Transportation Research Part C 19 (2011) 866–884

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Selective multi-depot vehicle routing problem with pricing

Necati Aras [a,*], Deniz Aksen [b], Mehmet Tuğrul Tekin [a]

[a] *Industrial Eng. Dept., Boğaziçi University, İstanbul, Turkey*
[b] *College of Admin. Sci. and Econ., Koç University, İstanbul, Turkey*

A R T I C L E   I N F O

Article history:
Received 16 December 2009
Received in revised form 18 August 2010
Accepted 18 August 2010

Keywords:
Reverse logistics
Selective multi-depot vehicle routing
Collection
Pricing
Tabu Search

A B S T R A C T

Firms in the durable goods industry occasionally launch trade-in or buyback campaigns to induce replacement purchases by customers. As a result of this, used products (cores) quickly accumulate at the dealers during the campaign periods. We study the reverse logistics problem of such a firm that aims to collect cores from its dealers. Having already established a number of collection centers where inspection of the cores can be performed, the firm's objective is to optimize the routes of a homogeneous fleet of capacitated vehicles each of which will depart from a collection center, visit a number of dealers to pick up cores, and return to the same center. We assume that dealers do not give their cores back free of charge, but they have a reservation price. Therefore, the cores accumulating at a dealer can only be taken back if the acquisition price announced by the firm exceeds the dealer's reservation price. However, the firm is not obliged to visit all dealers; vehicles are dispatched to a dealer only if it is profitable to do so. The problem we focus on becomes an extension of the classical multi-depot vehicle routing problem (MDVRP) in which each visit to a dealer is associated with a gross profit and an acquisition price to be paid to take the cores back. We formulate two mixed-integer linear programming (MILP) models for this problem which we refer to as the selective MDVRP with pricing. Since the problem is $\mathcal{NP}$-hard, we propose a Tabu Search based heuristic method to solve medium and large-sized instances. The performance of the heuristic is quite promising in comparison with solving the MILP models by a state-of-the-art commercial solver.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Environmentally conscious manufacturing, waste reduction and product recovery have emerged in the last two decades as alternative means of environmental sustainability. This motivated researchers and practitioners to focus on closed-loop supply chains which comprise both the (forward) supply chain and the reverse supply chain. The reverse supply chain management consists of the collection, inspection, classification and proper recovery of the products that were used by consumers. One aspect of effectively handling this set of activities is to solve a network design problem to determine the number and locations of to-be-opened facilities and find the amount of product flows between these facilities and demand nodes. The majority of the network design models in the literature utilize linear or nonlinear mixed-integer programming formulations, and solve them either exactly by commercial solvers or approximately by metaheuristic approaches. In this type of models, the location decisions are made only for facilities performing the collection, inspection, and remanufacturing operations. There are also papers which focus on both the forward and reverse distribution channels in a coordinated way. Hence,

* Corresponding author. Tel.: +90 212 3597506; fax: +90 212 2651800.
E-mail address: arasn@boun.edu.tr (N. Aras).

0968-090X/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.trc.2010.08.003

decisions are made for facilities where operations are performed for the forward flow of products to satisfy demand and the reverse flow of used products collected from the consumers. A recent and comprehensive review of both types of papers can be found in Akçalı et al. (2009) and Aras et al. (2010).

One of the key concerns of the companies involved in product recovery is used product acquisition as mentioned in Guide et al. (2003). It is indeed the first activity of product recovery, and triggers the other activities of the recovery system. The quantity and quality of returns can be increased by using buyback campaigns and offering financial incentives to product holders. Accepting all end-of-use products in the waste stream is not a viable strategy for most companies since a high percentage of these products will have poor quality, and hence will not be recoverable. As a consequence, adopting a proactive approach and implementing a used product acquisition strategy by offering the appropriate buyback price is crucial for a company engaged in product recovery. There are several papers that incorporate used product acquisition explicitly. The first study on incentive-based collection modeling which also takes into account the operational issues of network design is due to Boyacı et al. (2008). The authors develop a continuous location model for the collection network by introducing the option of a drop-off strategy. The product return rates depend on the collection strategy in place, the accessibility of the network, as well as on the financial subsidies offered. The return decisions of end-users are captured with a utility based choice model. An analytical framework is proposed to determine the optimal size of the collection area of each facility and the optimal financial incentive offered to end-users so that the profit of the collector is maximized under drop-off and pick-up collection strategies.

An uncapacitated collection center location problem (CCLP) for incentive- and distance-dependent returns is analyzed in Aras and Aksen (2008). The product holders' decision whether or not to participate in the buyback campaign is affected by the distance traveled to the nearest collection center (CC) and the financial incentive that depends on the quality state of the used product. Two mixed-integer nonlinear programming models are proposed for the fixed-charge and p-median versions of the CCLP. The p-median version of the same CCLP is considered in Aksen et al. (2009) under the pickup collection policy in which vehicles depart from CCs, visit a customer zone, and bring the collected used products back. The aim is to determine the locations of the CCs, the level of the financial incentive as well as the number and load mix of the vehicles. In both papers, the authors utilize Tabu Search based solution procedures. Most recently, Aksen et al. (2009) present a bilevel formulation framework describing the subsidy agreement between the government (the leader) and a company engaged in collection and recovery operations (the follower). The authors study two alternative policies. In the supportive policy, the government motivates the company with monetary incentive (subsidy) to achieve a target collection rate. In the legislative policy, on the other hand, the government mandates the collection target rate on the part of the company. In return, the company is guaranteed a threshold economic profitability. In both cases, the objective of the government is to minimize the subsidy to be paid per collected item, whereas the company's aim is to maximize its profit. They show that at the same profitability and collection levels, the government pays a lower subsidy under the legislative policy.

In the present paper, we focus on the vehicle routing aspect in the collection networks. An excellent source for the collection and vehicle routing issues in reverse logistics is Beullens et al. (2004). They describe in detail the features of various versions of the classical vehicle routing problem (VRP). We consider the following situation faced by a firm involved in used product acquisition. The firm has already opened a number of CCs with unlimited storage capacity. It adopts a pick-up policy to collect the used products, referred to as cores, accumulating at the dealers as a result of trade-in campaigns. In these campaigns, consumers purchase new products at a discounted price when they return their used products. This discount is also called trade-in rebate and dealers usually apply a price discrimination policy by offering a trade-in rebate only to the replacement customers to hasten their purchase decisions (Ray et al., 2005). Since the reverse flow of used products induced by trade-in rebates has the potential to generate revenues or cost savings through remanufacturing operations, the firm wants to collect them from the dealers by bearing all the collection-related costs, i.e., the costs of operating vehicles and transporting cores from dealers to CCs. The acquisition price offered by the firm affects the willingness of the dealers to give used products back. Dealers have different reservation prices, and they agree to sell their cores only if the firm's acquisition price exceeds their reservation price. A higher acquisition price reduces the unit revenue from a core, but at the same time increases dealers' willingness to sell the accumulated cores. It is important to note that the acquisition price offered by the firm is the same for all dealerships since a pricing policy discriminating with respect to dealers would not be perceived as a fair attitude. Besides, dealers could easily share among each other the pricing information of the firm. This, in turn, would render a price discrimination policy impractical.

The collection is carried out by a fleet of identical capacitated vehicles where each vehicle is assigned to one of the CCs. There is no restriction regarding the number of vehicles that can be allocated to a CC. Vehicles must depart from and come back to the same CC after visiting a number of dealers. The aim of the firm is to make the collection operation as profitable as possible. The production cost savings, which is due to the remanufacturing of the parts and components obtained from the collected cores, is a source of revenue for the firm. The costs incurred by the firm are the cost of acquisition of the cores and the cost of operating the vehicles. The firm has to decide the number of vehicles allocated to each CC, the assignment of dealers to vehicles, the routes of vehicles, and the uniform acquisition price to be paid for each core. Vehicle routes do not have to include all the dealers. If it is not profitable to collect cores from a dealer, the firm may choose not to send any vehicle to that dealer even when he is ready to sell his cores because the offered acquisition price is greater than or equal to his reservation price.

We refer to this problem as the "selective multi-depot vehicle routing problem with pricing", and denote it as SMDVRPP. The most similar problems to ours studied in the literature are the team orienteering problem (TOP) introduced by Chao et al. (1996), the multiple tour maximum collection problem (Butt and Cavalier, 1994) that is a variant of the TOP introducing heterogeneous vehicles, and the selective VRP with time windows (SVRPTW) proposed by Gueguen (1999) that incorporates additional capacity and time window constraints over TOP. As will be mentioned in the next section, both problems belong to the broader class of the vehicle routing problem with profits (VRPP). However, there exists only a single depot serving the selected customers. To the best of our knowledge, the multi-depot version of the VRPP has not been considered in the literature before. In fact, the classical MDVRP (without profits) generalizes the VRP by using more than one depot or center node. Each depot operates its own vehicle fleet to deliver service to a set of customers. The objective in MDVRP is to design vehicle routes for each depot so that all customers are visited exactly once at the minimum total distance traveled by the fleet. Depots are not allowed to share vehicles, and each tour must terminate at the same depot where it has started. The SMDVRPP that we focus on in this paper involves a set of CCs treated as depots, a set of dealerships acting as customer nodes, and an unlimited number of vehicles where an operating cost is incurred for each vehicle used. There is a profit associated with each customer visit (dealer visit in our case). Hence, the main contribution of our paper is the introduction of SMDVRPP which can be considered as a VRPP with multiple depots. Furthermore, pricing is also integrated into the problem such that an acquisition price at least as much as the visited dealer's reservation price must be paid to take back the dealer's used products. To the best of our knowledge, the pricing issue is not considered before in the literature within the context of a VRPP.

We propose two mixed-integer nonlinear programming (MINLP) formulations for this problem, which are linearized to yield their mixed-integer linear programming (MILP) versions. If the unit cost savings from a core is sufficiently high and if the dealers' reservation prices are all equal to zero, the problem clearly reduces to the classical multi-depot vehicle routing problem (MDVRP) which was shown to be $\mathcal{NP}$-hard (Lenstra and Rinnooy Kan, 1981). This implies that SMDVRPP is also $\mathcal{NP}$-hard. Large instances of SMDVRPP cannot be solved efficiently by solving the MILP formulations with commercial solvers. Thus, we develop a heuristic solution method based on Tabu Search principles.

The rest of the paper is organized as follows. A brief literature review on selective vehicle routing problems is provided in Section 2. The problem is defined and two mathematical programming formulations are given in Section 3. The detailed description of the proposed heuristic solution procedure is the subject of Section 4. Section 5 reports the computational tests of their results. Finally, Section 6 provides some concluding remarks.

2. Literature review

In this paper, we focus on a collection problem, called SMDVRPP, which is faced by a firm involved in product recovery. A number of collection centers (CCs) have been already opened, where inspection and separation on the used products are performed to determine their next destination such as a remanufacturing facility, a recycling facility, or a disposal site. The firm is mainly interested in the transportation of the cores from dealers to the CCs using capacitated vehicles. Apart from the additional requirement that an acquisition price is to be paid for each core, SMDVRPP belongs to the MDVRP class where it is not required to visit all customers (dealers in our problem). Thus, our literature review mainly includes the line of research on the VRP with profits, which is most relevant to our work.

Routing problems where all customers are visited have been the subject of a tremendous number of papers, whereas the number of publications on routing problems with profits is much less. In the latter, the customers to be served and vehicle routes are determined so as to optimize a given objective function. Although visiting all customers boosts the total revenue collected, the marginal profit of visiting certain customers (marginal revenue minus marginal cost) may be less than zero depending on the routing plan at hand. In this case, the total profit could improve if those customers are left out. Feillet et al. (2005) elaborate on the traveling salesman problem with profits (TSPP). TSPP is a generalization of the traveling salesman problem (TSP) where it is not necessary to visit all customers. Associated with each customer there is a profit known a priori. TSPP can be formulated as a discrete bicriteria optimization problem where the two goals are maximizing the profit and minimizing the traveling cost. It is also possible to define one of the goals as the objective function and the other as a constraint. In one version, which is known as the orienteering problem (OP), selective TSP (STSP), or maximum collection problem (MCP) in the literature, the objective is the maximization of the collected profit such that the total traveling cost (distance) does not exceed an upper bound. The other version named as the prize collecting TSP (PCTSP) is concerned with determining the tour with the minimum total traveling cost where the collected profit or prize is greater than a lower bound. There exists a third version of TSPP called the profitable tour problem (PTP) in which the objective is to maximize the difference between the total collected profit and the cost of the total distance traveled. Feillet et al. (2005) provide an excellent survey of the existing TSPP literature. Their survey lists various modeling approaches to TSPP and exact as well as heuristic solution methods.

The extension of TSPP to multiple vehicles is the VRPP. The multi-vehicle version of the OP, where the vehicles are assumed to be uncapacitated and there is a time constraint on each tour, is called the team orienteering problem (TOP). One of the earlier researches on the TOP is due to Chao et al. (1996). The authors propose a five-step metaheuristic similar to deterministic annealing to solve the problem. Butt and Cavalier (1994) address the TOP under the name of multiple tour maximum collection problem (MTMCP) in the context of recruiting football players from high schools. They propose a greedy tour construction heuristic. Butt and Ryan (1999) develop for the MTMCP an exact algorithm based on the branch-and-price method. TOP is solved in Tang and Miller-Hooks (2005) with a Tabu Search heuristic embedded in an adaptive memory procedure. Archetti et al. (2007) propose two variants of a generalized Tabu Search algorithm and a variable neighborhood search algorithm. Boussier et al. (2007) devise another branch-and-price algorithm with new branching strategies and several acceleration techniques. Ke et al. (2008) elaborate an ant colony optimization (ACO) approach for the TOP. One of the two most recent works on TOP is done by Archetti et al. (2009) where the authors study the capacitated version of the TOP and propose exact and heuristic procedures for it. They also investigate an extension of the PTP where a fleet of capacitated vehicles is available. The other work is due to Vansteenwegen et al. (2009). The authors present an algorithm that combines different local search heuristics with a guided local search method and a diversification procedure, which helps explore a wider solution space.

3. Model formulation

Before we present two MILP formulations for the SMDVRPP, we list the assumptions made in the description of the problem. The traveling distance d_{ij} between nodes i and j is known where nodes are the CC and dealer locations. The number of cores a_i owned by dealer i and his reservation price p_i is known by the firm. When the uniform acquisition price W offered to all dealers for taking back a unit core is greater than or equal to p_i, that dealer i sells all available cores in his possession. Each collected core generates a revenue r which represents the remaining value in the used product that can be captured by remanufacturing or recycling. The firm may decide not to collect the cores from a dealer i even if $W \geqslant p_i$, i.e., even if that dealer is willing to give the cores. This is the case when the total revenue that can be obtained from the dealer does not compensate the additional cost of visiting him. The collection operation is performed by a homogeneous fleet of vehicles with capacity q. There is a fixed cost c_1 of operating a vehicle and a variable cost c_2 charged per unit distance traveled. Each dealer can be visited by at most one vehicle, i.e., split collection is not permitted. Moreover, a vehicle must collect all the cores of the dealer it visits. A necessary assumption for the last condition is that $q \geqslant \max_i\{a_i\}$. If $q < \max_i\{a_i\}$, those dealers with $(a_i > q)$ number of cores would not be visited even if it is profitable to do so. Each vehicle must start and finish its tour at the CC to which its is assigned. It cannot be unloaded at another CC and dispatched again. Each CC can operate an unlimited number of vehicles. The objective of the firm is to maximize its profit given by the total revenue gained from collecting the cores minus the total cost of the collection comprised of the buyback cost, vehicle operating cost, and traveling cost.

The first formulation we develop for SMDVRPP called SMDVRPP-1 makes use of the lifted Miller–Tucker–Zemlin (MTZ) subtour elimination constraints (Kara et al., 2004), and tight bounds on the U_i variables used in these constraints. The second formulation SMDVRPP-2 is based on continuous flow variables and incorporate the Gavish–Graves single commodity flow constraints (Gavish and Graves, 1978). In both formulations, $\mathcal{IC}$ denotes the set of CC locations, $\mathcal{ID}$ denotes the set of dealer locations, and $\mathcal{I} = \mathcal{IC} \cup \mathcal{ID}$. Since the parameters are common to both formulations, we summarize their definitions in Table 1.

3.1. SMDVRPP-1

The variables in this model are defined as follows:

$$X_{ij} = \begin{cases} 1 & \text{if location } j \text{ is visited after location } i \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{ik} = \begin{cases} 1 & \text{if dealer } i \text{ is assigned to collection center } k \\ 0 & \text{otherwise} \end{cases}$$

W = acquisition price paid for each core collected

U_i = load of the vehicle right after departing from location i

Note that U_i is only necessary for writing the lifted Miller–Tucker–Zemlin (MTZ) subtour elimination constraints. Based on these variables and parameters, the mathematical programming formulation of the problem is given as follows:

Table 1
Parameters used in the models.

r	Revenue from a core
a_i	Number of cores at dealer i
c_1	Unit vehicle operating cost
c_2	Cost per unit distance traveled
q	Vehicle capacity
p_i	Reservation price of dealer i
d_{ij}	Distance between location i and location j

$$\text{Maximize} \quad \sum_{i\in\mathcal{ID}}\sum_{k\in\mathcal{IC}} Y_{ik}(r-W)a_i - c_1\sum_{i\in\mathcal{IC}}\sum_{j\in\mathcal{ID}} X_{ij} - c_2\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{I},j\neq i} d_{ij}X_{ij} \tag{1}$$

subject to

$$\sum_{j\in\mathcal{I},\ j\neq i} X_{ij} \leqslant 1 \quad i\in\mathcal{ID} \tag{2}$$

$$\sum_{j\in\mathcal{I},\ j\neq i} X_{ji} \leqslant 1 \quad i\in\mathcal{ID} \tag{3}$$

$$\sum_{i\in\mathcal{I},\ i\neq j} X_{ij} = \sum_{i\in I,\ i\neq j} X_{ji} \quad j\in\mathcal{I} \tag{4}$$

$$\sum_{i\in\mathcal{IC}}\sum_{j\in\mathcal{IC}} X_{ij} = 0 \tag{5}$$

$$q\sum_{j\in\mathcal{IC}}\sum_{i\in\mathcal{ID}} X_{ji} \geqslant \sum_{i\in\mathcal{ID}}\sum_{k\in\mathcal{IC}} a_iY_{ik} \tag{6}$$

$$\sum_{k\in\mathcal{IC}} Y_{ik} \leqslant 1 \quad i\in\mathcal{ID} \tag{7}$$

$$\sum_{j\in\mathcal{I},\ j\neq i} X_{ij} \geqslant \sum_{k\in\mathcal{IC}} Y_{ik} \quad i\in\mathcal{ID} \tag{8}$$

$$\sum_{j\in\mathcal{I},\ j\neq i} X_{ji} \geqslant \sum_{k\in\mathcal{IC}} Y_{ik} \quad i\in\mathcal{ID} \tag{9}$$

$$X_{ij} + Y_{ik} - Y_{jk} \leqslant 1 \quad i,j\in\mathcal{ID},\ \ i\neq j,\ \ k\in\mathcal{IC} \tag{10}$$

$$X_{ij} + Y_{jk} - Y_{ik} \leqslant 1 \quad i,j\in\mathcal{ID},\ \ i\neq j,\ \ k\in\mathcal{IC} \tag{11}$$

$$W \geqslant p_i\sum_{k\in\mathcal{IC}} Y_{ik} \quad i\in ID \tag{12}$$

$$U_i - U_j + qX_{ij} + (q-a_i-a_j)X_{ji} \leqslant q - a_j \quad i,j\in\mathcal{ID},\ \ i\neq j \tag{13}$$

$$U_i \geqslant a_i + \sum_{j\in\mathcal{ID},j\neq i} a_jX_{ji} \quad i\in\mathcal{ID} \tag{14}$$

$$U_i \leqslant q - \sum_{j\in\mathcal{ID},j\neq i} a_jX_{ij} \quad i\in\mathcal{ID} \tag{15}$$

$$U_i \leqslant q - (q-a_i)\sum_{j\in\mathcal{IC}} X_{ji} \quad i\in\mathcal{ID} \tag{16}$$

$$U_i \leqslant q - (q - \max_{j\neq i} a_j - a_i)\sum_{j\in\mathcal{IC}} X_{ji} - \sum_{j\in\mathcal{ID},j\neq i} a_jX_{ij} \quad i\in\mathcal{ID} \tag{17}$$

$$X_{ij}, Y_{ik} \in \{0,1\} \tag{18}$$

$$W \geqslant 0, U_i \geqslant 0 \tag{19}$$

Objective function (1) is the maximization of profit that is computed as the total revenue generated from the cores minus the total cost of purchasing cores and operating the vehicles. Constraints (2) and (3) ensure that a dealer can be visited at most once. Constraints (4) guarantee that the number of vehicles (zero or one) entering and leaving a dealer location or a CC must be the same. Constraint (5) makes sure that there is no vehicle travel between two CCs. Constraint (6) specifies that the total capacity of the utilized vehicles must be sufficient to transport the collected cores. The requirement that a dealer be assigned to at most one CC is satisfied by constraint (7). Constraints (8) and (9) make sure that a dealer must be visited by a vehicle if that dealer is assigned to a CC. Constraints (10) and (11) together imply that if two dealers are visited consecutively by a vehicle, then those dealers must be assigned to the same CC. Constraints (12) ensure that the offered acquisition price for each core must be greater than or equal to the maximum reservation price of the visited dealers. Recall that the policy of the firm is not to make a price discrimination among dealers. Constraint (13) is the lifted MTZ subtour elimination constraint, which is used together with tight lower and upper bounds on U_i variables. Constraints (14) put a lower bound on U_i and simply says that the load of the vehicle right after departing from dealer i must be at least equal to the number of cores picked up from dealer i plus the number of cores collected from dealer j visited immediately before dealer i. If the vehicle comes from a CC to dealer i, then second term in this constraint vanishes. Constraints (15)–(17) are all tight upper bounds on U_i variables. Constraints (15) imply that if a vehicle goes from dealer i to dealer j, then the remaining capacity $(q - U_i)$ of the vehicle right after departing from dealer i must be sufficient to collect the cores from dealer j, i.e., $q - U_i \geqslant a_j$. If the vehicle goes to a CC right after dealer i, then this constraint becomes a trivial upper bound on U_i, that is $U_i \leqslant q$. Constraints (16) ensure that if dealer i is visited right after a CC, then the load of the vehicle U_i after departing from dealer i cannot exceed a_i, i.e., the cores picked up from that dealer. If dealer i is not the first dealer on a tour, then the trivial upper bound is obtained. Constraints (17) are similar in nature to (15), but they incorporate the lifting coefficient $(q - \max_{j\neq i}a_j - a_i)$ as proposed in Desrochers and Laporte (1991).

This formulation is an MINLP or a mixed-integer bilinear model in particular because of the bilinear terms $Y_{ik}W$ in the objective function. Bilinear terms are neither concave nor convex, hence commercial solvers cannot guarantee global optimality for the problems involving them. The bilinear term $Y_{ik}W$ can be linearized as follows. First, we observe that $Y_{ik} \in \{0,1\}$ and $W \in [0, \max_{j\in\mathcal{ID}}\{p_j\}]$. Next, we define an auxiliary variable $V_{ik} = Y_{ik}W$ and introduce the following sets of constraints:

$$V_{ik} \geqslant 0 \quad i \in \mathcal{ID},\ k \in \mathcal{IC} \tag{20}$$

$$V_{ik} \geqslant W + \max_{j\in\mathcal{ID}}\{p_j\}(Y_{ik} - 1) \quad i \in \mathcal{ID},\ k \in \mathcal{IC} \tag{21}$$

Replacing the $Y_{ik}W$ terms in the objective function (1) with V_{ik} and adding constraints (20) and (21) gives rise to an equivalent MILP formulation with the objective function

$$\text{Maximize} \quad \sum_{i\in\mathcal{ID}}\sum_{k\in\mathcal{IC}} a_i(rY_{ik} - V_{ik}) - c_1\sum_{i\in\mathcal{IC}}\sum_{j\in\mathcal{ID}} X_{ij} - c_2\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{I},j\neq i} d_{ij}X_{ij} \tag{22}$$

Note that the smallest and largest possible values of V_{ik} is zero and $\max_{j\in\mathcal{ID}}\{p_j\}$, respectively. $Y_{ik} = 0$ means that $V_{ik} = 0$ since $V_{ik} = Y_{ik}W$. This is ensured by constraint (21) because $Y_{ik} = 0$ makes the right-hand side of this constraint $W - \max_{j\in\mathcal{ID}}\{p_j\} \leqslant 0$, which implies that constraint (21) is redundant. Hence, by constraint (20) and the objective function (22), $V_{ik} = 0$ is obtained. When $Y_{ik} = 1$, constraint (21) becomes $V_{ik} \geqslant W$. Since the sense of the optimization is maximization, $V_{ik} = W$ in an optimal solution. This is exactly the equality obtained by the definition of V_{ik}, i.e., $V_{ik} = Y_{ik}W$ with $Y_{ik} = 1$. The applied linearization procedure is proven by Al-Khayyal and Falk (1983) to provide the tightest possible convex underestimator for bilinear terms over a given rectangular region. Consequently, this MILP formulation can be solved by the state-of-the-art commercial MILP solvers such as CPLEX.

3.2. SMDVRPP-2

An alternative MILP formulation can be developed for our problem by defining continuous flow variables F_{ij} that are used in writing Gavish–Graves single commodity flow constraints (Gavish and Graves, 1978). These flow variables represent the amount of cores that are carried by a vehicle from dealer i to dealer j. Keeping the other variables and parameters the same, the new formulation SMDVRPP-2 after the linearization is given as follows:

$$\text{Maximize} \quad \sum_{i\in\mathcal{ID}}\sum_{k\in\mathcal{IC}} a_i(rY_{ik} - V_{ik}) - c_1\sum_{i\in\mathcal{IC}}\sum_{j\in\mathcal{ID}} X_{ij} - c_2\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{I},j\neq i} d_{ij}X_{ij} \tag{23}$$

subject to constraints (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), (21)

$$\sum_{j\in\mathcal{I},\ j\neq i} F_{ij} = \sum_{j\in\mathcal{I},\ j\neq i} F_{ji} + a_i\sum_{k\in\mathcal{IC}} Y_{ik} \quad i \in \mathcal{ID} \tag{24}$$

$$F_{ij} \leqslant (q - a_j)X_{ij} \quad i,j \in \mathcal{I}, j \neq i \tag{25}$$

$$F_{ij} \geqslant a_iX_{ij} \quad i,j \in \mathcal{I}, j \neq i \tag{26}$$

$$X_{ij}, Y_{ik} \in \{0,1\} \tag{27}$$

$$W \geqslant 0, F_{ij} \geqslant 0, V_{ik} \geqslant 0 \tag{28}$$

Constraints (24) are the flow balance constraints written for each dealer $i \in \mathcal{ID}$. Constraints (25) ensure that if a vehicle goes from dealer or CC location i to another location j, then the amount of cores F_{ij} carried from i to j cannot be larger than $(q - a_j)$. In other words, the remaining capacity of the vehicle after leaving location i must allow loading of the cores at location j. Since $a_j = 0$ for $j \in \mathcal{IC}$, these constraints simplify to $F_{ij} \leqslant q$ when location j is a CC. Constraints (26) simply say that the load of a vehicle between locations i and j should be at least equal to the number of cores collected from i. When location i is a CC, then a trivial nonnegativity constraint is obtained ($F_{ij} \geqslant 0$). The set of constraints (24)–(26) eliminates the subtours just as the lifted MTZ constraints do in SMDVRPP-1 formulation.

4. Solution procedure

Since SMDVRPP is at least as difficult as MDVRP, which is known to be $\mathcal{NP}$-hard, it is very difficult to obtain high quality solutions for relatively large-sized instances in a reasonable amount of time by using commercial MILP solvers. Therefore, the development of a metaheuristic algorithm would be beneficial.

Among the metaheuristics proposed for the solution of the MDVRP, the Tabu Search (TS) has been shown to be very effective (Renaud et al., 1996; Cordeau et al., 1997). Motivated by this fact, we also devise a heuristic procedure based on TS principles. TS performs an exploration of the solution space by moving from a solution S_t identified at iteration t to the best solution S_{t+1} in a subset of the neighborhood of S_t. Since S_{t+1} does not necessarily improve upon S_t, a tabu mechanism is put in place to prevent the process from cycling over a sequence of solutions. A simple way to prevent cycles is to forbid the process from going back to previously encountered solutions, but doing so would typically require excessive bookkeeping. Instead, some attributes of past solutions or past moves are declared as tabu such that any solution or move possessing these attributes may not be considered for the next κ iterations. This mechanism is often referred to as short term memory with the number κ called *tabu duration* or *tabu list size*. Other features such as diversification and intensification are often implemented. The purpose of diversification is to ensure that the search process will not be restricted to a limited portion of the solution space. It keeps track of past solutions and penalizes frequently performed moves. This is often called long term memory. Intensification consists of performing a greedy local search around the best known solutions.

We propose a rich neighborhood TS (TS-RN) heuristic for the SMDVRPP. TS-RN is coupled with strategic oscillation, i.e., we admit also those solutions in which one or more tours violate the vehicle capacity constraint. We refer to these as capacity infeasible solutions. In the calculation of such a solution's objective value, we multiply the total overcapacity (the amount of cores carried in excess of vehicle capacity summed over all infeasible tours) with a dynamically changing penalty, and add the resulting infeasibility cost to the total travel distance. At each iteration, the TS-RN algorithm explores a number of neighboring solutions of the current solution including the infeasible ones and selects the best of these in terms of highest objective value as the new current solution. The critical components of this algorithm are the following:

- The initial solution generation method.
- The neighborhood structures and associated tabu conditions.
- The selection of a neighboring solution as the new current solution.
- The procedure which assigns and updates the penalty value for the strategic oscillation.
- The termination conditions.

4.1. Generating an initial solution

The initial solution is formed by putting all dealers on a single tour where the dealers are sorted by their indices. That is, the first dealer comes first and the second dealer comes second on the tour. This tour is attached to the CC with the smallest index. Note that this initial solution is almost certain to be infeasible; thus, its objective value will bear a significant penalty cost for overcapacity.

4.2. Neighborhood structures and tabu conditions

The neighborhood structures or the so-called moves of TS-RN affect the trajectory of the solutions which are designated as the current solution during the iterations. There exist two types of moves in our implementation: routing moves and dealer selection moves. The specific seven routing moves are 1-0 Move, 1-1 Exchange, 2-2 Exchange, 2-Opt, Chain swap, 1-Split, and Inter-tour Exchange. The dealer selection moves are Add-1, Add-2, Drop-1, and Drop-2. Below we explain each move along with a visual representation for the routing moves, where dealers and CCs are designated by circles and squares, respectively.

1-0 Move: Given two dealers, the first one is removed from its current position and is inserted after the second one (Figs. 1 and 2).

1-1 Exchange: Given two dealers, they swap their positions (Figs. 3 and 4).

2-2 Exchange: Given two dealers, the first one and its successor swap their positions with the second one and its successor (Figs. 5 and 6).

2-Opt: Given two dealers on the same tour, the two arcs connecting them with their successors are removed, the dealers are connected, their successors are connected, and finally the chain between the successor of the first dealer and the second dealer is reversed (Fig. 7).

Chain swap: Given two dealers on different tours, the respective chains from each dealer to the last dealer on that tour are swapped (Fig. 8).

1-Split: Given a dealer on an existing tour, the chain from the CC until and including that dealer is retained, whereas the rest of the tour is introduced as a new tour. The new tour can either start at the same CC (Fig. 9) or it can be assigned to

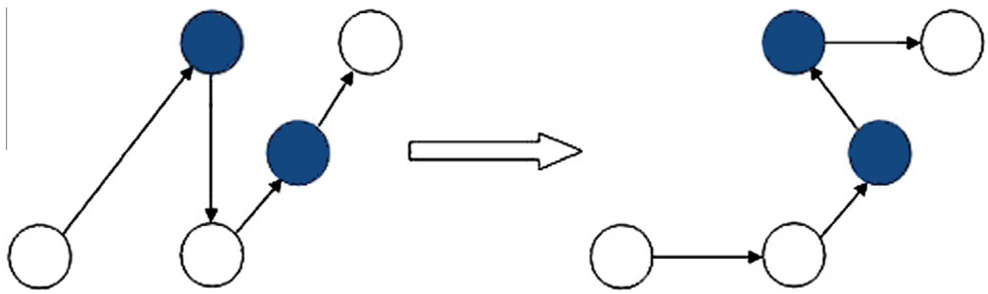

Fig. 1. 1-0 Move on the same tour.

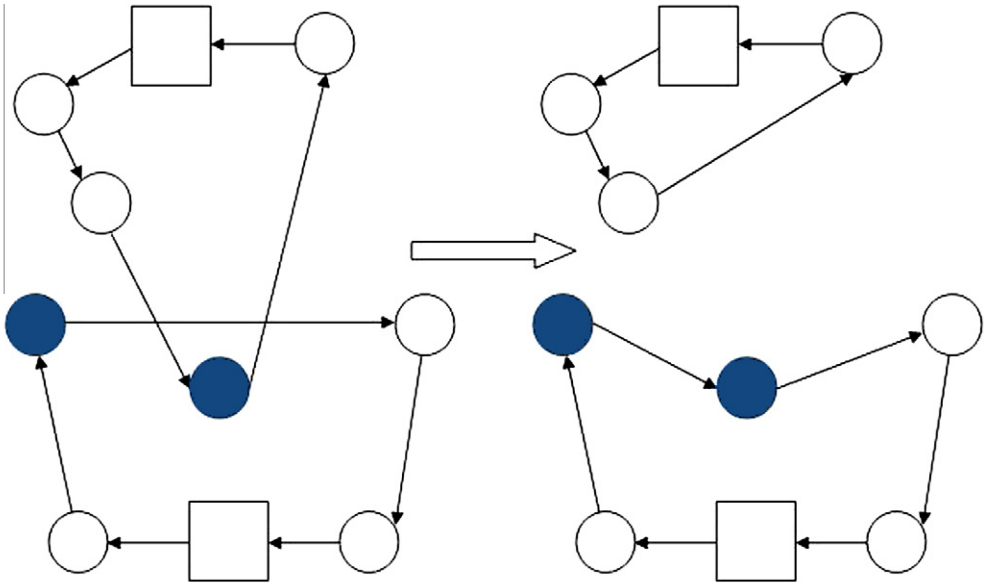

Fig. 2. 1-0 Move on different tours.

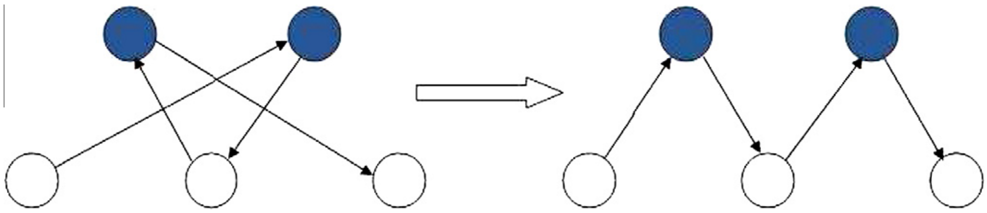

Fig. 3. 1-1 Exchange on the same tour.

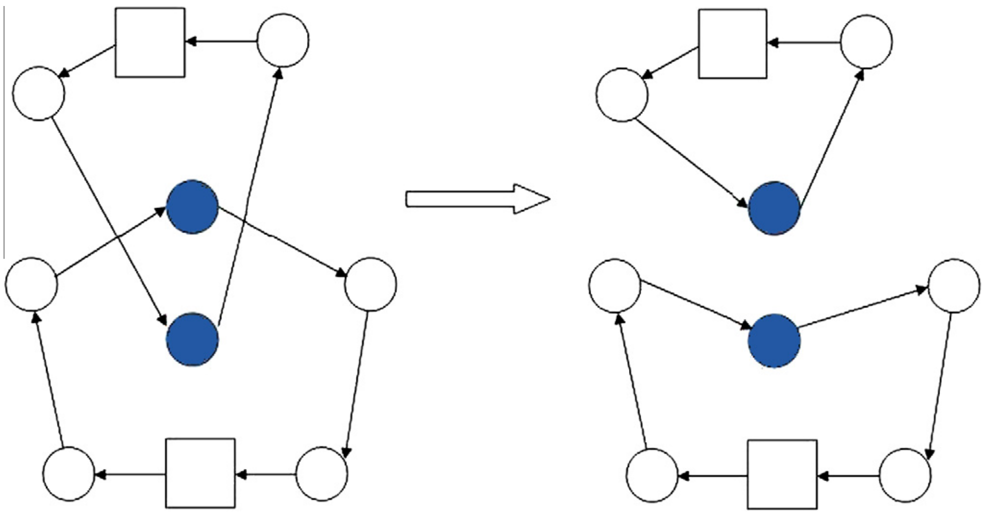

Fig. 4. 1-1 Exchange on different tours.

another CC (Fig. 10). 1-Split neighborhood always increases the total number of tours by one. Note that under the triangular inequality, 1-Split cannot decrease the total travel distance if the new tour is assigned to the same CC as the existing tour, but it may relieve the infeasibility due to overcapacity.

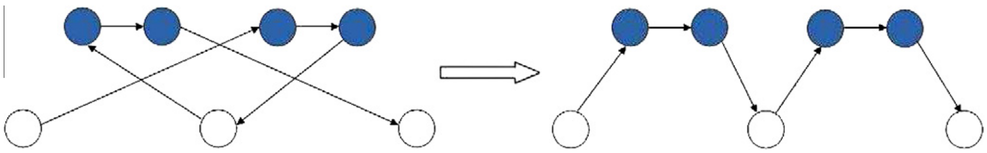

Fig. 5. 2-2 Exchange on the same tour.

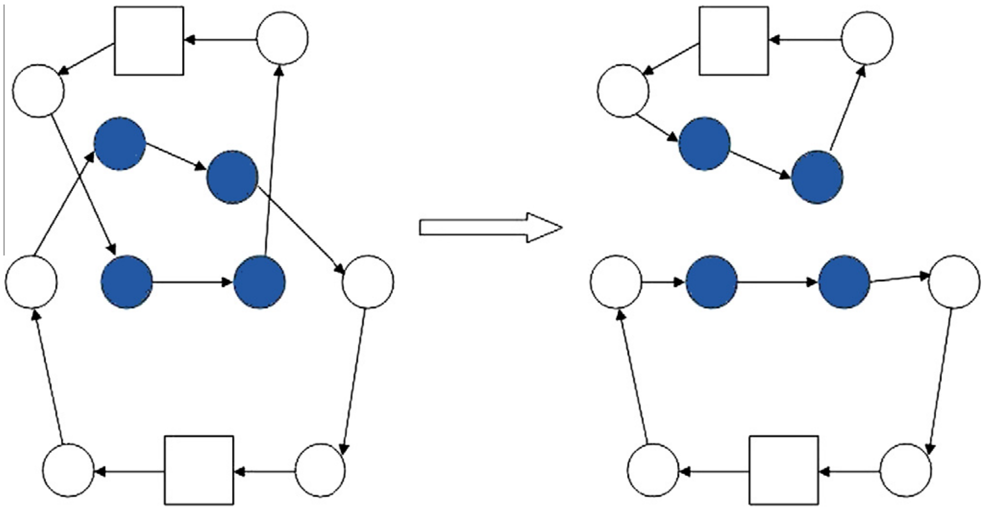

Fig. 6. 2-2 Exchange on different tours.

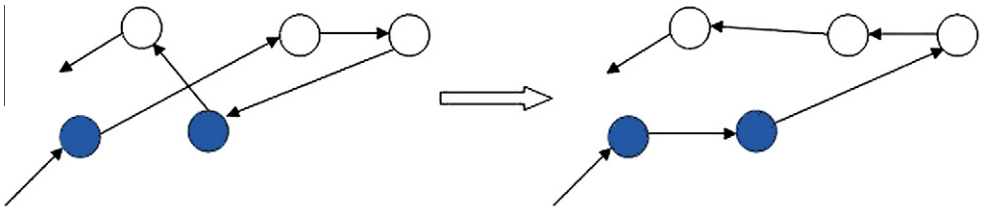

Fig. 7. 2-Opt on the same tour.

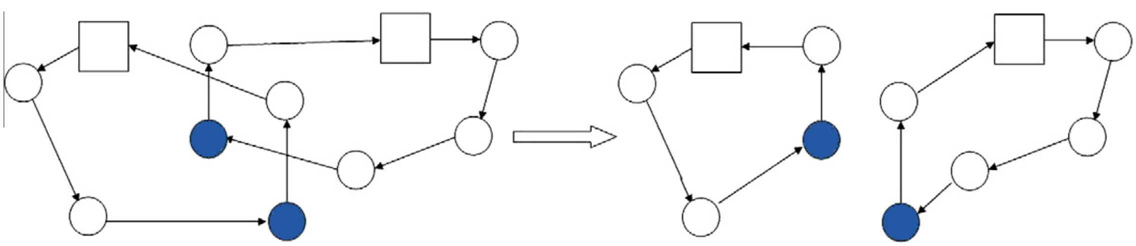

Fig. 8. Chain swap on different tours.

Inter-tour Exchange (ItE): There are three cases of this neighborhood structure.

Case a. A chain of dealers is extracted from a given tour and inserted as a new tour for the same or some other CC as illustrated in Fig. 11. Like 1-Split move, this case of ItE move increases the number of tours by one.

Case b. A chain of dealers in a given tour is moved from its current position into another tour that belongs either to the same or a different CC. This case is shown in Fig. 12.

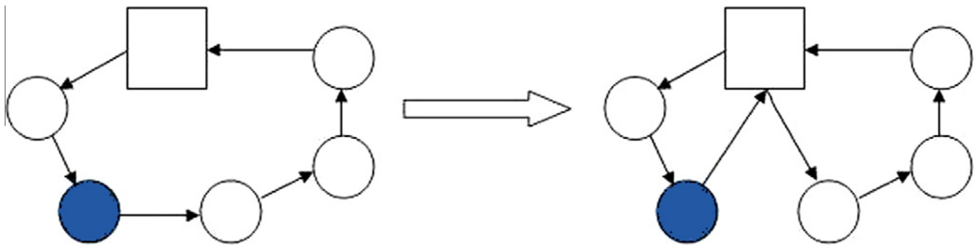

Fig. 9. 1-Split with the same CC.

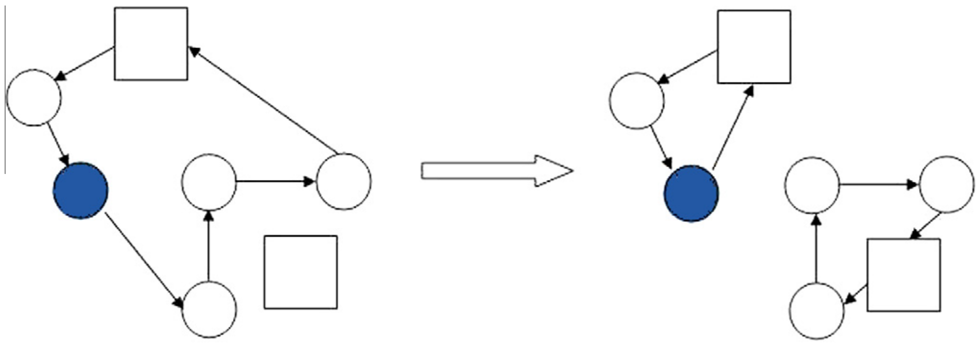

Fig. 10. 1-Split with different CCs.

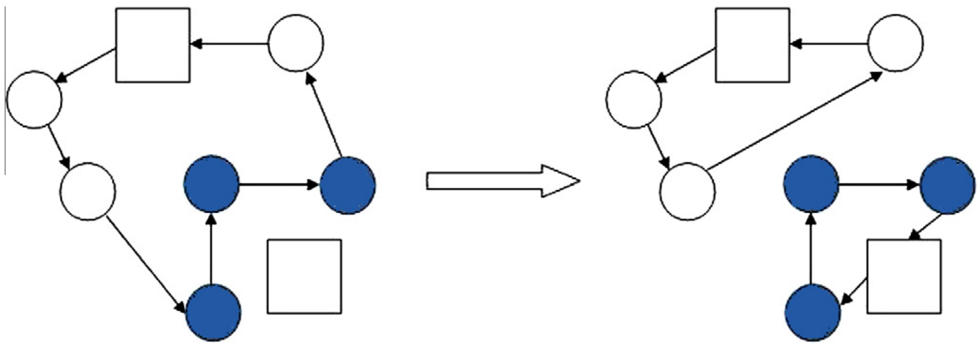

Fig. 11. Case (a) of Inter-tour Exchange.

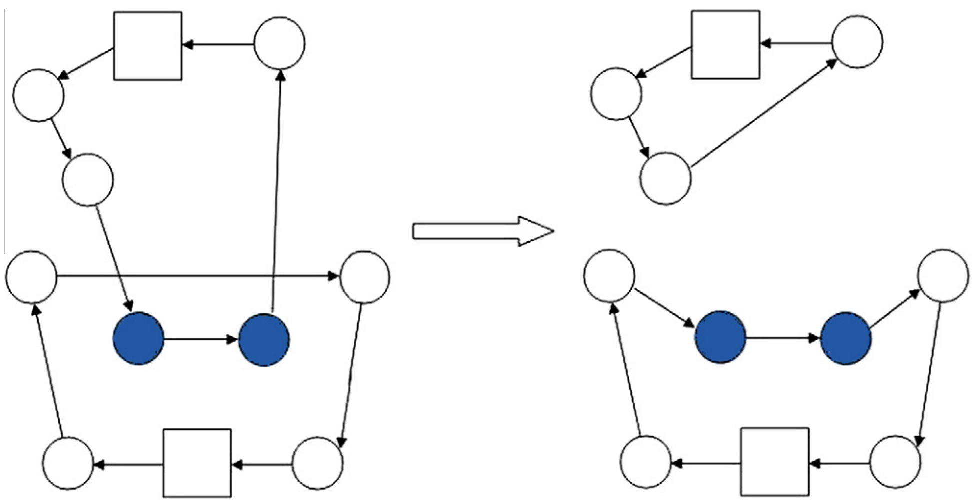

Fig. 12. Case (b) of Inter-tour Exchange.

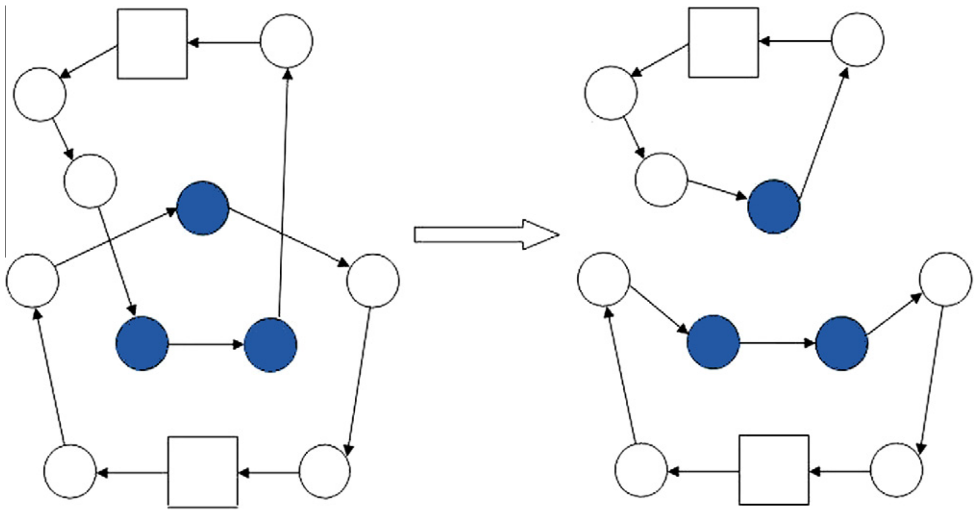

Fig. 13. Case (c) of Inter-tour Exchange.

Case c. Two dealers chains of arbitrary sizes which belong to two different tours are swapped. The third case of the ItE move is depicted in Fig. 13.

Add-1: Given a dealer not included in the tours, all possible insertion positions are examined, and the dealer is inserted in the best possible position. The best possible position is either the one promising the highest increase in the current solution's objective value (total profit), or if no such position exists, then it is the one which decreases the objective value the least. This move is in fact equivalent to the cheapest insertion rule used in the traveling salesman problem.

Add-2: Given two dealers not included in the tours, all possible insertion positions are examined and the given dealers are inserted in succession to each other in the best possible position, which is defined as in the explanation of the Add-1 move. This move corresponds to the cheapest insertion rule applied to a pair of dealers.

Drop-1: Given a dealer included in a tour, it is removed from the tour and its predecessor and successor are connected.

Drop-2: Given a dealer included in a tour, the dealer and his successor are removed from the tour, and the predecessor of the given dealer is connected to the successor of that dealer's successor.

TS-RN algorithm utilizes tabu conditions to enforce exploration of different regions of the solution space. To this end, whenever the solution is updated according to a certain neighborhood structure, a tabu condition is created associated with that structure. It is forbidden to override the tabu condition unless an immediate improvement in the best feasible objective value Z^{best} is realized. The tabu condition is lifted at the end of the tabu duration κ. In our algorithm, we use tabu durations randomly determined in the interval [7,24] independent of the problem size. The tabu conditions associated with all 11 neighborhood structures are listed in Table 2.

Table 2
List of tabu conditions associated with the employed neighborhood structures.

Neighborhood structure	Tabu conditions holding true during the tabu duration
1-0 Move	Dealer whose position is changed cannot be relocated by 1-0 Move
1-1 Exchange	Dealer who switch their positions cannot be re-swapped by 1-1 Exchange
2-2 Exchange	Dealers who were chosen for switching their positions in the 2-2 Exchange cannot be re-swapped by this move. However, it is free to re-swap their successors who were also involved in the 2-2 Exchange
2-Opt	Dealers involved in the 2-Opt move cannot be used together in another 2-Opt move
Chain swap	Dealers involved in the Chain swap move cannot be used together in another Chain swap move
1-Split	Dealers chosen for the splitting position of the 1-Split cannot be selected again in another 1-Split move
Inter-tour Exchange	Dealer(s) constituting the starting node(s) of the relocated (exchanged) chain(s) cannot be selected again in another move of Inter-tour Exchange
Add-1	The added dealer cannot be dropped from the solution
Add-2	The added dealers cannot be dropped from the solution
Drop-1	The dropped dealer cannot be added to the solution
Drop-2	The dropped dealers cannot be added to the solution

Although we propose a large number of different moves, it might be the case that not all of them are really necessary in producing high quality solutions. Implementing only some of them, in particular the easiest ones in Table 2 such as 1-0 Move, 1-1 Exchange and 2-2 Exchange may turn out to be sufficient for good performance. For this reason, we generate a number of move combinations and investigate their performance on the test instances considered in the paper.

4.3. Selecting a neighboring solution as the next current solution

In each TS-RN iteration, when we search the current solution's neighborhood $\mathcal{N}(S_t)$ in the quest for the next current solution S_{t+1}, depending on the move employed we fully explore all neighborhood structures except Inter-tour Exchange. In other words, the candidate subset $Cand\_\mathcal{N}(S_t)$ of $\mathcal{N}(S_t)$ that is to be explored exhaustively and $\mathcal{N}(S_t)$ are equivalent in five of the routing moves. The neighborhoods with respect to Case (b) and Case (c) of the Inter-tour Exchange move (Figs. 12 and 13) are only partially generated and explored due to their computational burden. The neighborhood of Case (a) (Fig. 11), on the other hand, is fully explored. In fact, Inter-tour Exchange is only then performed when the other neighborhood structures cannot provide a better neighboring solution R_t than the current solution S_t, where the basis of comparison is the objective function value inclusive of the infeasibility penalty. To better understand the partial exploration procedure in the last two cases of Inter-tour Exchange, we should first give the following three definitions:

The distance between a node and a tour: The distance between a node N_1 and a tour T, $N_1 \notin T$, is the distance between N_1 and the node $N_2 \in T$ which is closest to N_1 among all nodes in T. We denote this distance by $Dist(N_1,T)$.

The closest tour to a node: The closest tour to a node N_1 is that particular tour T not containing N_1 whose distance to N_1, namely $Dist(N_1,T)$, is the smallest among all other tours not containing that node. We denote the closest tour to N_1 by $T^*(N_1)$ where $N_1 \notin T^*$.

The closest tour set of a tourT: It is the set of the tours closest to each individual node in T. We denote the closest tour set of T by $\mathcal{T}(T)$. For example, let tour T contain only three nodes, N_1, N_2, and N_3. Let the closest tour of N_1 be T' and that of both N_2 and N_3 be T''. Then, $\mathcal{T}(T)$ will be $\{T',T''\}$. Note that T' being in $\mathcal{T}(T)$ does not imply that T is going to be in $\mathcal{T}(T')$.

With the above definitions, it is possible to explain the partial exploration procedure of the last two cases of Inter-tour Exchange. Given a tour T from which a chain is selected, the tours to be examined are only the ones in the closest tour set of T, namely $\mathcal{T}(T)$. That is, it is sufficient to determine and check the routes in the set $\mathcal{T}(T)$ for each individual tour $T \in S_t$. There is no need to consider the other routes not in $\mathcal{T}(T)$, when Case (b) and Case (c) neighborhoods of the Inter-tour Exchange move are explored for a particular tour T in the current solution.

Moreover, suppose that T_2 is in $\mathcal{T}(T_1)$ and T_1 is in $\mathcal{T}(T_2)$ for two tours T_1 and T_2. Then, in Case (c), whenever one of the tours (say T_2) was considered during the exploration of the Inter-tour Exchange neighborhood for the other one (say T_1), it is not necessary to check T_1 during the exploration of the neighborhood for T_2 to save computational time.

A final note should be made regarding the diversification rule of the TS-RN algorithm. While we explore the candidate neighborhood solutions, not only do we record the best neighboring solution R_t^{best}, but also the worst one having the lowest total profit cost minus the infeasibility penalty cost. Let R_t^{worst} denote the worst solution in the neighborhood. Now, if R_t^{best} is ineffective meaning that it bears exactly the same tour profit and infeasibility penalty cost as the current solution S_t, then the next current solution S_{t+1} is set to R_t^{worst}. If R_t^{best} is not ineffective, however, then S_{t+1} is set to R_t^{best} as usual. The jump from S_t to R_t^{worst} instead of R_t^{best} is made in a sense to diversify the search trajectory of our algorithm. By this diversification scheme we intend to break the possible cycling among multiple local optima all having the same objective value.

4.4. Updating the penalty value for strategic oscillation

As mentioned earlier, the TS-RN heuristic explores at each iteration a number of neighboring solutions, which can be feasible or infeasible with respect to the vehicle capacity constraints. The main rationale in choosing the value of the penalty parameter that is used in penalizing the infeasible solutions is to ensure that feasible and infeasible solutions have approximately the same likelihood to be the best neighboring solution and to qualify as the next current solution. This approach adds diversity to the exploration of the solution space. It is clear that a high penalty value may prevent the algorithm from visiting infeasible solutions at all, whereas a low penalty value may fail to identify any feasible solution. Therefore, the value of the penalty parameter is aimed to be changed dynamically to maintain as much as possible an equal number of feasible and infeasible solutions that are visited as current solutions.

For a given number of iterations (*penaltyControlCount*), the number of times an infeasible solution becomes the best neighboring solution and is selected as the next current solution or equivalently the number infeasible solutions visited (*numInfeasible*) is kept track of. If the percentage of *numInfeasible* is greater than 60% of *penaltyControlCount*, then the penalty value (*penaltyValue*) is increased. If that percentage is less than 40% of *penaltyControlCount*, then *penaltyValue* is decreased. In the case that the *numInfeasible* percentage lies in the range 40%–60% *penaltyValue* remains unchanged. The update in *penaltyValue* is performed by adding (subtracting) a certain value called *penaltyStepSize* to (from) *penaltyValue*. In other words, if *penaltyValue* should be increased, then it has to be updated as *penaltyValue*: = *penaltyValue* + *penaltyStepSize*, otherwise *penaltyValue*: = *penaltyValue* – *penaltyStepSize*. The magnitude of *penaltyStepSize* itself is not constant. There are four cases:

i. If *penaltyValue* has to be increased and in the previous control it was increased, then *penaltyStepSize* is doubled and added to *penaltyValue*.
ii. If *penaltyValue* has to be increased and in the previous control it was decreased, then *penaltyStepSize* is halved and added to *penaltyValue*.
iii. If *penaltyValue* has to be decreased and in the previous control it was decreased, then *penaltyStepSize* is doubled and subtracted from *penaltyValue*.
iv. If *penaltyValue* has to be decreased and in the previous control it was increased, then *penaltyStepSize* is halved and subtracted from *penaltyValue*.

Note that each time *penaltyValue* gets updated, the *numInfeasible* parameter is reset to 0. The penalty value update procedure is illustrated in Algorithm 1 in Appendix A where t indicates the current iteration count.

4.5. Termination conditions

Two types of termination conditions are employed in TS-RN. We terminate the algorithm as soon as any of them is met.

i. The maximum number of non-improving iterations limit (*MaxNonimprovIter*). This condition restricts the number of successive iterations of the algorithm during which the best feasible solution S^{best}does not improve.
ii. The time limit (*CPUtimeLimit*). It determines the maximum permissible CPU time for the TS to run.

4.6. The overall TS-RN heuristic

Based on the components explained in detail above, the steps of the resulting TS-RN heuristic is given as Algorithm 2.

Algorithm 2. TS-RN

Begin
    /* *Initialization* */
    Construct S_0 and compute Z_0.
    S^{best}: = S_0, Z^{best}: = Z_0 and $Tabu\_List_0 \leftarrow \emptyset$
    t: = 0
    Repeat
        /* *Neighborhood search to produce the next solution on the trajectory* */
        Generate $Cand\_\mathcal{N}(S_t)$ as a subset of $\mathcal{N}(S_t)$
        Compute the objective value of each neighboring solution R_t in $Cand\_\mathcal{N}(S_t)$
        by penalizing infeasibilities
        SelectionFlag: = FALSE While *SelectionFlag* is FALSE
            Determine the best neighboring solution R_t^{best} in $Cand\_\mathcal{N}(S_t)$
            If R_t^{best} is not in $Tabu\_List_t$
            Or If R_t^{best} is in $Tabu\_List_t$ but satisfies the aspiration criterion
                Then $S_{t+1} := R_t^{best},\ Z_{t+1} := Z(R_t^{best})$, and *SelectionFlag*: = TRUE
                Else $Cand\_\mathcal{N}(S_t) := Cand\_\mathcal{N}(S_t) \setminus R_t^{best}$
        EndWhile
        /* *Update the incumbent if necessary* */
        If $Z(R_t^{best}) < Z^{best}$ AND R_t^{best} is feasible
        Then $S^{best} := R_t^{best}$ and $Z^{best} := Z(R_t^{best})$
        If Z^{best} is updated, then
        apply local post optimization (LPO) to further improve S^{best} and Z^{best}
        /* *Update the tabu list to prevent cycling* */
        Obtain $Tabu\_List_{t+1}$ by adding the attributes of the move from S_t to S_{t+1} to $Tabu\_List_t$
        and deleting those attributes that have stayed in $Tabu\_List_{t+1}$ for the past κ iterations
        t: = t + 1
        /* *Checking for the stopping conditions* */
    Until any of the termination conditions is satisfied.
    Return S^{best} as the best solution found and Z^{best} as its objective value.
End.

Notice that the acquisition price offered per core is determined according to the maximum of the reservation prices of the visited dealers. This is, $W = \max_{i:i \text{ is visited}}\{p_i\}$. In the computation of the objective value of any solution, this value of W should be used. Furthermore, whenever the incumbent solution S^{best} is updated, we apply local post optimization (LPO) to improve S^{best}. LPO is implemented according to a first improvement strategy with the moves 1-1 Exchange, 1-0 Move, 2-Opt, 2-2 Exchange and again 2-Opt respecting this order. Each move is applied repeatedly as long as it produces an improvement in the incumbent solution.

5. Computational results

Here we first describe how we generated random problem instances. Then, we delineate the alternative move combinations we experiment with in our TS-RN heuristic, and report the results of our computations which contain performance comparisons among three solution methods. These include solving the SMDVRPP-1 and SMDVRPP-2 models by CPLEX 11.2 and our heuristic TS-RN which is implemented with the selected move combination. Finally, we also compare TS-RN on some benchmark instances available for the well-studied MDVRP in order to assess its performance.

By assigning five distinct values to the number of CCs ($|\mathcal{IC}| = 1,2,3,4,5$) and ten distinct values to the number of dealers ($|\mathcal{ID}| = 10,20,30,40,50,60,70,80,90,100$), we obtain 40 instances where each instance is labeled as ($|\mathcal{IC}|, |\mathcal{ID}|$). Note that when the number of dealers is relatively small, we also keep the number of CCs small. For example, when $|\mathcal{ID}| = 20, |\mathcal{IC}| = 1,2$ only. The (x,y) coordinates of the CC and dealer locations are sampled independently from a discrete uniform distribution in the interval [0,500]. The travel distances d_{ij} between locations are calculated using the Euclidean distance. The reservation price p_i of each dealer is sampled from a uniform random variable in [5,7.5]. The number of cores a_i at each dealer is generated from a discrete uniform distribution in the interval [5,15]. The fixed vehicle operating cost c_1 is equal to 100 and the variable vehicle cost c_2 per distance traveled is taken as one. The vehicle capacity q is set to 10-fold of the number of cores at the dealer with the maximum number of cores, i.e., $q = 10\max_i\{a_i\} = 150$. The revenue r per unit core is equal to 15. It is worth mentioning that in generating the random instances we pay attention not to obtain instances whose best solution is to visit all the dealers and collect all the cores. In that case, SMDVRPP would reduce to the classical MDVRP.

TS-RN was coded in Java and compiled with Java Server Virtual Machine on a computer having 16 GB of RAM and two Intel QuadCore Xeon 3.2 GHz processors. For all move configurations, TS-RN was implemented with the following parameter values: *penaltyControlCount* = 100, *penaltyValue* = 0, *penaltyStepSize* = 1, *MaxNonimprovIter* = 2000, *CPUtimeLimit* = 15 min.

5.1. Choosing the best move combination

We define a base move combination called C_0, which consists of the following moves: Add-1, Add-2, Drop-1, Drop-2, 1-Split, 2-Opt, and Chain swap. Note that the first four moves are dealer selection moves. Since we are dealing with a selective routing problem in which not all dealers have to be visited, the dealer selection moves are important and therefore all of them are included in the base combination. The other combinations are obtained as given in Table 3, where C_{15} includes all the moves used in this study.

By running TS-RN with different move combinations, C_9 with an average profit of 1116.29 over all the 40 instances considered is found to be the best move combination. Combinations C_{15} and C_{10} appear to be the second and third best combinations with average profits of 1049.35 and 1048.69, respectively. In order to assess the performance of TS-RN with C_9 on a similar problem to the SMDVRPP, we used benchmark instances on the MDVRP. Since our TS-RN heuristic is implemented for the SMDVRPP where we do not consider constraints on the duration of vehicle tours, i.e., the maximum tour duration constraints, we only selected those MDVRP instances that do not have such constraints. Hence, we have implemented TS-RN on 10 MDVRP test instances the data files of which are available on VRP Web (http://neo.lcc.uma.es/radi-aeb/WebVRP) and on the website of the Canada Research Chair in Distribution Management (http://neumann.hec.ca/chairedistributique/data). Table 4 includes the results where for each instance we report the best known objective value, the objective value obtained by TS-RN and its percent deviation from the best known objective value. It can be observed that the accuracy of TS-RN is quite satisfactory. As a consequence, being satisfied with the quality of TS-RN implemented with C_9 we use it in our further analysis.

Table 3
The move combinations considered in TS-RN.

C_1	C_0 + 1-1 Exch.	C_9	C_0 + 2-2 Exch. + ItE
C_2	C_0 + 2-2 Exch.	C_{10}	C_0 + 1-0 Move + ItE
C_3	C_0 + 1-0 Move	C_{11}	C_0 + 1-0 Move + 1-1 Exch. + 2-2 Exch.
C_4	C_0 + ItE	C_{12}	C_0 + 1-1 Exch. + 2-2 Exch. + ItE
C_5	C_0 + 1-1 Exch. + 2-2 Exch.	C_{13}	C_0 + 1-0 Move + 1-1 Exch. + ItE
C_6	C_0 + 1-0 Move + 1-1 Exch.	C_{14}	C_0 + 1-0 Move + 2-2 Exch. + ItE
C_7	C_0 + 1-1 Exch. + ItE	C_{15}	C_0 + 1-0 Move + 1-1 Exch. + 2-2 Exch. + ItE
C_8	C_0 + 1-0 Move + 2-2 Exch.		

Table 4
Performance of TS-RN on some MDVRP instances.

MDVRP instance	Best known	$Z_{\text{TS-RN}}$	Percent deviation
p01	576.87	576.87	0.00
p02	473.53	473.53	0.00
p03	640.65	641.19	0.08
p04	999.21	1017.96	1.88
p05	750.03	750.11	0.01
p06	876.5	890.06	1.55
p07	881.97	897.17	1.72
p12	1318.95	1318.95	0.00
p18	3702.85	3739.65	0.99
p21	5474.84	5516.62	0.76
Average			0.70

5.2. Comparison of the solutions

In Table 5 we present the results to provide a basis for comparing the accuracy and efficiency of the three approaches used to solve SMDVRPP instances. "*LB*" stands for the profit corresponding to the best feasible solution obtained by each approach.

Table 5
The performance of the solution methods on the problem instances.

Instance	SMDVRPP-1			SMDVRPP-2			TS-RN		
$(\lvert\mathcal{IC}\rvert, \lvert\mathcal{ID}\rvert)$	*LB*	*UB*	Time (s)	*LB*	*UB*	Time (s)	*LB*	Time (s)	NoD
(1,10)	0.00	0.00	0.18	0.00	0.00	0.16	0.00	0.41	0
(1,20)	392.21	392.21	35.96	392.21	392.21	44.90	392.21	2.84	11
(1,30)	933.32	1120.55	10,800	933.32	1111.45	10,800	933.32	9.05	24
(1,40)	191.45	671.09	10,800	173.80	521.13	10,800	249.02	22.10	14
(1,50)	401.87	1028.37	10,800	401.87	871.06	10,800	418.20	20.12	13
(1,60)	492.12	1418.30	10,800	669.09	1092.44	10,800	696.10	29.11	28
(1,70)	894.31	1499.35	10,800	912.86	1639.94	10,800	990.88	70.90	41
(1,80)	247.17	1926.83	10,800	579.09	1838.75	10,800	788.38	108.77	45
(1,90)	1023.83	3132.70	10,800	1091.69	3100.38	10,800	1755.59	127.23	68
(1,100)	992.70	3333.43	10,800	770.94	3190.63	10,800	1640.22	243.19	53
(2,20)	514.10	514.10	483.80	514.10	514.10	3362.27	514.10	4.39	13
(2,30)	1144.81	1780.19	10,800	1218.05	1651.11	10,800	1273.97	15.91	28
(2,40)	213.21	983.54	10,800	341.12	1097.01	10,800	319.07	16.62	23
(2,50)	685.15	1553.38	10,800	567.63	1863.77	10,800	685.83	18.19	27
(2,60)	703.19	2329.26	10,800	540.53	2547.51	10,800	712.60	28.39	22
(2,70)	543.98	2931.15	10,800	558.35	3236.72	10,800	1150.25	88.76	42
(2,80)	552.73	3213.58	10,800	374.31	3675.69	10,800	1278.65	148.64	54
(2,90)	370.83	3723.13	10,800	55.10	3559.29	10,800	1101.60	317.73	46
(2,100)	991.78	5246.33	10,800	461.91	5927.52	10,800	2419.65	328.85	55
(3,30)	878.82	1487.87	10,800	873.97	1578.09	10,800	893.68	8.56	21
(3,40)	227.65	1089.90	10,800	246.38	1266.25	10,800	306.37	22.67	14
(3,50)	71.82	1981.86	10,800	233.43	2750.48	10,800	571.61	27.91	26
(3,60)	558.60	2464.03	10,800	681.50	3150.65	10,800	1059.25	41.86	34
(3,70)	401.99	2767.86	10,800	472.12	4109.91	10,800	1156.12	66.98	47
(3,80)	215.43	3353.49	10,800	50.04	5070.39	10,800	1287.61	111.59	54
(3,90)	10.35	4030.02	10,800	0.00	5537.83	10,800	1414.18	199.55	44
(3,100)	208.61	4850.18	10,800	0.00	6866.97	10,800	1860.78	575.69	61
(4,40)	323.98	1255.86	10,800	365.64	1458.02	10,800	396.94	10.60	13
(4,50)	760.85	2285.34	10,800	983.75	2962.25	10,800	1134.32	21.25	34
(4,60)	416.35	2059.81	10,800	409.16	3506.92	10,800	925.07	104.21	45
(4,70)	286.27	3045.65	10,800	86.00	4493.90	10,800	807.19	46.21	27
(4,80)	773.42	3825.19	10,800	1048.77	5806.23	10,800	1760.31	66.94	56
(4,90)	573.70	4740.23	10,800	590.96	7055.75	10,800	1965.16	256.70	58
(4,100)	111.99	4904.26	10,800	999.24	8708.08	10,800	2084.36	496.79	73
(5,50)	354.54	2009.46	10,800	329.49	3322.11	10,800	751.93	39.40	28
(5,60)	1015.01	2784.92	10,800	717.74	4747.13	10,800	1214.53	59.92	40
(5,70)	359.00	3554.14	10,800	553.51	6003.64	10,800	1571.32	94.03	40
(5,80)	827.55	3790.32	10,800	961.37	6487.73	10,800	1758.26	67.90	57
(5,90)	257.83	4528.78	10,800	509.76	8080.62	10,800	1764.24	163.87	50
(5,100)	284.81	5607.99	10,800	0.00	9760.81	10,800	2648.84	149.62	80
Average	505.18	2580.37	10003.00	516.72	3513.86	10075.18	1116.29	105.84	

This value represents a lower bound on the optimal profit of the instance in consideration. "UB", on the other hand, denotes the upper bound found by CPLEX for the SMDVRPP-1 and SMDVRPP-2 models within the allowed time limit of 3 h or equivalently 10,800 s. "NoD" represents the number of dealers visited. CPLEX can obtain the optimal solution only for three instances, which are instances (1,10), (1,20), and (2,20). For the other instances, the optimal profit lies within the values of *LB* and *UB*. The first instance in Table 5 attains the value zero for both *LB* and *UB* regardless of the solution method. This indicates that it is not profitable to collect cores from any one of the ten dealers. Under the SMDVRPP-2 model, some of the *LB* values are zero, which means that CPLEX was not able to produce any feasible solution other than the trivial solution of collecting none of the cores.

When we sort the average *LB* values over all instances, we realize that TS-RN is the best method with $\overline{LB}_{\text{TS-RN}} = 1116.29$. Although the average *LB* of the SMDVRPP-2 model with $\overline{LB}_2 = 516.72$ is higher than that of the SMDVRPP-1 model with $\overline{LB}_1 = 505.18$, it is much worse than the average profit found by TS-RN. From the efficiency point of view as well, TS-RN outperforms the exact solution approaches. It solves the instances in 105.84 s on the average, whereas neither of the mathematical models can surpass the solution quality of TS-RN in 3 h of computing time except for the three instances solved to optimality. Note that although a CPU time limit of 15 min is allotted for TS-RN, the maximum amount of CPU time spent was about 9.5 min. In Table 6, we display the quality of the solutions in terms of the percent deviations of the lower bounds from the best *LB* calculated as $PD_i^{LB} = 100 \times (\max_i LB_i - LB_i)/\max_i LB_i$ and the percent deviations of the upper bounds from the best *UB* calculated as $PD_i^{UB} = 100 \times (UB_i - \min_i UB_i)/\min_i UB_i$ for each instance i.

As can be seen in Table 6, TS-RN provides the best feasible solution with the exception of instance (2,40). With regard to the upper bounds obtained, SMDVRPP-1 outperforms SMDVRPP-2 in 29 out of 40 instances tested. In nine of the instances

Table 6
Comparison of the methods with respect to lower and upper bounds.

Instance	Best	SMDVRPP-1	SMDVRPP-2	TS-RN	Best	SMDVRPP-1	SMDVRPP-2
$(\lvert\mathcal{IC}\rvert, \lvert\mathcal{ID}\rvert)$	*LB*	PD^{LB} (%)	PD^{LB} (%)	PD^{LB}(%)	*UB*	PD^{UB} (%)	PD^{UB} (%)
(1,10)	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(1,20)	392.21	0.00	0.00	0.00	392.21	0.00	0.00
(1,30)	933.32	0.00	0.00	0.00	1111.45	0.82	0.00
(1,40)	249.02	23.12	30.20	0.00	521.13	28.78	0.00
(1,50)	418.20	3.91	3.91	0.00	871.06	18.06	0.00
(1,60)	696.10	29.30	3.88	0.00	1092.44	29.83	0.00
(1,70)	990.88	9.75	7.87	0.00	1499.35	0.00	9.38
(1,80)	788.38	68.65	26.55	0.00	1838.75	4.79	0.00
(1,90)	1755.59	41.68	37.82	0.00	3100.38	1.04	0.00
(1,100)	1640.22	39.48	53.00	0.00	3190.63	4.48	0.00
(2,20)	514.10	0.00	0.00	0.00	514.10	0.00	0.00
(2,30)	1273.97	10.14	4.39	0.00	1651.11	7.82	0.00
(2,40)	341.12	37.50	0.00	6.46	983.54	0.00	11.54
(2,50)	685.83	0.10	17.23	0.00	1553.38	0.00	19.98
(2,60)	712.60	1.32	24.15	0.00	2329.26	0.00	9.37
(2,70)	1150.25	52.71	51.46	0.00	2931.15	0.00	10.42
(2,80)	1278.65	56.77	70.73	0.00	3213.58	0.00	14.38
(2,90)	1101.60	66.34	95.00	0.00	3559.29	4.60	0.00
(2,100)	2419.65	59.01	80.91	0.00	5246.33	0.00	12.98
(3,30)	893.68	1.66	2.21	0.00	1487.87	0.00	6.06
(3,40)	306.37	25.69	19.58	0.00	1089.90	0.00	16.18
(3,50)	571.61	87.44	59.16	0.00	1981.86	0.00	38.78
(3,60)	1059.25	47.26	35.66	0.00	2464.03	0.00	27.87
(3,70)	1156.12	65.23	59.16	0.00	2767.86	0.00	48.49
(3,80)	1287.61	83.27	96.11	0.00	3353.49	0.00	51.20
(3,90)	1414.18	99.27	100.00	0.00	4030.02	0.00	37.41
(3,100)	1860.78	88.79	100.00	0.00	4850.18	0.00	41.58
(4,40)	396.94	18.38	7.88	0.00	1255.86	0.00	16.10
(4,50)	1134.32	32.92	13.27	0.00	2285.34	0.00	29.62
(4,60)	925.07	54.99	55.77	0.00	2059.81	0.00	70.25
(4,70)	807.19	64.53	89.35	0.00	3045.65	0.00	47.55
(4,80)	1760.31	56.06	40.42	0.00	3825.19	0.00	51.79
(4,90)	1965.16	70.81	69.93	0.00	4740.23	0.00	48.85
(4,100)	2084.36	94.63	52.06	0.00	4904.26	0.00	77.56
(5,50)	751.93	52.85	56.18	0.00	2009.46	0.00	65.32
(5,60)	1214.53	16.43	40.90	0.00	2784.92	0.00	70.46
(5,70)	1571.32	77.15	64.77	0.00	3554.14	0.00	68.92
(5,80)	1758.26	52.93	45.32	0.00	3790.32	0.00	71.17
(5,90)	1764.24	85.39	71.11	0.00	4528.78	0.00	78.43
(5,100)	2648.84	89.25	100.00	0.00	5607.99	0.00	74.05
Average		44.12	42.15	0.16		2.51	28.14

Table 7
Investigation of the effect of r.

r	Average revenue	Average profit	Average acquisition price	Average no. dealers visited	Average no. vehicles used
15	6007.13	1116.29	6.25	37.73	2.80
18	9865.20	2659.38	7.02	52.58	3.85
21	12848.50	4530.20	7.24	59.30	4.45
24	15100.80	6428.89	7.29	61.25	4.63
27	17333.33	8414.25	7.34	62.78	4.70
30	19393.50	10393.72	7.38	63.40	4.75

SMDVRPP-2 yields a better *UB*, while in the remaining two instances both models produce the same *UB*. The average percent deviation from the best *UB* is 2.51% for the SMDVRPP-1 and 28.14% for the SMDVRPP-2.

5.3. Sensitivity analysis with respect to r

In this subsection, we analyze the effect of the unit revenue r on the results by increasing its value. To this end, the original value $r = 15$ is assigned values from the set $\{15, 18, 21, 24, 27, 30\}$. Table 7 includes the results where we present the average revenue from collecting the cores, the average profit, the average acquisition price paid to the dealers, the average number of dealers visited, and the average number of vehicles used in the collection as a function of r. As expected, an increase in r causes all of the four quantities to increase because it corresponds to a more profitable collection operation. We can also make the observation that the percentage increase in the average profit outweighs the percentage increase in the average revenue. This stems from the economies of scale in the variable cost within the total cost of operating the vehicles (the other is the fixed cost of the vehicles), and implies that the viability of the collection operation increases as the quality of the cores becomes higher. Therefore, grading the cores with respect to their quality and offering a quality-dependent acquisition price would have the potential of generating extra profit to the firm.

6. Concluding remarks

In this paper, we study the reverse logistics problem of a durable goods manufacturing firm that collects cores returned by the consumers and are accumulated at its dealerships. A dealer does not agree to turn in the cores unless he receives from the firm an acquisition price for each core that is equal to or greater than his reservation price. We assume that the firm has already established several collection centers where inspection and separation of the cores take place, and wants to make decisions about the following three issues: (i) What should be the acquisition price offered to the dealers for each core, (ii) How many vehicles should be used for collection and in what sequence should they visit the dealers, and (iii) What should be the allocation plan of the vehicles to collection centers. We refer to this problem as the selective multi-depot vehicle routing problem with pricing (SMDVRPP). To the authors' knowledge, this problem was not considered before in the literature, as all the existing versions of the routing problems with profits involve either a single vehicle (e.g., orienteering problem) or multiple vehicles with a single depot (e.g., team orienteering problem).

We propose two mixed-integer linear programming formulations for the SMDVRPP which can be solved by a state-of-the-art commercial solver such as CPLEX. However, since the problem is $\mathcal{NP}$-hard, relatively large instances cannot be solved in this way. Therefore, we also develop a heuristic algorithm called TS-RN based on Tabu Search principles. The performance of this heuristic in terms of both accuracy and efficiency is found to be quite promising according to our computational results obtained on 40 randomly generated instances. As a future research direction, we are planning to focus on the location-routing version of this problem where the firm has to simultaneously determine the vehicle routes as well as the number and locations of the collection centers. Another research direction is to consider the SMDVRPP or the location-routing version of it by taking into account the quality-dependent acquisition of the cores.

Acknowledgements

We gratefully acknowledge the support of The Scientific and Technological Research Council of Turkey (TÜBİTAK) under the Grant No: 107M258, and Boğaziçi University Research Fund under the Grant No: 08HA301D.

Appendix A

Algorithm 1. UpdatePenaltyValue

```
Begin
    If t is an integer multiple of penaltyControlCount Then
    begin
      If numInfeasible > 0.6 × penaltyControlCount Then
      begin
        direction: = 1
        If numInfeasible > 0.6 × penaltyControlCount in the previous control
        Then penaltyStepSize: = penaltyStepSize × 2
        Else If numInfeasible < 0.4 × penaltyControlCount in the previous control
        Then penaltyStepSize: = penaltyStepSize/2
      endIf
      Else If numInfeasible < 0.4 × penaltyControlCount Then
      begin
        direction: = −1
        If numInfeasible > 0.6 × penaltyControlCount in the previous control
        Then penaltyStepSize: = penaltyStepSize/2
        Else If numInfeasible < 0.4 × penaltyControlCount in the previous control
        Then penaltyStepSize: = penaltyStepSize × 2
      endIf
    penaltyValue: = penaltyValue + direction × penaltyStepSize
    Reset numInfeasible: = 0;
End.
```

References

Akçalı, E., Çetinkaya, S., Üster, H., 2009. Network design for reverse and closed-loop supply chains: an annotated bibliography of models and solution approaches. Networks 53 (3), 231–248.

Aksen, D., Aras, N., Karaarslan, A.G., 2009. Design and analysis of government subsidized collection systems for incentive-dependent returns. International Journal Production Economics 119 (2), 308–322.

Al-Khayyal, F.A., Falk, J.E., 1983. Jointly constrained biconvex programming. Mathematics of Operations Research 8 (2), 273–286.

Aras, N., Boyacı, T., Verter, V., 2010. Designing the reverse logistics network. In: Ferguson, M.E., Souza, G.C. (Eds.), Closed Loop Supply Chains: New Developments to Improve the Sustainability of Business Practices. CRC Press, pp. 67–97.

Aras, N., Aksen, D., 2008. Locating collection centers for distance and incentive-dependent returns. International Journal Production Economics 111 (2), 316–333.

Aras, N., Aksen, D., Tanuğur, A.G., 2008. Locating collection centers for incentive-dependent returns under a pick-up policy with capacitated vehicles. European Journal of Operational Research 191 (3), 1223–1240.

Archetti, C., Hertz, A., Speranza, M.G., 2007. Metaheuristics for the team orienteering problem. Journal of Heuristics 13 (1), 49–76.

Archetti, C., Feillet, D., Hertz, A., Speranza, M.G., 2009. The capacitated team orienteering and profitable tour problems. Journal of the Operational Research Society 60 (6), 831–842.

Beullens, P., Oudheusden, D.V., Van Wassenhove, L.N., 2004. Collection and vehicle routing issues in reverse logistics. In: Dekker, R., Fleischmann, M., Inderfurth, K., Van Wassenhove, L.N. (Eds.), Reverse Logistics: Quantitative Models for Closed-loop Supply Chains. Springer Verlag, Berlin, pp. 95–134.

Boussier, S., Feillet, D., Gendreau, M., 2007. An exact algorithm for team orienteering problems. 4OR 5 (3), 211–230.

Boyacı, T., Verter, V., Toyasaki, F., Wojanowski, R., 2008. Collection System Design, Strategy Choice and Financial Incentives for Product Recovery. Working Paper. Desautels Faculty of Management, McGill University.

Butt, S.E., Cavalier, T., 1994. A heuristic for the multiple tour maximum collection problem. Computers and Operations Research 21 (1), 101–111.

Butt, S.E., Ryan, D.M., 1999. An optimal solution procedure for the multiple tour maximum collection problem using column generation. Computers and Operations Research 26 (4), 427–441.

Chao, I., Golden, B., Wasil, E., 1996. The team orienteering problem. European Journal of Operational Research 88 (3), 464–474.

Cordeau, J.F., Gendreau, M., Laporte, G., 1997. A tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks 30 (2), 105–119.

Desrochers, M., Laporte, G., 1991. Improvements and extensions to the Miller–Tucker–Zemlin subtour elimination constraints. Operations Research Letters 10, 27–36.

Feillet, D., Dejax, P., Gendreau, M., 2005. Traveling salesman problems with profits. Transportation Science 39 (2), 188–205.

Gavish, B., Graves, S.C., 1978. The Traveling Salesman Problem and Related Problems. Working Paper GR-078-78. Operations Research Center, Massachusetts Institute of Technology.

Gueguen, C., 1999. Méthodes de résolution exacte pour les problèmes de tournées de véhicules. Thése de doctorat, école Centrale Paris.

Guide, V.D.R., Teunter, R., van Wassenhove, L.N., 2003. Matching demand and supply to maximize profits from remanufacturing. Manufacturing and Service Operations Management 5 (4), 303–316.

Kara, I., Laporte, G., Bektaş, T., 2004. A note on the lifted Miller–Tucker–Zemlin subtour elimination constraints for the capacitated vehicle routing problem. European Journal of Operational Research 158 (3), 793–795.

Ke, L., Archetti, C., Feng, F., 2008. Ants can solve the team orienteering problem. Computers & Industrial Engineering 54 (3), 648–665.

Lenstra, J.K., Rinnooy Kan, A.H.G., 1981. Complexity of vehicle routing and scheduling problems. Networks 11 (2), 221–227.

Ray, S., Boyacı, T., Aras, N., 2005. Optimal prices and trade-in rebates for durable, remanufacturable products. Manufacturing and Service Operations Management 7 (3), 208–228.

Renaud, J., Laporte, G., Boctor, F.F., 1996. A tabu search heuristic for the multi-depot vehicle routing problem. Computers & Operations Research 23 (3), 229–235.
Tang, H., Miller-Hooks, E., 2005. A tabu search heuristic for the team orienteering problem. Computers & Operations Research 32 (6), 1379–1407.
Vansteenwegen, P., Souffriau, W., Berghe, G.V., Van Oudheusden, D., 2009. A guided local search metaheuristic for the team orienteering problem. European Journal of Operational Research 196 (1), 118–127.