tour in } G, \tag{6.7}$$

where s is defined as before. Constraint (6.1) limits the tour according to the given weight restriction and (6.2) forces vertex 0 to be covered by the tour. Constraints (6.3) are the vertex balancing equations. A variable λ_{0j} has the value 2 if there is a tour that covers only vertex 0 and j, i.e., the edge $(0, j)$ is used twice. The condition (6.7) can be expressed by using the subtour elimination constraints [13]

$$2 \sum_{k \in T} s_k \leqslant |T| \left(\sum_{\substack{i \in T \\ j \notin T}} \lambda_{ij} + \sum_{\substack{i \notin T \\ j \in T}} \lambda_{ij} \right), \quad T \subset N,\ |T| \geqslant 3.$$

Problem (P_S^2) is closely related to the selective traveling salesman problem [13] and to the prize-collecting traveling salesman problem [1,8]. The selective TSP is the problem of finding a tour in G that maximizes the total profit of visiting the vertices, subject to a specified distance limit not being exceeded. The prize-collecting TSP minimizes the total distance minus total penalty of the tour, while the total prize of the tour must be at least a prescribed amount. The total penalty of a tour is given by the sum of the penalties of the vertices that are not covered by the tour.

In order to solve subproblem (P_S^2) we have adopted the enumerative scheme presented in [13]. There are two differences between the selective TSP and our subproblem, given by (P_S^2). First, we seek to maximize total profit minus total cost whereas the selective TSP minimizes only the total profit. Secondly, the limitation on our tour is given by weights of the vertices and on the selective TSP tour by weights (costs) of the edges. This latter difference is easy to handle, since vertex weights can always be transfered to the edges. The difference in the objective function, however, implies some modifications to the enumerative scheme. Below we give a short presentation of the algorithm used in order to solve (P_S^2).

The algorithm consists of gradually extending a simple path, originating in vertex 0, through a breath-first branch and bound process. The computation of an upper bound on π is based on the solution of a knapsack problem, where each item corresponds to a vertex not covered by the current path. Given the current path, the aim is to compute an upper bound on the maximum possible total net profit (total profit minus total cost) which can be obtained by including further vertices in the path.

Consider a node h of the search tree and let $I(h) = \{t_0 \equiv 0, t_1, \dots, t_h\}$ denote the sequence of the vertices included in the current path, $d(h) = \sum_{i=1}^{h} d_{t_i}$, the corresponding total weight and $f(h) = \sum_{i=1}^{h} f_{t_i} - \sum_{i=1}^{h} c_{t_{i-1}t_i}$ the actual net profit of the path. A branch is created for each vertex included in the set $F(h) = \{i \notin I(h) \mid d(h) + d_i \leqslant Q\}$. For the fathoming test and upper bound computation presented below, we compute the maximum possible net profit of including vertex i in the current path according to

$$\bar{f}_i = f_i - \tfrac{1}{2}\left(c_{\hat{k}_1 i} + c_{\hat{k}_2 i}\right), \quad i \in F(h),$$

where

$$c_{\hat{k}_1 i} = \min_{k_1 \in F(h) \cup \{0, t_h\} \setminus \{i\}} \{c_{k_1 i}\} \quad \text{and} \quad c_{\hat{k}_2 i} = \min_{k_2 \in F(h) \cup \{0, t_h\} \setminus \{i, \hat{k}_1\}} \{c_{k_2 i}\}.$$

The current path $I(h)$ is considered as a super-vertex from which the path originates, i.e., it replaces vertex 0. Thus, the maximum possible net profit of the super-vertex (sv) is

$$\bar{f}_{\mathrm{sv}} = f(h) - \tfrac{1}{2}\left(\min_{k_1 \in F(h)} \{c_{0k_1}\} + \min_{k_2 \in F(h)} \{c_{t_h k_2}\}\right).$$

If $|I(h)| = 1$, then $t_h = t_0 \equiv 0$.

In the fathoming criterium, which is of a knapsack type, we compute the maximum total net profit of the vertices not included in the current path. The criterium says that if $\sum_{i \in F(h)} \max(0, \bar{f}_i) + \bar{f}_{sv} \leqslant \mathrm{LBD}$, then node h can be fathomed. Otherwise the following knapsack problem, or any upper bound on its solution value, provides us with an upper bound on π:

$$(\mathrm{KP1}) \qquad \bar{\pi}_1 = \max \sum_{i \in F(h)} \bar{f}_i s_i + \bar{f}_{\mathrm{sv}}$$
$$\text{s.t.} \quad \sum_{i \in F(h)} d_i s_i \leqslant Q - d(h)$$
$$s_i \in \{0, 1\}, \quad i \in F(h).$$

In (KP1) the items correspond to vertices, $i \in F(h)$. A second type of upper bound computation is obtained by considering arcs as the items to be packed instead. Let $E(h) = \{(i, j) \mid i, j \in F(h) \cup \{0, t_h\}, i < j, (i, j) \neq (0, t_h)\}$. The net profit and weight of arc (i, j) is defined as

$$\bar{c}_{ij} = \begin{cases} \frac{1}{2}(f_i + f_j) - c_{ij}, \\ \frac{1}{2}f_j - c_{ij}, \end{cases} \quad \bar{d}_{ij} = \begin{cases} \frac{1}{2}(d_i + d_j), & i, j \in F(h),\ i < j, \\ \frac{1}{2}d_j, & i \in \{0, t_h\},\ j \in F(h). \end{cases}$$

The corresponding knapsack problem that provides us with an upper bound of π is then

$$\begin{aligned} \text{(KP2)} \qquad \bar{\pi}_2 = \quad & \max \quad \sum_{(i,j) \in E(h)} \bar{c}_{ij} s_{ij} + f(h) \\ & \text{s.t.} \quad \sum_{(i,j) \in E(h)} \bar{d}_{ij} s_{ij} \leqslant Q - d(h) \\ & \qquad s_{ij} \in \{0, 1\}, \quad (i, j) \in E(h). \end{aligned}$$

This bound is valid since (KP2), with $|I(h)| = 1$, is a relaxation of (P_S^2). This is realized by considering the sequence of vertices $I^* = (t_0 \equiv 0, t_1, \ldots, t_h \equiv 0)$ in an optimal solution to (P_S^2). For this solution the objective function of (KP2) takes the value

$$\begin{aligned} & \tfrac{1}{2}f_{t_1} - c_{0t_1} + \sum_{j=1}^{h-2} \left(\tfrac{1}{2}\left(f_{t_j} + f_{t_{j+1}}\right) - c_{t_j t_{j+1}} \right) + \tfrac{1}{2}f_{t_{h-1}} - c_{t_{h-1}t_h} \\ & = \sum_{j=1}^{h-1} f_{t_j} - \sum_{j=0}^{h-1} c_{t_j t_{j+1}} = \pi \end{aligned}$$

and the left-hand side of the knapsack constraint has the value

$$\tfrac{1}{2}d_{t_1} + \sum_{j=1}^{h-2} \tfrac{1}{2}\left(d_{t_j} + d_{t_{j+1}}\right) + \tfrac{1}{2}d_{t_{h-1}} = \sum_{j=1}^{h-1} d_{t_j} \leqslant Q.$$

Hence, an optimal solution to (P_S^2) is feasible in (KP2) and its corresponding value is equal to π.

(KP2) includes a maximum of $\frac{1}{2}n(n-1)$ items and its solution therefore requires more computational time than that of (KP1). However, according to [13], the complexity of the above bound can be reduced by using the Martello and Toth bound [14] to compute an upper bound of the solution value $\bar{\pi}_2$ of the knapsack problem.

Since the subproblem is solved by using an implicit enumeration procedure, the additional restriction in the feasible set of solutions, $S \notin \Omega$, is easy to handle. This is done by simply excluding the corresponding subsets when they occur.

6. Numerical examples

In order to illustrate the different stages of the constraint generation procedure the results for two different test problems are presented. The first example, E1, includes six

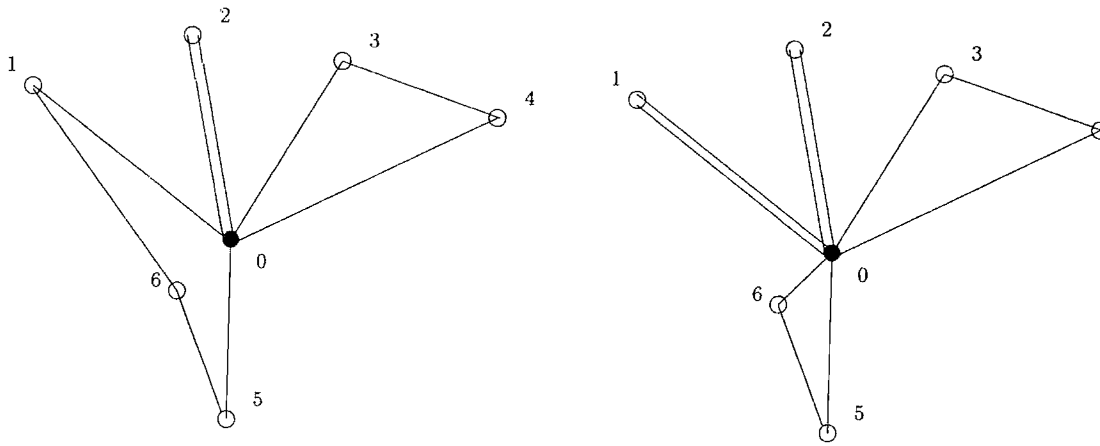

Fig. 2. (left) example E1, cost = 176; (right) example E2, cost = 189.

customers with a demand vector $d = (8, 24, 22, 6, 7, 10)$ and vehicle capacities of 30. The transportation cost matrix is given by

$$\{c_{ij}\} = \begin{pmatrix} - & 24 & 19 & 20 & 27 & 16 & 12 \\ & - & 17 & 31 & 44 & 36 & 23 \\ & & - & 16 & 29 & 35 & 25 \\ & & & - & 15 & 34 & 28 \\ & & & & - & 40 & 37 \\ & & & & & - & 13 \\ & & & & & & - \end{pmatrix}$$

and the customer area, including the optimal route configuration, is shown in Fig. 2 (left). The cost of the route configuration is equal to 176. The second example, E2, is obtained by changing the demand of customer 5 from 7 to 14 units. The corresponding route configuration, with a cost of 189, is shown in Fig. 2 (right). For example E1 problem (VRP) has no duality gap and for example E2 it has.

Before presenting the results of the constraint generation procedure for examples E1 and E2, it could be of interest to describe the core geometrically. This is done for example E1. First, we list the characteristic cost function, $c(S)$, for all *feasible* coalitions S, i.e.

$$\begin{array}{llll} c(\{1\}) = 48, & c(\{1,3\}) = 75, & c(\{3,4\}) = 62 & c(\{1,5,6\}) = 76, \\ c(\{2\}) = 38, & c(\{1,4\}) = 96, & c(\{3,5\}) = 70, & c(\{1,4,5\}) = 123, \\ c(\{3\}) = 40, & c(\{1,5\}) = 76, & c(\{4,5\}) = 83, & c(\{1,4,6\}) = 106, \\ c(\{4\}) = 54, & c(\{1,6\}) = 59 & c(\{4,6\}) = 76, & c(\{4,5,6\}) = 92, \\ c(\{5\}) = 32, & c(\{2,4\}) = 75, & c(\{5,6\}) = 41. & \\ c(\{6\}) = 24, & & & \end{array}$$

Table 1
Results for example E1

Stage	Iteration	w_{stage}	y	Subproblem value	New coalition
1	1	$9\frac{1}{3}$	$(38\frac{2}{3}, 38, 30\frac{2}{3}, 31\frac{1}{3}, 22\frac{2}{3}, 14\frac{2}{3})$	$-5\frac{2}{3}$	{5, 6}
	2	$6\frac{1}{2}$	$(41\frac{1}{2}, 38, 33\frac{1}{2}, 28\frac{1}{2}, 25\frac{1}{2}, 9)$	$-6\frac{1}{2}$	{1, 3}
	3	$6\frac{1}{2}$	$(41\frac{1}{2}, 38, 27, 35, 25\frac{1}{2}, 9)$	$-4\frac{1}{2}$	{2, 4}
	4	5	(40, 38, 30, 32, 27, 9)	4	
Active constraints for coalitions {5, 6}, {1, 3} and {2, 4}					
2	5	8	(40, 38, 30, 32, 24, 12)	−1	{1, 6}
	6	$7\frac{1}{2}$	$(40, 38, 30, 32, 24\frac{1}{2}, 11\frac{1}{2})$	4	
Active constraints for coalitions {5} and {1, 6}					
3	7	8	$(40, 38, 30, 32, 24\frac{1}{2}, 11\frac{1}{2})$ unique		

Based on these, straightforward calculations yield that the core of this VRP game can be described by means of the following equalities and inequalities:

$$y_2 = 38, \qquad y_3 + y_4 = 62, \qquad y_1 + y_5 + y_6 = 76,$$
$$35 \leqslant y_1 \leqslant 48, \qquad y_1 + y_3 \leqslant 75,$$
$$25 \leqslant y_3 \leqslant 40, \qquad y_1 + y_5 \leqslant 76,$$
$$17 \leqslant y_5 \leqslant 32, \qquad y_3 + y_5 \leqslant 70.$$

Further, the core possesses exactly the following 12 extreme points.

$$(35, 38, 25, 37, 17, 24),\quad (35, 38, 25, 37, 32, 9),\quad (35, 38, 38, 24, 32, 9),$$
$$(35, 38, 40, 22, 17, 24),\quad (35, 38, 40, 22, 30, 11),\quad (37, 38, 38, 24, 32, 7),$$
$$(44, 38, 25, 37, 32, 0),\quad (44, 38, 31, 31, 32, 0),\quad (48, 38, 25, 37, 17, 11),$$
$$(48, 38, 25, 37, 28, 0),\quad (48, 38, 27, 35, 17, 11),\quad (48, 38, 27, 35, 28, 0),$$

and from the introduction above, we know that the pre-nucleolus intuitively is the "midpoint" of these extreme points.

Now we continue with the constraint generation procedure. For example E1 problem (P_{M1}) is initialized with constraints corresponding to the singleton coalitions and the coalitions {3, 4} and {1, 5, 6}. Due to proposition 2, the constraints corresponding to {2}, {3, 4} and {1, 5, 6} are imposed as equalities. In Table 1 the results are shown for each iteration.

At stage 1 there are three constraints generated before $w_1 = 5$ is obtained. The value of w_1 is then fixed and the new master problem (P_{M2}) is formulated and solved. Next, after one coalition has been generated the value $w_2 = 7\frac{1}{2}$ is obtained. Finally, the solution to problem (P_{M3}) is unique and therefore equal to the pre-nucleolus. Since problem (VRP) has no duality gap, for example E1, we know that the pre-nucleolus is contained in the core. In total, 12 out of 62 constraints were needed in order to compute the pre-nucleolus.

Table 2
Results for example E2

Stage	Iteration	w_{stage}	y	Subproblem value	New coalition
1	1	$7\frac{1}{2}$	$(48, 38, 32\frac{1}{2}, 29\frac{1}{2}, 24\frac{1}{2}, 16\frac{1}{2})$	−13	{1, 6}
	2	1	(48, 38, 39, 23, 31, 10)	−13	{1, 3}
	3	1	(48, 38, 26, 36, 31, 10)	−4	{1, 5}
	4	−1	(48, 38, 28, 34, 29, 12)	4	
Active constraints for coalitions {1, 6} and {1, 5}					
2	5	3	(48, 38, 24, 38, 29, 12)	−4	{2, 4}
	6	1	(48, 38, 26, 36, 29, 12)	9	
Active constraints for coalitions {1, 3} and {2, 4}					
3	7	3	(48, 38, 26, 36, 29, 12) unique		

For example E2 the first master problem is also inizialized with the singleton coalitions, together with coalitions {3, 6} and {5, 6}. Since problem (VRP) has a duality gap, the core is empty and therefore all constraints are formulated with $\leqslant$ signs. In Table 2 the results for example E2 are shown.

The value of the first element of the excess vector, w_1, is, as expected, negative, i.e., $c(\{1, 5\}) - y_1 - y_5 = 76 - 77 = -1$. After three stages we obtain a unique solution and therefore the pre-nucleolus. As can be seen, for the coalitions corresponding to the optimal routes, {1}, {2}, {3, 4} and {5, 6}, the constraints $\sum_{i \in S} y_i \leqslant c(S)$ are satisfied with equality. This could, however, not be guaranteed since the core is empty. For E2 we also needed 12 out of 62 coalitions in order to compute the pre-nucleolus.

7. Computational experiences and conclusions

We have implemented the implicite enumeration algorithm for the solution of the subproblem. It is implemented on a SUN 4/390 and in Fortran. The upper bound is computed by using the knapsack problem (KP2). For the solution of the master problem we have used the XMP-package of Marsten. We are able to solve problems including up to 25 customers and a vehicle capacity of 15% of the total demand. If the number of customers or the vehicle capacity is increased to above that level, the computational time for the subproblem grows to levels above 5–6 minutes. For problems including less than 25 customers, we can handle vehicle capacities above 15%. For example, to solve the subproblem of a 20-customer problem with vehicle capacities of 21% of the total demand requires up to 1.5 minutes.

We report results for the 25-customers problem, which is, by far, the most difficult problem we have solved. This problem consists of the customers numbered 16–40 in problem 8 in [4]. The vehicle capacity is chosen as 48 units. A list of coordinates and demands of the customers is given in Table 3 (0 = terminal). The transportation costs

Table 3
Data for the 25-customer problem

Customer Number	x-coordinate	y-coordinate	Demand	Customer Number	x-coordinate	y-coordinate	Demand
0	30	40	–	13	43	67	14
1	51	43	15	14	58	48	6
2	27	23	3	15	58	27	19
3	17	33	41	16	37	69	11
4	13	13	9	17	38	46	12
5	57	58	28	18	46	10	23
6	62	42	8	19	61	33	26
7	42	57	8	20	62	63	17
8	16	57	16	21	63	69	6
9	8	52	10	22	32	22	9
10	7	38	28	23	45	35	15
11	27	68	7	24	59	15	14
12	30	48	15	25	5	6	7

are here defined as the Euclidian distances, rounded to one decimal. In Fig. 3 an optimal route configuration, with a cost of 608.9, is shown.

Since this configuration was not known to us, we initiate the procedure by solving the vehicle routing problem. This is done by a column generation approach, with a restricted version of $(\overline{\mathrm{VRP}})$ as the masterproblem and P_S (with $w = 0$) as the subproblem. We denote this by stage 0. The singleton coalitions and a set of coalitions corresponding to routes of a heuristic solution are chosen as initial, with a total of 33 constraints. After another 54 constraints have been generated the procedure terminates with an optimal integer solution. Thus, (VRP) has, for the 25-customer case, no duality gap. Next the computation of the pre-nucleolus is initiated. The constraints corresponding to the coalitions in the optimal route configurations are imposed as equalities.

The computation of the pre-nucleolus was performed in 116 stages and 203 iterations. Together with the 55 iterations from stage 0 we have performed totally 258 iterations. The number of coalitions generated within one stage ranges from zero to three. Out of $2^{24} - 2 = 16\,777\,214$ coalitions we needed 87 in stage 0 and 87 in stage 1–101. The pre-nucleolus, with its elements rounded to two decimals, is

$$\begin{aligned} y = (&19.73, 12.14, 29.60, 24.70, 44.88, 21.78, 11.67, 18.98, 29.68, 29.44, \\ &19.80, 9.84, 23.42, 13.27, 29.42, 19.81, 11.75, 42.43, 39.98, 33.41, \\ &25.06, 14.58, 17.83, 31.69, 34.02). \end{aligned}$$

An interesting observation is that in the last 70 stages, only elements 4 and 25 in the cost allocation vector changed the value.

The computational time for solving the subproblem in stage 1–101 was in the interval (301, 484) CPU-seconds which is rather expensive. In 150 out of 180 iterations (83%) it was in the interval (301, 360) CPU-seconds. In order to approach larger vehicle routing problems there is an obvious need for investigating different solution techniques for the

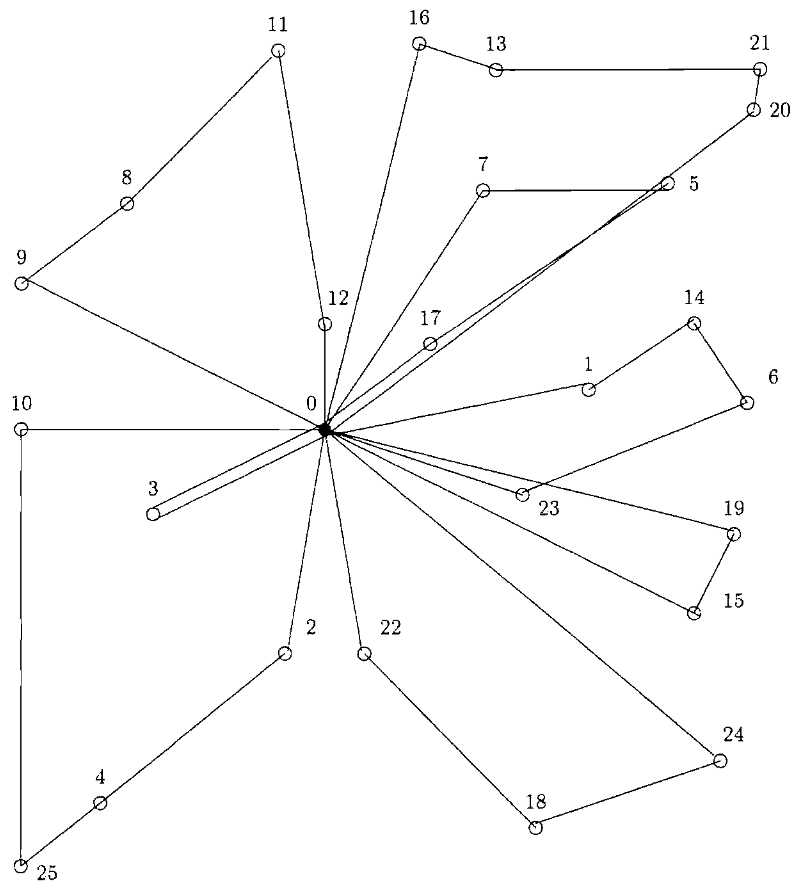

Fig. 3. 25-customer problem.

subproblem further. This is interesting, not only for solving large cost allocation problems, but also for solving (VRP) with a column generation approach.

Our results show that the computation of the pre-nucleolus for medium scale vehicle routing games is possible and that a constraint generation approach is a plausible alternative. The main difficulty found with this approach is to solve the subproblem. We have presented an implicit enumeration procedure for the subproblem, however, it is possible to use a heuristic procedure in order to generate new constraints and use an optimization procedure only in the case when the heuristic fails.

Acknowledgements

We appreciate the conversational feedback and comments given by professor G. Laporte. We would also like to thank one of the anonymous referees for his/her helpful comments.

References

[1] E. Balas, "The prize collecting traveling salesman problem," *Networks* 19 (1989) 621–636.

[2] M.L. Balinski and R.E. Quandt, "On an integer program for a delivery problem," *Operations Research 12*, (1964) 300–304.

[3] L. Bodin, B. Golden, A. Assad and M. Ball, "Routing and scheduling of vehicles and crews: The state of the art," *Computers & Operations Research 10* (1983) 62–212.

[4] N. Christofides and S. Eilon, "An algorithm for the vehicle dispatching problem," *Operational Research Quarterly* 20 (1969) 309–318.

[5] M. Desrochers, J. Desrosiers and M. Solomon, "A new optimization algorithm for the vehicle routing problem with time windows," *Operations Research* 40 (1992) 342–354.

[6] I. Dragan, "A procedure for finding the nucleolus of a cooperative n person game," *Zeitschrift für Operations Research* 25 (1981) 119–131.

[7] M. Dror, "Cost allocation: The traveling salesman, binpacking, and the knapsack," *Applied Mathematics and Computation* 35 (1990) 191–207.

[8] M. Fischetti and P. Toth, "An additive approach for the optimal solution of the prize-collecting travelling salesman problem," in: B.L. Golden and A.A. Assad, (Eds.) *Vehicle Routing: Methods and Studies*, Studies in Management Science and Systems vol. 16, (North-Holland, Amsterdam, 1988) pp. 319–344.

[9] P.C. Gilmore, R.E. Gomory, "A linear programming approach to the cutting stock problem," *Operations Research* 9 (1961) 849–859.

[10] B.L. Golden and A.A. Assad, eds., *Vehicle Routing: Methods and Studies*, Studies in Management Science and Systems, Vol. 16 (North-Holland, Amsterdam, 1988).

[11] A. Hallefjord, R. Helmig and K. Jörnsten, "Calculation of the nucleolus when the characteristic function is given implicitly: A constraint generation approach," Internal Report, Chr. Michelsen Institute, Bergen, Norway (1988).

[12] A. Kopelowitz, "Computation of the kernels of simple games and the nucleolus of n person games," Research Memorandum No. 31, Department of Mathematics, The Hebrew University of Jerusalem (1967).

[13] G. Laporte and S. Martello, "The selective travelling salesman problem, " *Discrete Applied Mathematics* 26 (1990) 193–207.

[14] S. Martello and P. Toth, "An upper bound for the zero-one knapsack problem and a branch and bound algorithm," *European Journal of Operational Research* 1 (1977) 169–175.

[15] M. Maschler, B. Peleg and L.S. Shapley, "Geometric properties of the kernel, nucleolus, and related solution concepts," *Mathematics of Operations Research* 4 (1979) 303–338.

[16] D. Schmeidler, "The nucleolus of a characteristic function game," *SIAM Journal on Applied Mathematics* 17 (1969) 1163–1170.

[17] A, Tamir, "On the core of a traveling salesman cost allocation game," *Operations Research Letters* 8 (1989) 31–34.