

A MATHEMATICAL FORMULATION AND HEURISTIC APPROACH FOR THE HETEROGENEOUS FIXED FLEET VEHICLE ROUTING PROBLEM WITH SIMULTANEOUS PICKUP AND DELIVERY

BARIŞ KEÇECİ*

Department of Industrial Engineering, School of Engineering, Başkent University
Bağlıca Kampüsü Fatih Sultan Mahallesi, Eskişehir Yolu 18.km
Etimesgut, Ankara, TR 06790, Türkiye

FULYA ALTIPARMAK AND İMDAT KARA

Department of Industrial Engineering, School of Engineering, Gazi University
Eti Mahallesi, Yükseliş Sokak, No:5
Maltepe, Ankara, TR 06570, Türkiye
and

Department of Industrial Engineering, School of Engineering, Başkent University
Bağlıca Kampüsü Fatih Sultan Mahallesi, Eskişehir Yolu 18.km
Etimesgut, Ankara, TR 06790, Türkiye

(Communicated by Changzhi Wu)

ABSTRACT. This study considers a variant of the vehicle routing problem (VRP) called the heterogeneous VRP with simultaneous pickup and delivery (HVRPSPD). The HVRPSPD may broadly be defined as identifying the minimum cost routes and vehicle types. To solve the HVRPSPD, first, we propose a polynomial-size mixed integer programming formulation. Because the HVRPSPD is an NP-hard problem, it is difficult to determine the optimal solution in a reasonable time for moderate and large-size problem instances. Hence, we develop a hybrid metaheuristic approach based on the simulated annealing and local search algorithms called SA-LS. We conduct a computational study in three stages. First, the performance of the mathematical model and SA-LS are investigated on small and medium-size HVRPSPD instances. Second, we compare SA-LS with the constructive heuristics, nearest neighborhood and Clarke-Wright savings algorithms, adapted for the HVRPSPD. Finally, the performance of SA-LS is evaluated on the instances of the heterogeneous VRP (HVRP), which is a special case of the HVRPSPD. Computational results demonstrate that the mathematical model can solve small-size instances optimally up to 35 nodes; SA-LS provides good quality solutions for medium and large-size problems. Moreover, SA-LS is superior to simple constructive heuristics and can be a preferable solution method to solve HVRP and VRP-SPD instances successfully.

2020 *Mathematics Subject Classification.* Primary: 90B06, 90C11, 90C59; Secondary: 90C27.

Key words and phrases. Vehicle routing problem, simultaneous pickup-delivery, heterogeneous fixed fleet, mathematical model, simulated annealing algorithm.

* Corresponding author: Barış Keçeci.

1. Introduction. At the present time, an increasing population and scarce resources emphasize the importance of logistics and distribution systems for the economy of countries, sectors, and companies, and this can be seen in dramatic reports. Toth and Vigo reported that the use of computerized procedures for distribution process planning produces substantial savings (generally from 5% to 20%) in global transportation costs [63]. A more recent report from the World Bank states that logistics activities in France, which has a higher LPI score in the world ranking, represents 10% of its national gross domestic product, €200 billion turnover, and 1.8 million jobs [3]. Coşar and Demir emphasized that efficient logistics enables countries to participate in global supply chains and exploit their comparative advantages [15]. Their findings have important developmental implications, for example, the more an industry is transportation-sensitive, the greater the reduction in transportation costs.

The vehicle routing problem (VRP) introduced by Dantzig and Ramser is a core application field of operational-level logistics [16]. The VRP, which is an NP-hard problem, can be defined as identifying the minimum cost routes to satisfy customer needs in addition to some side constraints.

The classical VRP consists of identical (homogeneous) vehicles. However, in real-life applications, the vehicles in a fleet may have different features in terms of, for example, purchasing cost, unit transportation cost, and capacity. Moreover, customer (freight) needs may require different vehicle properties. Thus, to reduce the logistics expenditure, both the decisions of fleet selection and route identification have to be examined by firms. In particular, fleet selection is a strategic decision in fleet investment. The classical VRP also assumes that customers have either a delivery or pickup demand. In particular applications, each customer has both delivery and pickup demands simultaneously. In this situation, major economic benefits can be obtained when both activities are performed by the same vehicle.

The heterogeneous VRP (HVRP) may be encountered in various real-life applications. For example, it is common practice to use vehicles with different capacities in residential and commercial waste collection activities. Other examples are shipping services for FedEx Ground and single copy newspaper delivery operations. The VRPSPD is mainly related to the reverse logistics. The management of the reverse flow of products, and raw and work-in-process materials are more challenging topics for companies. Environmental issues are another example where reverse logistics forces firms to use their distribution network more effectively, for example, in waste management, recycling, reprocessing, remanufacturing, and the evaluation of used products. In particular, The VRPSPD can be encountered in many real-life practices such as; the grocery store chains where delivery of fresh food to markets and pickup of outdated items from markets; beverage industry where delivery of soft drink to markets and pickup of empty bottles from markets; foundry industry where delivery of purified reusable sand and pickup of used sand.

In this study, we consider the heterogeneous VRPSPD (HVRPSPD), which includes more realistic features by considering the heterogeneity of the fleet, and simultaneous pickup and delivery. We propose a polynomial-size decision model for the HVRPSPD. Because of the NP-hardness of the HVRPSPD, the proposed mixed integer programming (MIP) formulation can obtain optimal solutions for small-size problems. Thus, we propose a hybrid heuristic algorithm based on simulated annealing (SA) and local search (LS) called SA-LS to solve medium and large-size instances of the problem. It should be noted that SA is a popular neighborhood

search metaheuristic algorithm that can escape from the local optimum. In our implementation, while SA is used to reach different points of the solution space, LS is used to intensively search a particular subspace. To evaluate the effectiveness of the developed hybrid heuristic algorithm, we also adapt well-known and basic constructive heuristics, nearest neighborhood (NN) and Clarke-Wright savings (CWS) algorithms, for the HVRPSPD.

We conduct a three-stage experimental study to investigate the performance of the MIP formulation and SA-LS on HVRSPD instances derived from HVRP instances in the literature. The first stage evaluates the performance of the MIP formulation and SA-LS on small and medium-size HVRSPD instances, and the second stage compares SA-LS with NN and CWS, on large-size HVRSPD instances. Finally, the last stage evaluates the performance of SA-LS on HVRP instances, which is a special case of the HVRPSPD.

The main contributions of this study are as follows: 1) A rarely studied special type of VRP is considered. This problem is directly related to reverse logistics, which is one of the key concepts in green logistics, recycling, and waste management. 2) A new hybrid algorithm based on SA and LS is proposed as a solution approach for the HVRPSPD. 3) Two basic constructive heuristics known in VRP literature are adapted to quickly generate initial solutions for heuristics. 4) A new demand separation method is developed to derive HVRPSPD instances from HVRPs. Additionally, new test instances for the HVRPSPD are introduced. Especially, the relationship of HVRPSPD with several routing problems in logistics is described (see Section 3.3). Furthermore the detailed description about how to adapt the proposed formulation for the variants is provided in detail.

The remainder of the paper is organized as follows: In Section 2, we present a literature review of related works. In Section 3, we define the HVRPSPD, and then present an MIP formulation for the problem and describe the adaptation of the formulation to special cases of the HVRPSPD. In Section 4, we explain the details of the proposed hybrid metaheuristic for the HVRPSPD. We report computational results in Section 5, and discuss the conclusion and future research in Section 6.

2. Literature review. Because there are only a few studies of the HVRPSPD in the literature, first we review the literature for two sub-problems: HVRP and VRPSPD.

Two main types of the HVRP have been studied in the literature for cases in which the fleet has an unlimited or limited number of vehicles. Golden et al. was the first to propose an HVRP in which the fleet has an unlimited number of vehicles of each type [23]. This problem is referred to as the fleet size and mix VRP [23], fleet size and composition VRP [21], and vehicle fleet composition problem [52]. Taillard was the first to study an HVRP that had a limited number of vehicles of each type in the fleet [58]. This type is more realistic and referred to by several terms in the literature, such as the VRP with a heterogeneous fleet of vehicles [58] and heterogeneous fixed fleet VRP [61]. Although these two problems are very similar, their applications are different: the HVRP for an unlimited number of vehicles is more appropriate for strategic decisions when a company wants to buy a vehicle fleet and needs to define its size and composition, whereas the HVRP for a limited number of vehicles better represents the operational decisions of defining the vehicles that should be used, among those available, to serve customers [11]. Related with this paper, a more recent work is done by Juan et al. to analyze different fleet

configurations and proposed several approximation methods [27]. Additionally, it is also possible to classify HVRPs depending on whether fixed and transportation costs are considered. Baldacci et al. presented a classification scheme and provided exact solution algorithms [7] and Baldacci et al. provided latest advances and challenges for the HVRP [6]. We refer interested readers to the papers of [24], [44], [26] and [34] for an extensive review and recent past of this problem and its variants.

The VRPSPD was first introduced by Min [40], who studied a book circulation problem for a public library. The author provided an MIP formulation for the problem and proposed a heuristic algorithm based on the cluster first-route second approach to obtain a solution. In the following years, studies were conducted on the VRPSPD. The mathematical models developed for the VRPSPD can be found in [18], [41] and [42]. Dell’Amico et al. proposed the first exact algorithm for the VRPSPD [17]. Exact dynamic programming and state space relaxation approaches were used for the sub-pricing problem. The algorithm can solve instances up to 40 customers optimally. Ropke and Pisinger developed a large neighborhood search heuristic for the VRPSPD and some VRP variants [50]. Bianchessi and Righini proposed a heuristic approach based on SA for the VRPSPD [9]. Ai and Kachitvichyanukul developed a heuristic algorithm based on particle swarm optimization (PSO) [1]. Gajpal and Abad presented saving-based heuristics for the VRPSPD, where the heuristics were based on merging two existing routes [19]. The feasibility of the new route was checked using a cumulative net-pickup approach. Subramanian et al. presented a parallel heuristic approach based on iterative LS and variable neighborhood search algorithms [55]. Furthermore, Subramanian et al. proposed a branch-cut-and-price algorithm for the VRPSPD [57]. Goksal et al. proposed a hybrid metaheuristic algorithm based on PSO in which the variable neighborhood descent (VND) algorithm was implemented for LS [22]. Tasan and Gen presented a genetic algorithm (GA), which implemented a permutation-based representation [62]. The initial population was constructed randomly, and genetic operators, crossover, and mutation were implemented on the members of the population. Polat et al. considered the VRPSPD with time limitations [46]. In this study, a mathematical formulation based on the model by Montane and Galvao [41] was provided. To solve the problem, the authors presented a perturbation-based neighborhood search algorithm. Avci and Topaloglu developed a hybrid adaptive LS solution approach based on SA and VND for the VRPSPD and one of its variants in which customers were visited in a mixed order [4].

To the best of the authors’ knowledge, there are few studies on the HVRPSPD. Kececi et al. give the mathematical formulations and propose a simple constructive heuristic method [31]. Kececi et al. also developed a matheuristic (mathematical model based heuristic) approach [32]. Avci and Topaloglu proposed a hybrid metaheuristic algorithm for the HVRPSPD in which an LS procedure with a non-monotone threshold adjusting strategy was integrated with the tabu search [5]. In their algorithm, the authors applied an encoding structure, which was proposed in our previous study [30]. They also provided an arc-based MIP formulation, including variables with three indices, which was adapted from Montané and Galvao’s formulation [41]. The performance of the proposed hybrid metaheuristic algorithm was tested by comparing its results with those of a GA on two randomly generated test instance sets, where each set had only 14 problems. Çetin and Gencer considered the heterogeneous fleet VRPSPD with time windows [13]. The proposed MIP

formulation for the problem was based on Dethloff's model [18]. In a recent paper a variant of HVRPSPD which includes the location decision is studied in Wang [64].

Unlike the aforementioned studies, we propose a hybrid algorithm based on the SA and LS and, as observed from the literature review, this type of hybridization of SA and LS is the first application to the HVRPSPD. The rationale behind this hybridization originates from the fact that combining features of different heuristics in a complementary manner can result in more robust and effective optimization tools. Thus, in this paper, we propose an effective solution approach for the HVRPSPD by hybridizing SA with LS, called SA-LS. SA searches for good solutions in the solution space, whereas LS improves the best solution obtained by SA for each iteration. Recently, the hybridization of SA and LS has been used to solve various combinatorial optimization problems in the literature (e.g., scheduling by Liao et al. [35] and Calleja et al. [12], and VRP by Allahyari et al. [2], Zhang et al. [67] and Masmoudi et al. [38]).

3. Mixed integer programming formulation for the HVRPSPD. In this section, after defining the HVRPSPD, we present a MIP formulation for the problem and then explain how the formulation can be revised for special cases.

3.1. Definition of the problem. The definition of HVRPSPD is as follows: Let $G = (N, A)$ be a complete directed graph, where $N = 0, \dots, n$ is the set of nodes and $A = \{(i, j) : i, j \in N, i \neq j\}$ is the set of arcs (0 refers to the depot and the remaining nodes are customers). There are b types of vehicles, with $B = 1, \dots, b$ in the fleet. For each $k \in B$, there are T_k available vehicles with the capacity of Q_k and fixed cost of f_k . Each arc $(i, j) \in A$ is associated with a nonnegative cost, $c_{ij} = \theta_k l_{ij}$ where l_{ij} is the distance from node i to j and $l_{ij} = l_{ji}$. The triangular inequality holds (i.e., $l_{ij} + l_{jk} \geq l_{ik}$) $\forall i, j, k \in N$. θ_k is the variable transportation cost per unit distance of vehicle type $k \in B$. Each customer $i \in N$ has both a pickup (p_i) and delivery (d_i) demand, and $0 \leq d_i, p_i \leq Q_k, \forall k \in B, d_0 = p_0 = 0$. The problem is to determine both the vehicle tours with minimum total cost and vehicle type on each tour under the following restrictions:

- exactly one vehicle type is used on each tour;
- each customer is visited by one vehicle type;
- each vehicle starts and ends its tour at the depot; and
- the total vehicle load is not allowed to exceed the capacity.

3.2. Proposed MIP formulation. To solve the HVRPSPD, we propose an MIP formulation with a polynomial number of constraints and decision variables. The MIP formulation is obtained from the adaptation of Kara and Derya's formulation [28] for the VRP. Additionally, the MIP formulation of the HVRPSPD generalizes the formulations of the VRPSPD proposed by Karaoglan [29] and Montané and Galvao [41] in terms of a heterogeneous fleet.

Because the proposed formulation includes auxiliary decision variables defined on each arc of the distribution network, it is also called an arc-based formulation (ABF). Waters was the first researcher to propose the ABF for the VRP, in which the subtour elimination constraints inspired by mass balance equations were used to restrict both capacity and tour length [66]. Kara and Derya strengthened the model for the same problem using tighter bounding constraints that resulted in a better linear relaxation value [28]. We adapt the subtour elimination and capacity constraints proposed by Kara and Derya [28] in our formulation considering both

pickup and delivery activities. The decision variables of the formulation are given as follows:

0-1 Decision variables: x_{ijk} equal to 1 if arc (i, j) is on the tour of vehicle type k and 0 otherwise, $\forall i, j \in N, \forall k \in B$.

Continuous decision variables: y_k : number of type k vehicles in the fleet ($k \in B$), m : number of vehicle tours.

Auxiliary decision variables: z_{ij} : remaining delivery load if the vehicle passes from i to j immediately after visiting node i , 0 otherwise; t_{ij} : load picked up if the vehicle passes from i to j immediately after visiting node i , 0 otherwise.

The proposed mathematical model is given as follows:

Formulation (ABF): The objective function is the minimization of the total transportation and vehicle utilization costs and is denoted by

$$\min z = \sum_{i \in N} \sum_{\substack{j \in N \\ i \neq j}} \sum_{k \in B} c_{ij} x_{ijk} + \sum_{k \in B} f_k y_k \quad (1)$$

subject to the degree constraints for the depot given by

$$\sum_{j \in N \setminus \{0\}} \sum_{k \in B} x_{0jk} \leq m \quad (2)$$

$$\sum_{i \in N \setminus \{0\}} \sum_{k \in B} x_{i0k} \leq m \quad (3)$$

The following constraints force the solution to have exactly one vehicle type moving in and out at any node; hence, it provides vehicle type continuity:

$$\sum_{\substack{i \in N \\ i \neq j}} \sum_{k \in B} x_{ijk} = 1 \quad \forall j \in N \setminus \{0\} \quad (4)$$

$$\sum_{\substack{j \in N \\ i \neq j}} x_{ijk} = \sum_{\substack{j \in N \\ i \neq j}} x_{jik} \quad \forall i \in N \setminus \{0\}, \forall k \in B \quad (5)$$

On any arc, the constraint

$$z_{ij} + t_{ij} \leq \sum_{k \in B} Q_k x_{ijk} \quad \forall i, j \in N, i \neq j \quad (6)$$

does not allow the violation of the vehicle capacity. The following constraints relate to the delivery and pickup load, respectively, and both prevent the subtours:

$$\sum_{\substack{j \in N \\ i \neq j}} z_{ji} - \sum_{\substack{j \in N \\ i \neq j}} z_{ij} = d_i \quad \forall i \in N \quad (7)$$

$$\sum_{\substack{j \in N \\ i \neq j}} t_{ij} - \sum_{\substack{j \in N \\ i \neq j}} t_{ji} = p_i \quad \forall i \in N \setminus \{0\} \quad (8)$$

The following constraints bound the delivery and pickup loads, respectively:

$$\sum_{k \in B} d_j x_{ijk} \leq z_{ij} \leq \sum_{k \in B} (Q_k - d_i) x_{ijk} \quad \forall i, j \in N, i \neq j \quad (9)$$

$$\sum_{k \in B} p_i x_{ijk} \leq t_{ij} \leq \sum_{k \in B} (Q_k - p_j) x_{ijk} \quad \forall i, j \in N, i \neq j \quad (10)$$

The following constraints initially provide value to the pickup and delivery loads, respectively, because a vehicle starts and ends its tour with an empty load:

$$t_{0j} = 0 \quad \forall j \in N \setminus \{0\} \quad (11)$$

$$z_{i0} = 0 \quad \forall i \in N \setminus \{0\} \quad (12)$$

The following constraint provides at most m vehicles used in the fleet:

$$\sum_{k \in B} y_k \leq m \quad (13)$$

The following constraint is the vehicle type availability restriction:

$$y_k \leq T_k \quad \forall k \in B \quad (14)$$

The following constraint ensures a feasible solution, which is that the leaving arcs from the depot associated with vehicle type k must be exactly the same as the number of vehicles of type k :

$$\sum_{j \in N \setminus \{0\}} x_{0jk} = y_k \quad \forall k \in B \quad (15)$$

The integrality and non-negativity constraints are

$$y_k \geq 0 \text{ and integer} \quad \forall k \in B \quad (16)$$

$$m \geq 0 \quad (17)$$

$$z_{ij}, t_{ij} \geq 0 \quad \forall i, j \in N \quad (18)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i, j \in N, \forall k \in B \quad (19)$$

In the ABF, the number of binary decision variables is $O(n^2b)$, number of positive variables is $O(2n^2)$, number of positive and integer variables is $O(b)$, and number of restrictions is $O(3n^2)$.

3.3. Special cases of HVRPSPD. It should be noted that the proposed MIP formulation of the HVRPSPD is easily adaptable to special variants, which arise in logistics problems. Figure 1 shows the connections between the HVRPSPD and its variants. An HVRP is obtained by setting either p_i or d_i for all $i \in N$ to zero in

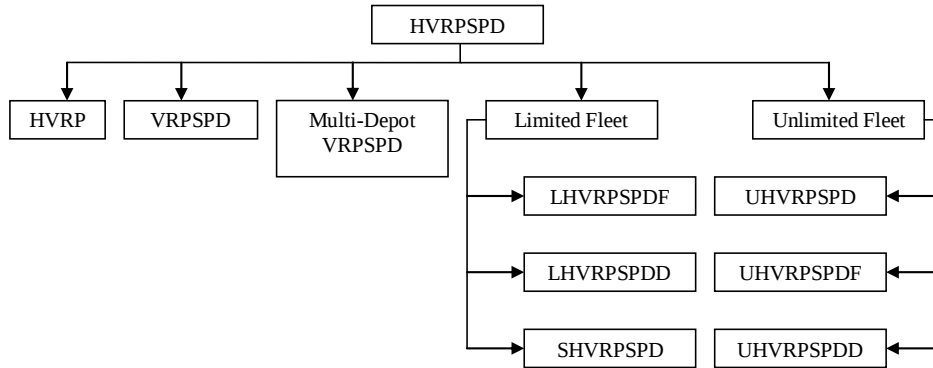


FIGURE 1. Relationship between the HVRPSPD and its variants

the HVRPSPD. Any HVRPSPD instance can easily be reduced to a VRPSPD by setting $|B| = 1$ and $\theta_k = 1, f_k = 0, T_k = +\infty, Q_k = Q$ for all $k \in B$. The multi-depot VRPSPD (MVRPSPD) is an extension of the VRPSPD in which a customer can be served by an unlimited fleet of identical vehicles of capacity Q located at D depots. Let $[\hat{d}_{ij}]$ be a symmetric cost matrix with dimensions $(|N| - 1 + D) \times (|N| - 1 + D)$, where $\hat{d}_{(|N|-1+r),i}$ is the travel cost of passing from depot $r = 1, \dots, D$ to customer $i \in N \setminus \{0\}$. Any MVRPSPD instance can be converted into an equivalent HVRPSPD instance (i.e., unlimited fleet HVRPSPD with dependent routing costs (UHVRPSPDD)) that generates $b = D$ vehicle types ($B = \{1, \dots, D\}$) and sets for each vehicle type $k \in B$: $Q_k = Q, m_k = |N| - 1, f_k = 0$, and $c_{ijk} =$

$$\begin{cases} \hat{d}_{(|N|-1+r),j} & , \text{ if } i = 0 \\ \hat{d}_{ij} & , \text{ otherwise} \end{cases}, \forall (i, j) \in N. \text{ According to the fleet size, variants of}$$

the HVRPSPD can be grouped into two classes. The proposed formulation of the HVRPSPD can easily be adapted for the limited fleet HVRPSPD with fixed costs (LHVRPSPDF) by setting $\theta_{k_1} = \theta_{k_2}, \forall k_1, k_2 \in B, k_1 \neq k_2$ or $\theta_k = 1, \forall k \in B$. For this case, the routing costs do not depend on the vehicle type and there will only be vehicle dependent fixed costs. For the opposite case, where fixed costs do not depend on the vehicle type (or do not exist) but routing costs do, the problem becomes the limited fleet HVRPSPD with dependent routing costs (LHVRPSPDD). The proposed formulation of the HVRPSPD can easily be adapted for the LHVRPSPDD by setting $f_{k_1} = f_{k_2}, \forall k_1, k_2 \in B, k_1 \neq k_2$ or $f_k = 0, \forall k \in B$ if there is no fixed cost. In the SHVRPSPD, there is a limited number of heterogeneous vehicles available for the service with no fixed costs, and routing costs do not depend on the vehicle type. However, each customer may include restrictions on the vehicle types that may visit it. To adapt the HVRPSPD formulations to this special case, let $B'_j, \forall j \in N \setminus \{0\}$ be the set of vehicle types that can visit the customer. Set $Q_k = M, \forall k \in B \setminus B'_j, j \in N \setminus \{0\}$ for the vehicle types that are not allowed to visit the customer; set $\theta_{k_1} = \theta_{k_2}, \forall k_1, k_2 \in B'_j, k_1 \neq k_2$ or $\theta_k = 1, \forall k \in B'_j, j \in N \setminus \{0\}$ for the vehicle types that are allowed to visit the customer, where M is a very large number; ignore the second term of the objective function given in (1). If there is an unlimited number of vehicles in the fleet, the problem becomes the HVRPSPD with unlimited fleet (UHVRPSPD). A HVRPSPD instance can be converted to UHVRPSPD by setting $T_k = +\infty$ or by omitting (14) in the formulation. If there is an unlimited fleet with routing costs independent of the vehicle types and with vehicle dependent fixed costs, the HVRPSPD formulation can be adapted to the unlimited fleet HVRPSPD with fixed costs (UHVRPSPDF) by omitting (14) and setting $\theta_k = 1, \forall k \in B$. For the opposite case, where fixed costs do not depend on the vehicle type (or do not exist) but routing costs do with an unlimited fleet of vehicles, the HVRPSPD formulation can be adapted to the UHVRPSPDD by omitting (14) and setting $f_{k_1} = f_{k_2}, \forall k_1, k_2 \in B, k_1 \neq k_2$ or $f_k = 0, \forall k \in B$ if there are no fixed costs. With these reductions, all of these aforementioned special cases can be solved using the proposed MIP formulation for the HVRPSPD.

4. Hybrid simulated annealing algorithm for the HVRPSPD. Because of the NP-hardness of HVRPSPD, the MIP formulation cannot directly be applied to solve optimally large and even medium-size problems. Thus, heuristics are required to rapidly obtain reasonable solutions. In this study, to solve medium and large-size

HVRPSPD instances, we propose a hybrid heuristic method, which combines SA and LS.

After SA was proposed by Metropolis et al. [39], it was improved by Kirkpatrick et al. [33] to solve optimization problems. The name of the algorithm was derived from the analogy between solving optimization problems and simulating the annealing process of solids.

Studies in recent decades have considered hybrid algorithms, in which metaheuristics are used together with LS, or other metaheuristic or exact solution algorithms to take advantage of more than one approach [60]. LS is also a well-known simple heuristic that searches neighbors of the current solution until no improved solution is obtained.

In our hybrid algorithm, while SA is used to reach different points of the solution space, LS is used to intensively search a particular subspace. The proposed hybrid algorithm is called SA-LS. This successive use of SA and LS continues until stopping criteria are met.

In this section, the basic structure of SA-LS is described by considering the representation of the solution, generation of the initial solution, neighborhood structures, and LS.

4.1. Solution representation. Representation is an important issue and it directly affects the quality of metaheuristics. Generally, data structures or representations used in different problems are not identical. In SA-LS, we use a $|N| \times |N|$ matrix which is known as a permutation encoding where every sequence is a string of numbers to represent the solutions of the HVRPSPD. Each row in the matrix provides the route and vehicle-type information of the solution. The first element in the row indicates the type of vehicle that is assigned to the route and the remaining elements indicate the sequence of customers that are visited by the vehicle. Figure

3	7	1	8		
1	2	9			
3	4	5	3	10	6

FIGURE 2. Representation of the HVRPSPD solution with 10 customers and three routes

2 illustrates an example for the solution representation of the HVRPSPD instance with 10 customers and three vehicle types. There are three routes in the solution. Vehicle type 3 visits customers 7, 1, and 8 for the first route, vehicle type 1 visits customers 2 and 9 for the second route, and vehicle type 3 visits customers 4, 5, 3, 10, and 6 for the third route. In this example, there are three vehicle types; however, only two are assigned to routes.

4.2. Generating the initial solution. Initial solutions are used to start search mechanisms. There are two common approaches to generate an initial solution: one is to generate it randomly and the other is to use specific heuristics developed for

the problem. Because heuristics obtain better quality solutions than a randomly generated approach, using a heuristic algorithm is mostly preferred to avoid the computational burden. In this study, we propose a heuristic approach to initialize a solution for the HVRPSPD. This approach, based on the giant tour and its partition, is a type of “route first-cluster second” approach, which is very common in VRP literature. After this approach was proposed by Beasley [8] for the VRP, it was successfully applied to variants of VRP, for example, by Golden et al. [23], Prins [47], Prins [48], and Goksal et al. [22].

The proposed initialization approach to generate the starting solution for the HVRPSPD consists of three main steps: Generate a giant tour, partition and check feasibility. These steps are described as follows:

For the first step, a giant tour is generated by solving the traveling salesman problem (TSP) in which the pickup and delivery demands of customers and the capacity restrictions of vehicles are omitted. We use the Concorde TSP solver to optimally solve the TSP, that is, to obtain an ordered set of customers $E = \{E_1, E_2, \dots, E_{|N|-1}\}$ over the given network.

For the second step, the giant tour along E is partitioned into routes. An acyclic auxiliary graph $H = (\hat{E}, F)$ is built to partition the giant tour. In this graph, $F = \{(i, j, k) : i < j, i, j \in N, k \in B\}$ is the set of arcs and $\hat{E} = \{0, 1, 2, \dots, |N| - 1\}$ is the set of nodes, which contains a dummy node 0 and the nodes 1 to $|N| - 1$ for customers E_1 to $E_{|N|-1}$. Arc (i, j, k) in the graph represents the vehicle of type k leaving the depot, visiting customers E_{i+1} to E_j , and returning to the depot. Weight $\hat{c}_{ijk}$ of arc (i, j, k) is calculated as in Equation (20) where M is a sufficiently large number.

$$\hat{c}_{ijk} = \begin{cases} f_k + \theta_k \left(l_{0, E_{i+1}} + \sum_{r=i+1}^{j-1} l_{E_r, E_{r+1}} + l_{E_j, 0} \right), \\ i < j, \max \left(\sum_{r=i+1}^j d_{E_r}, \sum_{r=i+1}^j p_{E_r} \right) \leq Q_k \\ M, \text{ otherwise} \end{cases} \quad (20)$$

An optimal partition of E is obtained by a minimum cost path from nodes 0 to $|N| - 1$ in H , where at most T_k arc (i, j, k) must be selected because there is an availability restriction for the vehicle of type k . This problem is called the shortest path problem with resource constraints [25]. Although it is an NP-hard problem, it can be solved quickly in practice using dynamic programming methods [48]. Thus, we use the following MIP formulation (SPPRC) to optimally partition E into routes. In this formulation, the binary decision variable $\hat{x}_{ijk} = 1$ if and only if arc (i, j, k) is in the solution, otherwise $\hat{x}_{ijk} = 0$. The SPPRC formulation is given as follows:

The objective function minimizes the total arc weights:

$$\min \hat{z} = \sum_{i \in N} \sum_{j \in N} \sum_{k \in B} \hat{c}_{ijk} \hat{x}_{ijk} \quad (21)$$

subject to,

$$\sum_{j \in N \setminus \{0\}} \sum_{k \in B} \hat{x}_{0jk} = 1 \quad (22)$$

$$\sum_{i \in N \setminus \{|N|-1\}} \sum_{k \in B} \hat{x}_{i, |N|-1, k} = 1 \quad (23)$$

that satisfy exactly one arc that leaves the origin and enters the destination, respectively. The following constraint yields exactly one arc that enters and leaves the intermediate nodes:

$$\sum_{i \in N} \sum_{k \in B} \hat{x}_{ijk} = \sum_{i \in N} \sum_{k \in B} \hat{x}_{jik}, \quad \forall j \in N \setminus \{0, |N| - 1\} \quad (24)$$

The following constraint guarantees that at most T_k arc (i, j, k) are selected in the optimal solution:

$$\sum_{i \in N} \sum_{j \in N} \hat{x}_{ijk} = T_k, \quad \forall k \in B \quad (25)$$

The following constraint defines the integrality conditions of the decision variables:

$$\hat{x}_{ijk} \in \{0, 1\}, \quad \forall i, j \in N, \forall k \in B \quad (26)$$

For the third step, after partitioning the giant tour into routes, every route is checked to determine whether the maximum load along the route exceeds the vehicle capacity. If a violation is observed in any route in terms of the vehicle capacity, that route is repaired by considering it as a VRPSPD with one vehicle. As was indicated in Section 3.3, the VRPSPD with one vehicle is a special case of the HVRPSPD. Thus, this problem is easily solved by the MIP formulation of the HVRPSPD considering the following settings: $N = N'$, $B = \{k^*\}$, $m = 1$, where N' is a node set that consists of a depot and nodes from the tour to be repaired and k^* is the type of vehicle on the route.

4.3. Neighborhood structures. The proposed SA-LS uses seven inter-route and four intra-route moving mechanisms to search the neighbors of a current solution. The neighborhood structures are briefly explained in this section.

The inter-route moving strategies are defined as follows and the illustrative examples for inter-route moving strategies are shown in Figure 3: *Shift(1,0)*: Node i from tour r_1 is transferred to tour r_2 . *Shift(2,0)*: Two adjacent nodes i and j from tour r_1 are transferred to tour r_2 . *Swap(1,1)*: Node i from tour r_1 and node j from tour r_2 are exchanged. *Swap(2,1)*: Two adjacent nodes, i and j , from tour r_1 are replaced by node k from tour r_2 . *Swap(2,2)*: Two adjacent nodes, i and j , from tour r_1 are exchanged by another two adjacent nodes, k and l , from tour r_2 . *Cross*: Arcs between nodes i and j from tour r_1 and between k and l from tour r_2 are removed. Then two arcs connecting nodes i and l and nodes k and j are inserted. *K-Shift*: Subset of consecutive nodes from tour r_1 is transferred to the end of tour r_2 .

The intra-route strategies are explained as follows and Figure 4 shows the examples of intra-route moving strategies: *Or-opt*: One, two, or three adjacent nodes from a tour are removed and inserted into another position of the same tour. *2-opt*: Two non-adjacent arcs are deleted and another two are added in such a manner that a new tour is generated. *Exchange*: Positions of two nodes i and j in a tour are exchanged. *Reverse*: Tour direction is reversed if there is a chance to reduce the maximum load along the tour.

SA-LS implements the best improvement strategy when searching within the solution space by a moving mechanism to select a neighbor of the current solution. Furthermore, it implements only feasible moves that do not exceed capacity restrictions. We conducted several tests on inter-route moving mechanisms to determine

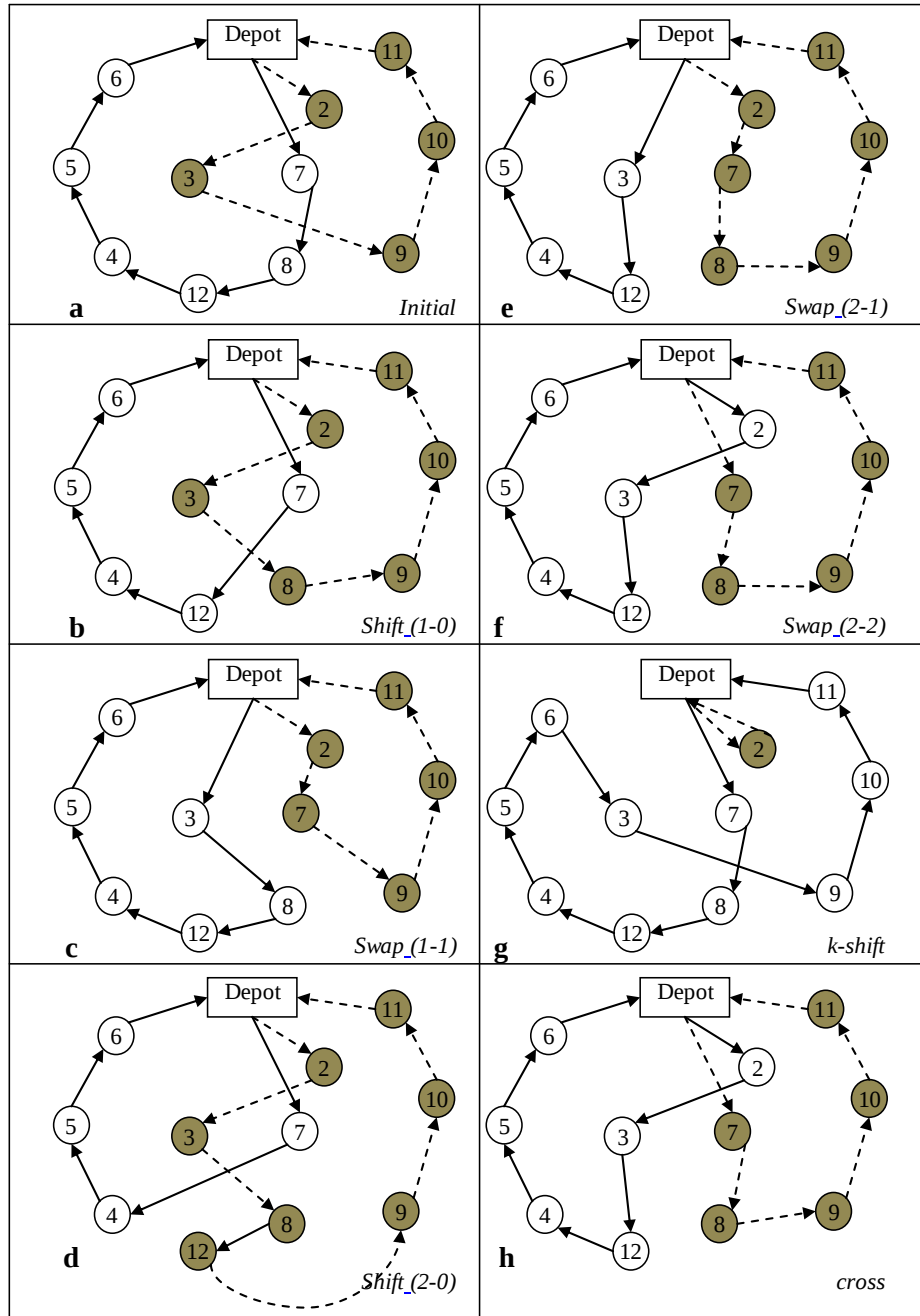


FIGURE 3. Inter-route neighborhood structures

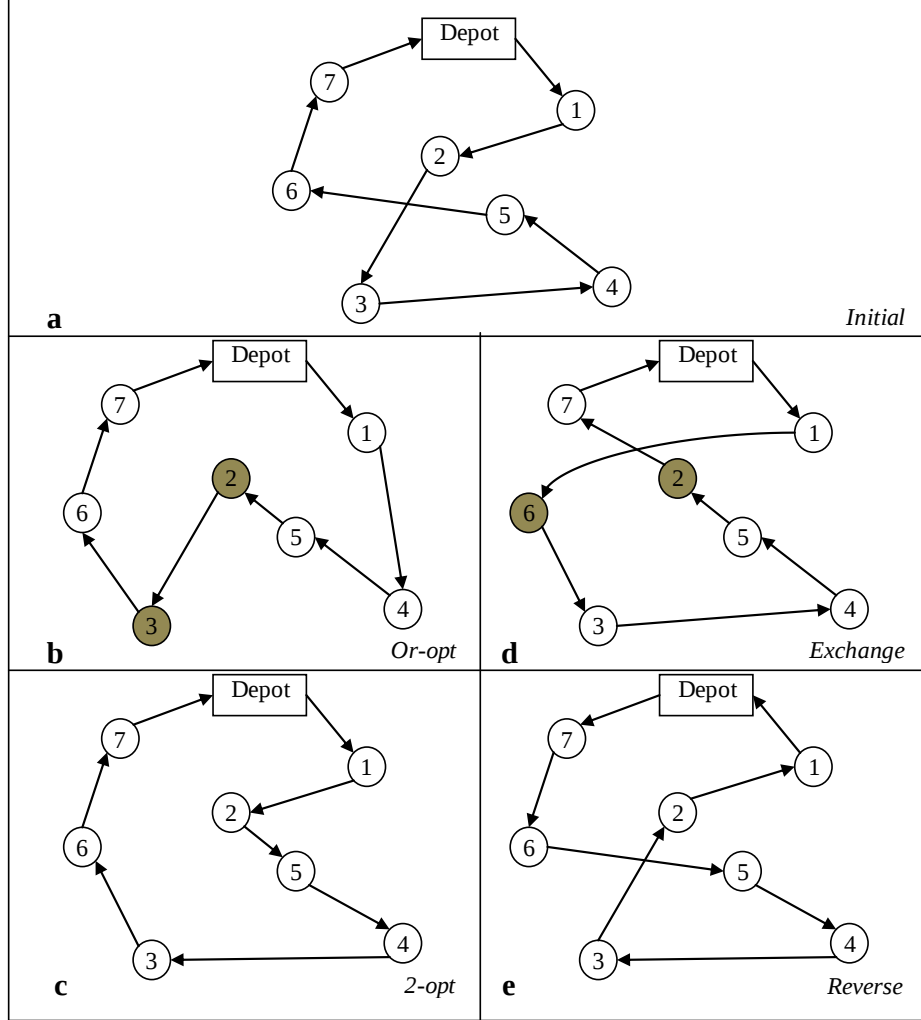


FIGURE 4. Intra-route neighborhood structures

which are more effective for the performance of the LS. This analysis shows that the inter-route moving strategies K -Shift, $Shift(2,0)$, and $Shift(1,0)$ (classified as Type-2) are more effective than $Cross$, $Swap(2,2)$, $Swap(1,1)$, and $Swap(2,1)$ (classified as Type-1) on the performance of the LS. Thus, regarding the behavior of each strategy we determined that Type-1 strategies are used to generate the best neighbor of the current solution in SA-LS, whereas Type-2 strategies are used to improve the solution obtained by the LS procedure of SA-LS. Moreover, the intra-route moving strategies (classified as Type-3) are used in the order of *Exchange*, *2-opt*, *Reverse*, and *Or-opt* at the end of each LS implementation of SA-LS.

4.4. Local search. The improvement of feasible solutions by an LS algorithm for a certain step of the search procedure has a significant impact on increasing the

effectiveness of metaheuristic algorithms. Solution approaches in which metaheuristics are used together with a simple LS, other metaheuristics or exact solution algorithms are called hybrid algorithms. Talbi provided a classification scheme for hybrid metaheuristic algorithms [59]. In this study, the SA is hybridized with simple LS, called SA-LS, to obtain a good solution through the interactive use of two algorithms together within a reasonable solution time.

The pseudo code of the LS is shown in Figure 5 (lines 00-28). The LS procedure simply takes a feasible solution (S_{LS}) as an input and attempts to improve it. If there is no improvement, it returns the input solution (S_{LS}) as the output. In the beginning of LS one moving strategy was randomly chosen from the neighborhood structures of Type-1 to obtain a new solution at a certain iteration of the SA-LS algorithm (lines 01-03). If the objective function value of the new solution is better than the current solution (line 04), then the new solution is set as the current solution (line 06). Next, a random number between zero and one is generated uniformly (line 07). If the random number is greater than p , then the following two steps (lines 11-16) are repeated while the number of iterations without any improvement in the objective function is less than a certain number (this number is set as 5 for the preliminary trials): 1) A new solution is obtained by a moving strategy chosen randomly among the neighborhood structures of Type-2 and 2) if the objective value of the new solution is less than the current solution, then the new solution is set to the current solution. In this way, diversification is performed by jumping to different points in the solution space with the neighborhood structures of Type-1. Additionally, intensification is performed by searching a particular subspace with the neighborhood structures of Type-2. Finally, the neighborhood structures of Type-3 are all applied in the given order (lines 18-25). If the objective value of the new solution is worse than the current solution, then the LS procedure returns the input solution as the output without any update (line 28).

4.5. Steps of SA-LS. The overall steps of the methodology proposed for the HVRPSPD are shown in Figure 6 (lines 00-16) and summarized as follows: SA-LS starts with an initial solution S (line 01). The current temperature T is set to T_0 (line 02). The main steps of SA-LS between lines 04 and 14 are repeated until a stopping condition is met. At each temperature of SA-LS, the current feasible solution is submitted to the LS procedure as an input to improve the solution (line 05). Within the LS procedure, a new solution S_{new} is generated in the neighborhood of the current solution S using a moving mechanism, which is randomly selected within a group of strategies (called Type-1 moving strategies). If the new solution S_{new} is better than the current solution S , then the search process continues with simple LS using a moving mechanism, which is randomly chosen each time within a group of strategies (called Type-2 moving strategies), and updates the new solution S_{new} . The LS continues until no improvement is obtained for five successive iterations. At the end of the LS, four moving strategies (called Type-3 moving strategies) are implemented in a predetermined order for S_{new} , and S_{new} is updated.

Eventually, the LS procedure returns a new solution if the objective function is improved, and if there is no improvement, the current solution is returned to SA-LS ($S_{new} := LS(S)$). If the new solution is better than the current solution, that is, $f(S_{new}) < f(S)$, then it is accepted as the current solution (line 06), otherwise it is accepted with the probability of $e^{(-\Delta S/T_{itr})}$ as the current solution, where T_{itr} is the current temperature. ΔS is the relative deviation of the quality of new solution from the current solution and is calculated as $[(f(S_{new}) - f(S)) / f(S)] \times$

Algorithm: LS algorithm for SA.

Input: Feasible HVRPSPD solution (S_{LS}).

Output: Feasible HVRPSPD solution (S_{LS}).

```

00. BEGIN
01.    $N_1 := \{Swap(1,1), Swap(2,1), Swap(2,2), Cross\};$ 
02.   Choose a random move  $M^* \in N_1$ ;
03.   Find the best neighbor ( $S'_{LS}$ ) of  $S_{LS}$  with move  $M^*$ ;
04.   If  $f(S'_{LS}) < f(S_{LS})$  then,
05.   {
06.      $S_{LS} := S'_{LS}$ ;
07.     Generate random number  $U \in [0,1]$ ;
08.     If  $U \geq p$  then;
09.     {
10.       Repeat while the # of iterations without improvement is less than 5.
11.       {
12.          $N_2 := \{Shift(1,0), Shift(2,0), K-Shift\};$ 
13.         Choose a random move  $M^* \in N_2$ ;
14.         Find the best neighbor ( $S'_{LS}$ ) of  $S_{LS}$  with the move  $M^*$ ;
15.         If  $f(S'_{LS}) < f(S_{LS})$  then,  $S_{LS} := S'_{LS}$ ;
16.       }
17.     }
18.     Find the best neighbor ( $S'_{LS}$ ) of  $S_{LS}$  with the move Exchange;
19.     If  $f(S'_{LS}) < f(S_{LS})$  then,  $S_{LS} := S'_{LS}$ ;
20.     Find the best neighbor ( $S'_{LS}$ ) of  $S_{LS}$  with the move 2-opt;
21.     If  $f(S'_{LS}) < f(S_{LS})$  then,  $S_{LS} := S'_{LS}$ ;
22.     Find the best neighbor ( $S'_{LS}$ ) of  $S_{LS}$  with the move Reverse;
23.     If  $f(S'_{LS}) < f(S_{LS})$  then,  $S_{LS} := S'_{LS}$ ;
24.     Find the best neighbor ( $S'_{LS}$ ) of  $S_{LS}$  with the move Or-opt;
25.     If  $f(S'_{LS}) < f(S_{LS})$  then,  $S_{LS} := S'_{LS}$ ;
26.   }
27.   Else  $S_{LS} := S'_{LS}$ ;
28. END

```

FIGURE 5. Pseudo code of the LS

100. The temperature is updated for every iteration according to the geometric cooling schedule $T_{itr} = 0.95T_{itr-1}$ (lines 07-13). SA-LS stops whenever the final temperature T_f is achieved or the best solution is not improved for n successive iterations, and reports the final solution (line 15). For SA-LS, SA is implemented to search for good vehicle routes in the solution space, whereas LS is used to improve the best vehicle routes determined during the search process of SA.

5. **Experimental study.** The results of computational experiments are outlined in this section and organized in six phases. In the first phase, the performance of SA-LS in comparison with the proposed MIP formulation on small and medium-size HVRPSPD problems is evaluated. The second phase investigates the performance of SA-LS in comparison with NN and CWS, on medium and large-size HVRPSPD

Algorithm: SA-LS algorithm for the HVRPSPD.
Input: HVRPSPD data and parameter settings for the algorithm.
Output: Feasible HVRPSPD solution (S).

```

00. BEGIN
01.   Generate a feasible initial solution  $S$ ;
02.   Set the starting temperature ( $T := T_0$ );
03.   Repeat until the stopping criterion is met.
04.   {
05.       Improve the solution  $S$  with LS ( $S_{new} := LS(S)$ );
06.       If  $f(S_{new}) \leq f(S)$  then,  $S := S_{new}$ ;
07.       Else If  $f(S_{new}) > f(S)$  then,
08.       {
09.           Calculate  $\Delta = 100 * (f(S_{new}) - f(S)) / f(S)$ ;
10.           Generate random number  $U \in [0, 1]$ ;
11.           If  $U \leq (e^{-\Delta/T})$  then,  $S := S_{new}$ ;
12.           Update  $T$ ;
13.       }
14.   }
15.   Stop and report  $S$ ;
16. END

```

FIGURE 6. Pseudo code of SA-LS

instances. The third phase attempts to determine the performance of SA-LS on the HVRP instances, which is a special case of the HVRPSPD. The fourth phase compares the performance of SA-LS on the VRPSPD instances, which is another special case of the HVRPSPD. The fifth phase compares the SA-LS with the results obtained in the study of Avci and Topaloglu [5]. And the last phase shows the convergence behavior of the SA-LS due to the change in different parameters. The section for computational experiments ends with some managerial insights for the companies in logistics sector. The OPL language and state-of-the-art LP/MIP solver engine CPLEX® (version 12.3) were used to model and solve the MIP models and their relaxations. The heuristic algorithms, which used the CPLEX Concert Technology, were coded in C++. Computational experiments were run on a PC with a 2.67 GHz Intel Core i5 750 CPU and 2 GB of RAM.

5.1. Test problems. Two HVRP problem sets were used to generate the HVRPSPD test problems. The first HVRP problem set was derived by Taillard [58] from Golden et al.'s [23] problems. This set included four instances with 50 customers, two instances with 75 customers, and two instances with 100 customers. The second HVRP test set was derived by Liu and Shen [36] from Solomon's [54] problems. This set consisted of 80 instances with 100 customers. The HVRPSPD test problems could easily be generated from the HVRP instances using demand separation approaches.

We used two demand separation techniques to generate the pickup and delivery demands of customers in each HVRPSPD test problem. The first was proposed by

TABLE 1. Characteristics of the test problems

Problems	n	# of instances
Small and medium-size	20	$4 \times (8^a + 18^b)$
	25	$4 \times (8^a + 18^b)$
	30	$4 \times (8^a + 18^b)$
	35	$4 \times (8^a + 18^b)$
	40	$4 \times (8^a + 18^b)$
Large-size	50	4×4^a
	75	4×2^a
	100	$4 \times (2^a + 18^b)$

^aTaillard's [58] test problems, ^bLiu and Shen's [36] test problems.

Salhi and Nagy [51]. In this technique, $r_i = \min\{x_i/y_i, y_i/x_i\}$ is calculated for each customer and the original demands are split using this ratio. For instance, let q_i be the original demand of the customer i . Then the pickup and delivery demands are calculated by $p_i = (1 - r_i)q_i$ and $d_i = r_iq_i$, respectively. These problems are called *Type X*. In the same manner, another problem type (referred to as *Type Y*) is obtained by shifting each demand of the customer to the next customer's demand. The second separation technique is proposed in this study. This technique splits the original demands using the so-called "golden ratio". In mathematics, if we use the golden ratio to divide a line segment $|AB|$, this segment should be divided by point C such that $|CB|/|AC| = |AB|/|CB| = \varphi$, where φ is a constant ratio, which is an irrational number, and $\varphi = (1 + \sqrt{5})/2$. Regarding these definitions, if i is odd and $|AB| = q_i$, $|AC| = d_i$ and $|CB| = p_i$, then the original demand is divided into pickup and delivery demands as $p_i = \lfloor q_i/\varphi \rfloor$, $d_i = q_i - p_i$. If i is even and $|AB| = q_i$, $|AC| = p_i$ and $|CB| = d_i$, then the original demand is divided into the delivery and pickup demands as $d_i = \lfloor q_i/\varphi \rfloor$, $p_i = q_i - d_i$. These problems are called *Type W* and we obtain *Type Z* problems as explained above for *Type Y* problems. As a result, 104 (4) large-size test instances were generated using 26 original HVRP problems and four separation procedures (*Type X*, *Y*, *W*, and *Z*).

In addition to these problems, 520 (5) small and medium-size instances were obtained using the first 20, 25, 30, 35, and 40 customers in each of the 104 problems. The main characteristics of our test problems are presented in Table 1.

5.2. Evaluating the performance of SA-LS with the MIP formulation. In the first phase, we investigated the performance of the proposed hybrid heuristic approach, SA-LS, on small and medium-size problems.

Based on our preliminary experiments, we set the following parameter values in SA-LS: the initial temperature is 380, in which an inferior solution (inferior by 40% relative to the current solution) is accepted with a probability of 0.90; the cooling rate is set to 0.95; the final temperature is 0.15, such that a solution that is inferior by 1% relative to the current solution is accepted with a probability of 0.001; and the LS procedure within SA-LS is used with the probability of 0.6. SA-LS stops whenever the temperature reaches the final temperature or the best solution is not improved for successive iterations. Each instance was run five times by the proposed

TABLE 2. Computational results of SA-LS on small and medium-size test problems

n	Type	SA-LS				MIP Formulation			Math-LS*		
		Gap%	Imp%	#OpSol	CPU (s)	Gap%	#OpSol	CPU (s)	Gap%	#OpSol	CPU (s)
20	X	4.50	5.50	6	1.90	1.90	20	306.01	5.38	8	16.53
	Y	4.40	5.30	7	2.00	1.51	20	321.93	5.49	7	13.6
	W	4.00	7.10	7	1.90	0.49	23	290.63	4.38	12	13.24
	Z	6.30	3.30	3	1.80	3.75	13	295.44	5.23	11	12.33
	(Total)Average	4.80	5.30	(23)	1.90	1.91	(76)	303.50	5.12	(38)	13.93
25	X	5.10	3.80	3	3.50	2.39	13	1174.24	9.35	5	18.48
	Y	5.00	3.50	2	2.70	2.38	15	1362.22	8.87	7	17.52
	W	6.10	3.80	3	3.00	2.65	14	644.11	7.21	7	16.74
	Z	8.10	3.20	1	3.00	2.78	13	580.30	9.69	7	15.25
	(Total)Average	6.08	3.58	(9)	3.05	2.55	(55)	940.22	8.78	(26)	17
30	X	8.75	4.30	2	5.30	5.91	7	2133.27	13.06	1	21.95
	Y	11.20	2.80	4	4.10	7.08	5	2154.08	12.95	2	28.12
	W	6.55	6.30	0	5.90	6.26	4	1672.95	11.11	5	22.58
	Z	6.18	4.90	3	5.20	4.39	9	1694.82	11.56	4	18.93
	(Total)Average	8.17	4.58	(9)	5.13	5.91	(25)	1913.78	12.17	(12)	22.89
35	X	8.32	11.70	0	7.90	10.76	0	3984.36	26.08	1	32.78
	Y	9.29	5.72	0	7.50	9.00	1	3724.80	16.14	0	31.31
	W	7.97	3.99	1	7.40	9.90	4	3741.48	14.92	0	25.42
	Z	9.88	3.67	1	8.20	9.59	6	3242.11	13.69	1	26.26
	(Total)Average	8.87	6.27	(2)	7.75	9.81	(11)	3673.19	17.71	(2)	28.94
40	X	16.60	5.44	0	11.70	20.54	0	5833.66	27.66	0	41.16
	Y	18.10	8.10	0	11.60	19.06	0	6216.06	19.28	0	46.15
	W	9.08	4.93	0	11.30	13.82	1	5685.47	15.68	0	28.66
	Z	8.36	6.66	2	11.50	10.68	6	4810.54	11.87	0	28.79
	(Total)Average	13.04	6.28	(2)	11.53	16.02	(7)	5636.43	18.62	(0)	36.19
Overall(Total)		8.19	5.20	(45)	5.87	7.24	(174)	2493.42	12.48	(78)	23.79

*Math-LS is the algorithm proposed by Kececi et al. [32].

heuristic with a different random number of seeds and the best of five runs for each instance was considered as the solution of the heuristic; the computation time of MIP formulation was limited to 2 CPU hours.

Table 2 summarizes the computational results of SA-LS on small and medium-size problems. The first two columns of the table show the number of customers in a HVRPSD instance (n) and the demand separation strategy. The subsequent four columns show the average value of the percentage gap ($Gap\%$), average improvement of the initial solution ($Imp\%$), number of problems solved optimally ($\#OpSol$) with SA-LS, and average computation time (CPU). The next three columns show the lower bound percentage gap obtained by the MIP formulation ($Gap\%$), number of problems solved optimally ($\#OpSol$) with MIP and average computation time of the MIP formulation (CPU). Finally, the last three columns give the $Gap\%$, $\#OpSol$ and CPU values for the matheuristic algorithm proposed by Kececi et al. [32]. It should be noted that the average and maximum percentage gaps of the heuristic solutions are calculated by considering optimal solutions, or if unknown, the best lower bounds, which are obtained by solving the MIP formulation with CPLEX for a maximum of 2 CPU hours.

From Table 2, we can observe that SA-LS obtained good results (8.19%, on average, for the optimal solution or the best lower bound) in a very short computation time. The average computation time of SA-LS was 5.87 s, whereas the MIP formulation required 2,493 s and Math-LS 23.79 s, on average, to solve HVRPSD instances. The average percentage gap of SA-LS was 8.19% and 12.78% for Math-LS; however, this value reduced to 7.24% for the MIP formulation. Moreover, the

performance of the MIP formulation quickly degenerated for instances with 35 and 40 customers. SA-LS optimally solved 45 out of 174 instances and MathH-LS solved 78 instances, which were optimally solved by the MIP formulation. Additionally, SA-LS improved upper bounds obtained by the MIP formulation for 216 instances. These results show that SA-LS is superior to the MIP formulation and MathH-LS as well, especially in terms of the solution time. MathH-LS can optimally solve more instances than SA-LS, however SA-LS can obtain better objective function value on the instances which could not be solved optimally, since it gives lower gap. Moreover, SA-LS can improve the initial solution by 5.20%, on average, and this improvement is between 2.80% and 11.70%.

5.3. Investigating the performance of SA-LS with simple heuristics. In the second phase, we compare the performance of SA-LS with NN and CWS on large-size instances in terms of solution quality and solution time. In this study, we adapted the NN and CWS algorithms to solve HVRPSPD. The details of both heuristics are given in the Appendix.

Table 3 summarizes the computational results of SA-LS in comparison with the NN and CWS algorithms. The first two columns of the table are the same as the previous tables. The next three columns show the average percentage deviation (*Dev%*) of the best solution determined by SA-LS from the solutions determined by the NN and CWS algorithms and the best lower bounds, which are obtained by solving the MIP formulation with CPLEX for a maximum of 2 CPU hours, respectively. In comparison with the MIP model, the best lower bounds are used since MIP model may not provide an integer feasible solution for the large-size test instances. Finally, the last column contains the average computation time (*CPU*) of SA-LS in seconds.

Observing Table 3, we conclude that SA-LS is superior to NN and CWS. SA-LS obtained, on average, 53% and 16% better solutions than those obtained by the NN and CWS algorithms, respectively. The solution time of the NN and CWS algorithms was very short (less than 1 s) because they are simple constructive heuristics; hence, they are not shown in the results. However, SA-LS outperformed the simple heuristics in terms of the solution quality within a reasonable computation time. It is remarkable to note that SA-LS achieved good solutions only within 103 s, on average. As compared to the lower bounds obtained by the MIP formulation, the algorithm performs solution quality less than 15 percent in average. The minimum and maximum deviations from the lower bounds of MIP formulation are 5.57 and 17.55. The gap value might be quite big because the MIP formulation is not as tight as it is necessary. With the addition of some valid inequalities, the formulation could be made tighter and it gives higher lower bounds. It is interesting to observe that both 75 and 100 nodes instances have similar average gap, which is approximately 14%, and also the test instance, in which the maximum gap value is obtained, is 75 nodes instance rather than 100 nodes instance. The test data might be decisive of this issue.

5.4. Comparison of SA-LS with several heuristics on HVRP instances.

As is known, the HVRPSPD is a generalization of the HVRP. Thus, the proposed solution methodology can be used to solve HVRP instances by considering the original demand of each customer as the delivery demand ($d_i = q_i$) and the pickup demand as zero ($p_i = 0$). To investigate the performance of SA-LS on the HVRP, we solved the well-known HVRP test instances of Golden et al. [23] and compared

TABLE 3. Comparison of SA-LS with NN and CWS on large-size test problems

n	Type	Dev% _{NN}	Dev% _{CWT}	Dev% _{MIPLB}	CPU (s)
50	X	-36.69	-14.23	6.71	19.62
	Y	-35.79	-18.23	5.57	20.75
	W	-42.96	-7.83	8.79	23.65
	Z	-42.43	-9.62	9.26	24.42
	Average	-39.47	-12.48	7.58	22.11
75	X	-58.39	-16.50	11.96	72.23
	Y	-56.69	-18.08	11.79	58.14
	W	-59.49	-18.32	15.11	82.00
	Z	-59.14	-12.37	17.55	57.12
	Average	-58.43	-16.32	14.10	67.37
100	X	-60.04	-20.91	15.65	210.87
	Y	-60.46	-20.23	16.17	220.00
	W	-63.42	-16.71	12.20	223.16
	Z	-63.30	-16.32	14.95	221.59
	Average	-61.81	-18.54	14.74	218.90
Overall Average		-53.23	-15.78	12.14	102.80

the results with several heuristics proposed in the HVRP literature. In these test instances, there were 20 problems with varying size 20 to 100. We excluded the eight instances (problems 1, 2, and 7 to 12) that used non-Euclidean distance. Golden et al. [23] considered the unlimited number of vehicles of each type for their vehicle fleet mix (VFM) problem. Because our algorithm is designed to solve the HVRP with a limited number of vehicles of each type, we calculated the number of available vehicles $T_k, \forall k \in B$ for each test problem as $\min_{k \in B} \{Q_k\} / \sum_{i \in N} q_i$. Additionally, there was no variable cost θ_k per distance unit of vehicle $k \in B$. Hence, we assumed that $\theta_k = 1, \forall k \in B$.

Table 4 presents the results of SA-LS on HVRP instances. In this table, the first and second columns represent the name and size of the HVRP instances, respectively. Next each subsequent column contains the best objective function values obtained by the corresponding author(s). The last three columns contain the best objective function value obtained by SA-LS, percentage deviation ($Dev\%$) from the best objective calculated as $100(f_{SA-LS} - f_{Best}) / f_{Best}$, and average computation time per run in seconds (CPU) over the five runs. The text in bold indicates the minimum objective function value (f_{Best}) in a row.

The results for the HVRP test instances are quite promising. SA-LS obtained, on average, 1.36% (average of the $Dev\%$ column) worse than the best solutions in the literature found so far. The average computation time was 16.56 s, and it varied between 0.46 s and 60.70 s. According to the results, SA-LS can be a preferable method to solve HVRP instances successfully.

TABLE 4. Comparison of SA-LS with HVRP test instances

Inst.	n	Salhi and Rand [52]	MRPERT	TSVM	Taillard [58]	Gendreau et al. [20]	Wassan and Osman [65]	Renaud and Boctor [49]	Choi and Tcha [14]	Brandao [10]	Prins [48]	Liu et al. [37]	Penna et al. [45]	Subramanian et al. [56]	Simeonova et al. [53]	SA-LS	Dev%	CPU (s)
G.3	20	1003	971.95	43	961.03	961.03	961.03	963.61	961.03	961.03	961.03	961.03	961.03	961.03	961.03	961.03	0.00	0.46
G.4	20	6447	6447.80	6445.10	6437.33	6437.33	6437.33	6437.33	6437.33	6437.33	6437.33	6437.33	6437.33	6437.33	6437.33	6437.33	0.00	0.87
G.5	20	1015	1015.13	1009.15	1008.59	1007.05	1007.05	1007.96	1007.05	1007.05	1007.05	1007.05	1007.05	1007.05	1007.05	1007.05	0.00	0.47
G.6	20	6516	6516.56	6516.56	6516.47	6516.47	6516.47	6537.74	6516.47	6516.47	6516.47	6516.47	6516.47	6516.47	6516.47	6516.47	0.00	0.86
G.13	50	2493	2462.01	2471.07	2413.78	2408.41	2422.10	2406.43	2406.36	2406.36	2406.36	2406.36	2408.41	2406.36	2406.36	2412.36	0.25	5.21
G.14	50	9153	9141.69	9125.65	9119.03	9119.03	9119.86	9122.01	9119.03	9119.03	9119.03	9119.03	9119.03	9119.03	9119.03	9121.71	0.03	11.86
G.15	50	2623	2600.31	2606.72	2586.37	2586.37	2586.37	2618.03	2586.37	2586.84	2586.37	2586.37	2586.37	2586.37	2586.37	2588.77	0.09	7.02
G.16	50	2765	2745.04	2745.01	2741.50	2741.50	2730.08	2761.96	2720.43	2728.14	2729.08	2724.22	2724.22	2720.43	2720.43	2779.72	2.18	6.97
G.17	75	1767	1766.81	1762.05	1747.24	1749.50	1755.10	1757.21	1744.83	1736.09	1746.09	1734.53	1734.53	1734.53	1734.53	1770.09	2.05	31.30
G.18	75	2439	2439.40	2412.56	2373.63	2381.43	2385.52	2413.39	2371.49	2376.89	2369.65	2369.65	2371.48	2369.65	2369.65	2434.78	2.75	16.43
G.19	100	8751	8704.20	8685.71	8661.81	8675.16	8659.74	8687.31	8664.29	8667.26	8665.12	8662.94	8662.86	8661.81	8667.26	9173.19	5.93	60.70
G.20	100	4187	4166.03	4188.73	4047.55	4086.76	4061.64	4094.54	4039.49	4048.09	4044.78	4038.46	4037.9	4032.81	4038.45	4153.74	3.00	56.60
Avg.																	1.36	16.56

MRPERT: Modified version of procedure RPERT proposed by Salhi and Rand [52], TSVFM: Tabu search for the vehicle fleet mix problem (VFM).

TABLE 5. Comparison of SA-LS with VRPSPD test instances

Instance	n	Q	Salhi and Nagy [51]	Goksal et al. [22]	SA-LS				
					Min	Avg	Max	Dev%	CPU (s)
CMT1X	50	160	601	466.77	487	494.4	503	4.33	2.65
CMT1Y	50	160	603	466.77	467	473.0	486	0.05	2.48
CMT2X	75	140	873	668.77	698	709.0	726	4.37	4.58
CMT2Y	75	140	924	663.25	672	683.6	699	1.32	5.32
CMT3X	100	200	923	721.27	731	740.8	753	1.35	19.00
CMT3Y	100	200	923	721.27	714	721.6	739	-1.01	21.66
CMT4X	150	200	1178	852.46	884	891.0	899	3.70	50.35
CMT4Y	150	200	1178	852.46	849	853.2	857	-0.41	50.06
CMT5X	199	200	1509	1029.25	1115	1134.0	1151	8.33	86.07
CMT5Y	199	200	1477	1029.25	1021	1042.8	1060	-0.80	90.89
CMT11X	120	200	1500	833.92	916	937.2	978	9.84	34.47
CMT11Y	120	200	1500	830.39	785	803.0	835	-5.47	38.87
CMT12X	100	200	820	644.7	674	700.0	725	4.54	14.87
CMT12Y	100	200	873	659.52	632	655.0	671	-4.17	15.02
Overall Avg.			1063.00	745.72	760.36	774.19	791.57	1.86	31.16

5.5. Comparison of SA-LS with VRPSPD instances. To investigate the performance of SA-LS on the VRPSPD, we solved the test instances of Salhi and Nagy [51] and compared the best results in the VRPSPD literature. In these test instances, there are 14 problems with varying size 50 to 199. To adapt the test problems to be solved by our proposed algorithm, we set $|B| = 1$ and $\theta_k = 1, f_k = 0, T_k = +\infty, Q_k = Q$ for each $k \in B$.

Table 5 presents the results of SA-LS on VRPSPD instances. In this table, the first and second columns represent the name and size of the VRPSPD instances, respectively. The third column is the capacity of the vehicle in the problem. Next each subsequent column contains the best objective function values reported by the corresponding author(s). The last five columns contain the minimum, average, maximum and the best objective function value obtained by SA-LS over the five runs; the percentage deviation ($Dev\%$) from the best objective calculated as $100(f_{SA-LS} - f_{Best})/f_{Best}$, and the average computation time per run in seconds (CPU) over the five runs. The text in bold indicates the minimum objective function value (f_{Best}) in a row.

The results for the VRPSPD test instances are quite promising. SA-LS obtained, on average, 1.86% (average of the $Dev\%$ column) worse than the best solutions in the literature found so far. The average computation time was 31.16 s, and it varied between 2 s and 90 s. According to the results, SA-LS can be a preferable method to solve VRPSPD instances successfully.

5.6. Comparison of SA-LS with the results of Avci and Topaloglu [5]. Avci and Topaloglu [5] are proposed a hybrid metaheuristic algorithm to solve the HVRPSPD. In their study they compare their algorithm with several metaheuristic algorithms. They conduct their experimental computations on randomly generated test instances. To investigate the performance of SA-LS on the HVRPSPD, we also solved the test instances of Avci and Topaloglu [5] and compared with their best results.

In these test instances there are two sets, where each set has 14 problems. The first set has small and medium size problems varying in size 10 to 100. The second set has medium and large size problems varying in size 150 to 550.

TABLE 6. Comparison of SA-LS on set 1 test instances

Intance	n	B	Avci and Topaloglu [5]			SA-LS				
			Min	Avg	CPU (s)	Min	Avg	Max	Dev%	CPU (s)
1	10	2	620.2	620.2	17.2	607.7	618.3	626.1	-2.02	0.0
2	10	2	588.5	588.5	14.7	585.6	587.5	590.7	-0.49	0.0
3	15	3	445.1	445.1	22.7	415.3	433.1	445.5	-6.71	0.1
4	15	4	437.1	437.1	24.5	417.1	435.3	441.5	-4.57	0.1
5	20	3	494	498.9	27.1	480.3	493.3	503.4	-2.77	0.2
6	20	4	542.7	551.9	26.7	539.4	561.8	578.1	-0.60	0.2
7	35	3	1108.2	1123.4	56.6	1126.6	1140.2	1158.3	1.66	1.0
8	35	3	1586.5	1601.2	54.7	1614.5	1673.8	1711.6	1.76	0.7
9	50	3	964.4	990.2	91.4	943.9	948.3	960.5	-2.12	2.3
10	50	2	1197.7	1228.6	95.8	1177.7	1192.2	1206.3	-1.67	2.0
11	75	3	1642.2	1673.9	143.8	1538.4	1572.9	1609.8	-6.32	7.4
12	75	2	973.1	1002.5	164.9	969.5	982.1	1007.3	-0.37	6.9
13	100	2	1299.5	1353.5	288.5	1262.3	1279.7	1311.0	-2.86	22.7
14	100	2	1658.2	1678.2	310.3	1525.6	1545.0	1580.0	-8.00	19.2
Overall Avg.			968,4	985,2	95.6	943.1	961.7	980.7	-2.51	4.5

Table 6 presents the results of SA-LS on set 1 instances of Avci and Topaloglu [5]. In this table, the first column represents the name of problem. The second and third columns are respectively the size and the number of vehicle types in the problem. Next three columns contain the minimum and average objective function values and average *CPU* times over several runs, reported by Avci and Topaloglu [5]. The last five columns contain the minimum, average, maximum and the best objective function value obtained by SA-LS over the five runs; the percentage deviation (*Dev%*) from the best objective calculated as $100(f_{SA-LS} - f_{Best})/f_{Best}$, and the average computation time per run in seconds (*CPU*) over the five runs. The text in bold indicates the minimum objective function value (f_{Best}) in a row.

The results of SA-LS on small size test instances show that the SA-LS surpass the metaheuristic algorithm proposed by Avci and Topaloglu [5]. The SA-LS obtained, on average, 2.51% (average of the *Dev%* column) better objective value than Avci and Topaloglu [5]. The 12 out of 14 instances are improved by the SA-LS.

Table 7 presents the results of SA-LS on set 2 instances of Avci and Topaloglu [5]. In this table, all columns are same as in Table 6.

The results of SA-LS on large size test instances also show that the SA-LS outperform the metaheuristic algorithm proposed by Avci and Topaloglu (2016). The SA-LS obtained, on average, 19.82% (average of the *Dev%* column) better objective value than Avci and Topaloglu [5]. All instances are improved by the SA-LS. In some instances the improvement is higher than 20%.

5.7. Convergence behavior of SA-LS. The parameters of SA-LS are optimized through the Taguchi method to solve effectively HVRPSPD instances. The parameter values set in the algorithm are already mentioned in the Section 5.2. In this section we give information about the impact of parameters on the performance of the SA-LS. Thus, we run SA-LS on *Type X* test instance coded LS.C201A with parameter values: 760, 570, 380, 190 (inferior by %80, %60, %40, %20 relative to the current solution) for the initial temperature; 0.45, 0.3, 0.15, 0.075 (inferior by %3, %2, %1, %0.5 relative to the current solution) for the final temperature; 0.99, 0.95, 0.90, 0.85 for the cooling rate.

TABLE 7. Comparison of SA-LS on set 2 test instances

Intance	n	B	Avci and Topaloglu [5]			SA-LS				
			Min	Avg	CPU (s)	Min	Avg	Max	Dev%	CPU (s)
1	150	3	1499.4	1624.5	592.5	1468.4	1493.1	1520.3	-2.07	157.8
2	150	3	2144.8	2152.5	548.9	2105.9	2168.6	2241.4	-1.81	118.1
3	200	3	3673.1	3688.8	831.6	3188.5	3302.3	3335.5	-13.19	116.0
4	200	2	2485.3	2682.5	919.4	2169.6	2181.8	2207.9	-12.70	313.8
5	250	3	2639.6	2810.0	1448.1	2373.7	2428.6	2541.9	-10.08	483.9
6	250	2	2549.8	2605.6	1521.3	2192.4	2219.6	2272.0	-14.02	311.8
7	300	3	3205.0	3431.1	2440.4	2502.0	2523.7	2549.2	-21.94	775.7
8	300	2	3252.8	3364.6	2518.2	2459.7	2533.5	2573.6	-24.38	908.8
9	350	3	3457.9	3637.0	3770.1	3270.1	3380.9	3434.2	-5.43	2385.5
10	350	2	3760.9	3897.5	3988.7	2262.0	2332.3	2409.0	-39.85	1766.4
11	400	2	5809.5	6018.9	5662.4	3188.3	3236.3	3276.6	-45.12	4466.4
12	400	2	4045.9	4447.0	5791.5	2871.9	2964.2	3048.6	-29.02	3306.5
13	500	2	11008.8	12062.5	7200.0	7889.3	7948.0	8056.6	-28.34	7789.4
14	550	2	12762.0	13046.3	7200.0	8987.9	9041.7	9089.8	-29.57	7620.6
Overall Avg.			4449.6	4676.3	3173.8	3352.1	3411.1	3468.3	-19.82	2180.0

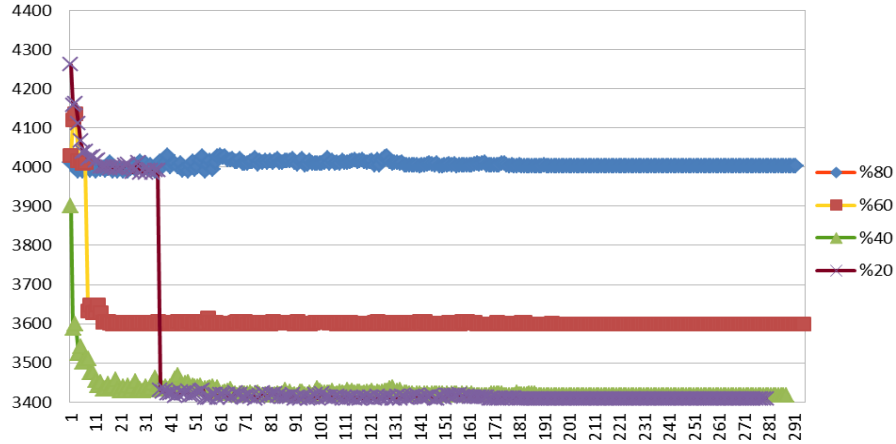


FIGURE 7. The convergence of SA-LS with different initial temperature values

Figure 7, Figure 8 and Figure 9 show the convergence of SA-LS with different initial temperature, final temperature and cooling rate values, respectively. As seen from the figures, the SA-LS slowly converges to better solutions when the parameters are set to 380, 0.15 and 0.95. This result supports that the parameter values used in the SA-LS help the algorithm to avoid local optima.

5.8. Managerial insights of the results. In today's competitive environment, logistics, transportation and distribution are at the center of managers' concern. Transportation constitutes one-third of the logistics costs in the supply chain. In this regard companies try to create competitive advantages by strengthening their transportation and distribution activities and as well as making their fleet more

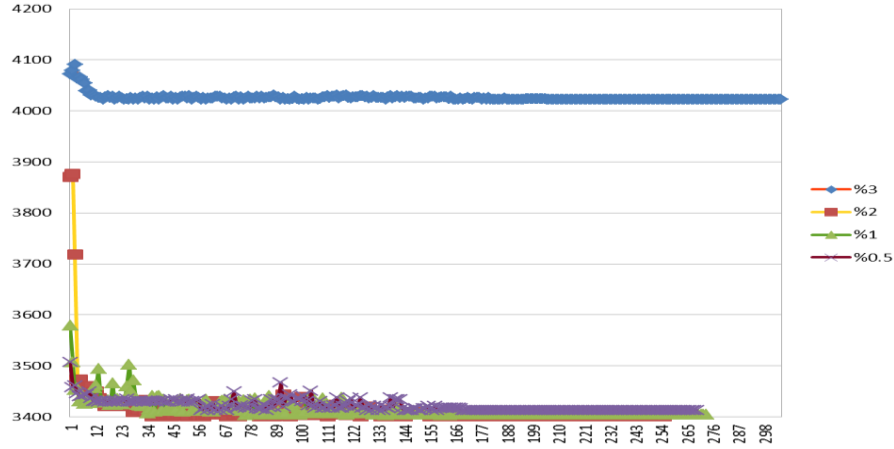


FIGURE 8. The convergence of SA-LS with different final temperature values

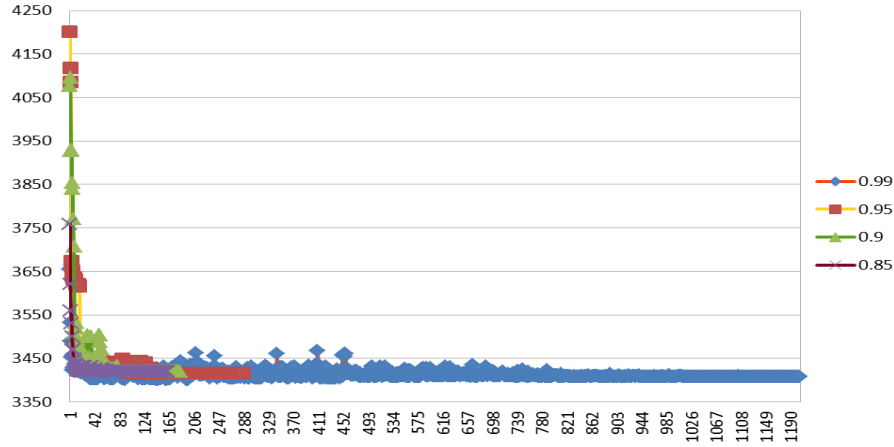


FIGURE 9. The convergence of SA-LS with different cooling rate values

productive. On the other hand, customers expect quick responses and want cheaper products.

As an extension of the well-known VRP, the HVRPSPD is considered in this paper. HVRPSPD is based on the optimization of several important problems such as, fleet management, vehicle assignment and routing, individually and simultaneously. The HVRPSPD presents a dashboard for decision makers where not only the transportation or distribution cost, but also the fleet investment cost is taken into account. We describe a procedure to solve the problems to be able to make right and effective decisions. To this point we give some theoretical explanations and

results for the problem. But in real life implementations these results correspond to some managerial insights for the logistics and distribution companies.

The first term of objective function in the MIP formulation is the total transportation or distribution cost and the second term is the fleet investment cost. The second term also reflects the occupation of the vehicles. The size of problem instance means the size of decision problem which the company would face.

The performance indicators used for evaluating the algorithms can also be interpreted from the distribution firm's point of view. Such as, a company can recognize its current state with Gap% or Dev% by examining the current performance with the best one. In business and operations management this topic is considered as gap analysis which is a four step process and helps the organizations to make use of their resources, capital and technology to reach their full potential. With the use of Gap% or Dev% the logistics company can determine where it is today and the management team can create an action plan to move the firm forward and fill in the performance gaps. Another aforementioned percentage indicator is the Imp%. This indicator can be thought as the percentage of improvement in total transportation/distribution and fleet investment costs. On the other hand it shows the decrease in total cost as well as total occupation of the vehicles in fleet.

In decision theory, there is a phenomenon called speed-accuracy trade-off. This fact explains the negative correlation between the quality and speed of any decision made. The constructive solutions methods are quick but may be less effective in solution quality. Besides the SA-LS method (or any exact solution methodology, *e.g solution of MIP model*) is slower but can be more effective in finding good quality solutions, relative to the constructive algorithms. But at this point the judgements of decision makers step in. Such that, in strategic level decisions where the consequences of any decision may affect the company in long-term and it is hard to return back, the quality of the solution is more important than the speed. Furthermore in operational level decisions, to be able to act quickly may be vital than to give best decision; since the daily problems need instantaneous solutions or in some cases there might be some opportunities which could not be missed. From the distribution company's perspective the quality of the decision can be measured by the Gap% or Imp% indicators and the speed of the decision can be measured by the CPU time of the solution.

Eventually, based on experimental studies we conclude that operational decisions can be significantly effective in a way that reduces transportation and fleet investment costs and improves the customer satisfaction.

6. Conclusions. In this study, we considered the HVRPSPD. We proposed a polynomial-size node-based MIP formulation for the HVRPSPD. We explained how the MIP formulation could be adapted to solve several variants of the HVRPSPD. Because the HVRPSPD is in the NP-hard problem class, we proposed a hybrid heuristic approach based on SA and LS algorithms called SA-LS to solve medium and large-size problems. We generated the initial solutions with the giant tour approach for SA-LS. We used seven inter-route and four intra-route moving strategies to generate neighbor solutions in SA-LS. The parameters of SA-LS were determined and enhanced based on our preliminary studies.

A series of experiments was conducted to evaluate the performance of the MIP formulation and the proposed hybrid heuristic approach using test instances derived from the literature. The computational results over 620 test instances show that

the proposed hybrid heuristic algorithm is computationally efficient for solving the HVRPSPD, and good quality solutions (8.19% on average) can be obtained in a reasonable computation time (approximately 6 s). Moreover, the proposed algorithm can determine solutions better than simple constructive heuristics, the NN and CWS algorithms, up to 50%. SA-LS is also efficient and preferable for solving HVRP and VRPSPD instances, which is a special case of HVRPSPD.

There are several interesting options for further research. The overall procedure could be adapted to solve the problems which are indicated as the special cases of the HVRPSPD. User friendly computer programs, which include the solution approaches proposed in this paper, may be developed with in a decision support framework that can offer service directly to the logistics firms. More sophisticated approaches or exact algorithms such as branch-and-cut, branch-and-price, column generation, which use the proposed formulations and valid inequalities, can be developed to obtain optimal solutions for medium and large-size instances of the HVRPSPD. Another powerful approach called matheuristics, which hybridizes the complex mathematical models with clever heuristic and metaheuristic algorithms, can be developed to solve the HVRPSPD. In real-life situations, there may be conflicting objectives in the problems considered. Thus, for multi-objective HVRPSPD, algorithms that can determine Pareto optimal solution sets can be developed. Finally, it is possible to extend the problem to consider uncertainty in demand and investigate the effects of very small capacity violations with non-linear penalty functions, as well as the effect of test data to the overall solution quality.

Acknowledgments. We would like to show our gratitude to the Edanz Editing for assistance with scientific and language review that greatly improved the manuscript.

Appendix A. Constructive Heuristics.

NN Algorithm. The steps of the NN algorithm adapted for the HVRPSPD are shown in Figure 10. Note that on the route obtained by the NN algorithm, despite the total delivery and pickup demands at the start and end nodes not exceeding the vehicle capacity, the load of the intermediate nodes may exceed the capacity because the vehicle load depends on the order of nodes visited. Thus, the route with an infeasible vehicle load can be repaired by considering it as a VRPSPD with one vehicle. As is known, the VRPSPD with one vehicle is a special case of the HVRPSPD, and this problem is easily solved by the MIP formulation of the HVRPSPD considering the following settings: $N = N', B = \{k^*\}, m = 1$, where N' is the set of nodes, which includes the depot and the nodes on the route to be repaired, and k^* is the vehicle type assigned to the route.

CWS Algorithm. In this study, we adapted the parallel CWS (PCWS) algorithm for the HVRPSPD. The steps of the algorithm are shown in Figure 11. The PCWS algorithm starts by listing all pairs (i, j) and corresponding savings in the network G in the first step. In the second step, the savings list is sorted from high to low. At the beginning of the algorithm, none of the nodes belongs to a route. Starting from the top of the sorted savings list, each pair (i, j) is checked. If none of the nodes in the pair (i, j) has been included in any route, then a new route to visit these nodes is created. An available vehicle type with minimum cost is assigned to this new route. If only one of the nodes in the pair (i, j) is on the first or last order of a route (let the route be r^*), then the node that has not yet been included in any route is added to route r^* if there is sufficient capacity. If both nodes in the pair

Algorithm: NN algorithm for the HVRPSPD. Input: HVRPSPD data. Output: Feasible HVRPSPD solution (S).
Step 1. $i := 1, S := \{\}$; Step 2. $j :=$ Depot node, $Route_i := \{\}$, $\Sigma Load_i := 0$; Assign the vehicle with minimum cost (k^*) to the i^{th} route; Step 3. Determine the nearest node (j^*) to node j on the i^{th} route and Goto Step 4; If no such node (j^*) is determined Goto Step 5; Step 4. If $\Sigma Load_i + d_{j^*} \leq Q_{k^*}$ and $\Sigma Load_i + p_{j^*} \leq Q_{k^*}$ then, $Route_i := Route_i \cup \{j^*\}$; $\Sigma Load_i := \max\{\Sigma Load_i + d_{j^*}, \Sigma Load_i + p_{j^*}\}$; $j := j^*$; Goto Step 3; Else, $S := S \cup Route_i$; $i := i + 1$; Goto Step 2; End If. Step 5. Stop and report the solution S;

FIGURE 10. Pseudo code of the NN

Algorithm: PCWS algorithm for the HVRPSPD. Input: HVRPSPD data. Output: Feasible HVRPSPD solution (S).
Step 1. Build the savings list; Step 2. Sort the savings list from high to low; $r := 0, Route_r := \{\}$; Step 3. Repeat for each (i, j) in the savings list from top to bottom; If neither i nor j does not belong to any route then, $r := r + 1$; Assign one of the available vehicle types (k^*) with minimum cost, where $d_i + d_j \leq Q_{k^*}$ and $p_i + p_j \leq Q_{k^*}$; $Route_r := Route_r \cup \{i, j\}$; End If. If either i or j is on the first or last order of the route (let the route be r^*) then, If the addition of node ($\#$), which does not belong to r^*, does not exceed the vehicle capacity then, $Route_{r^*} := Route_{r^*} \cup \{\#\}$; End If. End If. If both i and j are on the first or last order of the different routes then, All nodes are transferred to the available route with the minimum cost vehicle type; End If. Step 5. Stop and report solution S;

FIGURE 11. Pseudo code of the PCWS

(i, j) are on the first or last order of different routes, then all nodes are transferred to the available route with the minimum cost vehicle type. The algorithm stops when it reaches the end of the sorted savings list and a solution for the HVRPSPD is obtained. As mentioned in the NN algorithm, the load of intermediate nodes may exceed the capacity because the vehicle load depends on the order of nodes visited. Thus, the route with an infeasible vehicle load can be repaired by considering it as a VRPSPD with one vehicle.

REFERENCES

- [1] T. J. Ai and V. Kachitvichyanukul, [A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery](#), *Computers & Operations Research*, **36** (2009), 1693–1702.
- [2] S. Allahyari, M. Salari and D. Vigo, [A hybrid metaheuristic algorithm for the multi-depot covering tour vehicle routing problem](#), *European Journal of Operational Research*, **242** (2015), 756–768.
- [3] J.-F. Arvis, D. Saslavsky, L. Ojala, B. Shepherd, C. Busch, A. Raj and T. Naula, [Connecting to Compete 2016: Trade Logistics in the Global Economy-The Logistics Performance Index and Its Indicators](#), World Bank, 2016.
- [4] M. Avci and S. Topaloglu, [An adaptive local search algorithm for vehicle routing problem with simultaneous and mixed pickups and deliveries](#), *Computers & Industrial Engineering*, **83** (2015), 15–29.
- [5] M. Avci and S. Topaloglu, [A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery](#), *Expert Systems with Applications*, **53** (2016), 160–171.
- [6] R. Baldacci, M. Battarra and D. Vigo, [Routing a heterogeneous fleet of vehicles](#), *The Vehicle Routing Problem: Latest Advances and New Challenges, Oper. Res./Comput. Sci. Interfaces Ser., Springer, New York*, **43** 2008, 3–27.
- [7] R. Baldacci, P. Toth and D. Vigo, [Exact algorithms for routing problems under vehicle capacity constraints](#), *Annals of Operations Research*, **175** (2010), 213–245.
- [8] J. E. Beasley, [Route first-Cluster second methods for vehicle routing](#), *Omega*, **11** (1983), 403–408.
- [9] N. Bianchessi and G. Righini, Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery, *Computers & Operations Research*, **34** (2007), 578–594.
- [10] J. Brandão, A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem, *European journal of operational research*, **195** (2009), 716–728.
- [11] J. Brandão, [A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem](#), *Comput. Oper. Res.*, **38** (2011), 140–151.
- [12] G. Calleja, A. Corominas, A. García-Villoria and R. Pastor, [Hybrid metaheuristics for the Accessibility Windows Assembly Line Balancing Problem Level 2 \(AWALBP-L2\)](#), *European Journal of Operational Research*, **250** (2016), 760–772.
- [13] S. Çetin and C. Gencer, Heterojen araç filolu zaman pencereli eş zamanlı dağıtım-toplamalı araç rotalama problemleri: Matematiksel model, *Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi*, **3** (2011), 19–27.
- [14] E. Choi and D.-W. Tcha, [A column generation approach to the heterogeneous fleet vehicle routing problem](#), *Computers & Operations Research*, **34** (2007), 2080–2095.
- [15] A. K. Coşar and B. Demir, Domestic road infrastructure and international trade: Evidence from turkey, *Journal of Development Economics*, **118** (2016), 232–244.
- [16] G. B. Dantzig and J. H. Ramser, [The truck dispatching problem](#), *Management Science*, **6** (1959/60), 80–91.
- [17] M. Dell'Amico, G. Righini and M. Salani, A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection, *Transportation science*, **40** (2006), 235–247.
- [18] J. Dethloff, [Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up](#), *OR-Spektrum*, **23** (2001), 79–96.
- [19] Y. Gajpal and P. Abad, [Saving-based algorithms for vehicle routing problem with simultaneous pickup and delivery](#), *Journal of the Operational Research Society*, **61** (2010), 1498–1509.

- [20] M. Gendreau, G. Laporte, C. Musaraganyi and É. D. Taillard, [A tabu search heuristic for the heterogeneous fleet vehicle routing problem](#), *Computers & Operations Research*, **26** (1999), 1153–1173.
- [21] F. Gheysens, B. Golden and A. Assad, [A new heuristic for determining fleet size and composition](#), *Netflow at Pisa, Math. Programming Stud.*, 1986, 233–236.
- [22] F. P. Goksal, I. Karaoglan and F. Altıparmak, [A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery](#), *Computers & Industrial Engineering*, **65** (2013), 39–53.
- [23] B. Golden, A. Assad, L. Levy and F. Gheysens, [The fleet size and mix vehicle routing problem](#), *Computers & Operations Research*, **11** (1984), 49–66.
- [24] A. Hoff, H. Andersson, M. Christiansen, G. Hasle and A. Løkketangen, [Industrial aspects and literature survey: Fleet composition and routing](#), *Comput. Oper. Res.*, **37** (2010), 2041–2061.
- [25] S. Irnich and G. Desaulniers, [Shortest path problems with resource constraints](#), *Column Generation*, Springer, (2005), 33–65.
- [26] S. Irnich, M. Schneider and D. Vigo, Chapter 9: Four variants of the vehicle routing problem, *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, SIAM, (2014), 241–271.
- [27] A. A. Juan, J. Faulin, J. Caceres-Cruz, B. B. Barrios and E. Martinez, [A successive approximations method for the heterogeneous vehicle routing problem: Analysing different fleet configurations](#), *Eur. J. Ind. Eng.*, **8** (2014), 762–788.
- [28] I. Kara and T. Derya, [Polynomial size formulations for the distance and capacity constrained vehicle routing problem](#), *AIP Conference Proceedings*, **1389** (2011), 1713–1718.
- [29] I. Karaoglan, *Location Routing Problem with Simultaneous Pickup and Delivery in Distribution Network Design*, PhD thesis, Gazi University, Institute of Science, Ankara, Turkey, 2009.
- [30] B. Keçeci, F. Altıparmak and I. Kara, The heterogeneous vehicle routing problem with simultaneous pickup and delivery: A hybrid heuristic approach based on simulated annealing, *CIE44 & IMSS'14 Proceedings*, (2014).
- [31] B. Keçeci, F. Altıparmak and I. Kara, Heterogeneous vehicle routing problem with simultaneous pickup and delivery: Mathematical formulations and a heuristic algorithm, *Journal of the Faculty of Engineering and Architecture of Gazi University*, **30** (2015), 185–195.
- [32] B. Keçeci, F. Altıparmak and I. Kara, [A hybrid constructive mat-heuristic algorithm for the heterogeneous vehicle routing problem with simultaneous pick-up and delivery](#), *Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Comput. Sci.*, Springer, Cham, 9595 (2016), 1–17.
- [33] S. Kirkpatrick, C. D. Gelatt, Jr. and M. P. Vecchi, [Optimization by simulated annealing](#), *Science*, **220** (1983), 671–680.
- [34] Ç. Koç, T. Bektaş, O. Jabali and G. Laporte, [Thirty years of heterogeneous vehicle routing](#), *European Journal of Operational Research*, **249** (2016), 1–21.
- [35] T. W. Liao, P.-C. Chang, R. J. Kuo and C.-J. Liao, [A comparison of five hybrid metaheuristic algorithms for unrelated parallel-machine scheduling and inbound trucks sequencing in multi-door cross docking systems](#), *Applied Soft Computing*, **21** (2014), 180–193.
- [36] F.-H. Liu and S.-Y. Shen, The fleet size and mix vehicle routing problem with time windows, *Journal of the Operational Research society*, **50** (1999), 721–732.
- [37] S. G. Liu, W. L. Huang and H. M. Ma, [An effective genetic algorithm for the fleet size and mix vehicle routing problems](#), *Transportation Research Part E: Logistics and Transportation Review*, **45** (2009), 434–445.
- [38] M. A. Masmoudi, M. Hosny, K. Braekers and A. Dammak, [Three effective metaheuristics to solve the multi-depot multi-trip heterogeneous dial-a-ride problem](#), *Transportation Research Part E: Logistics and Transportation Review*, **96** (2016), 60–80.
- [39] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, [Equation of state calculations by fast computing machines](#), *The Journal of Chemical Physics*, **21** (1953), 1087–1092.
- [40] H. Min, [The multiple vehicle routing problem with simultaneous delivery and pick-up points](#), *Transportation Research Part A: General*, **23** (1989), 377–386.
- [41] F. Alfredo Tang Montané and R. D. Galvão, [A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service](#), *Comput. Oper. Res.*, **33** (2006), 595–619.

- [42] G. Nagy and S. Salhi, [Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries](#), *European Journal of Operational Research*, **162** (2005), 126–141.
- [43] I. H. Osman and S. Salhi, Local search strategies for the vehicle fleet mix problem, *Modern Heuristic Search Methods*, 131–153.
- [44] D. C. Paraskevopoulos, P. P. Repoussis, C. D. Tarantilis, G. Ioannou and G. P. Prastacos, [A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows](#), *Journal of Heuristics*, **14** (2008), 425–455.
- [45] P. H. V. Penna, A. Subramanian and L. S. Ochi, [An iterated local search heuristic for the heterogeneous fleet vehicle routing problem](#), *Journal of Heuristics*, **19** (2013), 201–232.
- [46] O. Polat, C. B. Kalayci, O. Kulak and H.-O. Günther, [A perturbation based variable neighborhood search heuristic for solving the vehicle routing problem with simultaneous pickup and delivery with time limit](#), *European Journal of Operational Research*, **242** (2015), 369–382.
- [47] C. Prins, [A simple and effective evolutionary algorithm for the vehicle routing problem](#), *Comput. Oper. Res.*, **31** (2004), 1985–2002.
- [48] C. Prins, [Two memetic algorithms for heterogeneous fleet vehicle routing problems](#), *Engineering Applications of Artificial Intelligence*, **22** (2009), 916–928.
- [49] J. Renaud and F. F. Boctor, [A sweep-based algorithm for the fleet size and mix vehicle routing problem](#), *European Journal of Operational Research*, **140** (2002), 618–628.
- [50] S. Ropke and D. Pisinger, [An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows](#), *Transportation science*, **40** (2006), 455–472.
- [51] S. Salhi and G. Nagy, A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling, *Journal of the operational Research Society*, **50** (1999), 1034–1042.
- [52] S. Salhi and G. K. Rand, [Incorporating vehicle routing into the vehicle fleet composition problem](#), *European Journal of Operational Research*, **66** (1993), 313–330.
- [53] L. Simeonova, N. Wassan, S. Salhi and G. Nagy, [The heterogeneous fleet vehicle routing problem with light loads and overtime: Formulation and population variable neighbourhood search with adaptive memory](#), *Expert Systems with Applications*, **114** (2018), 183–195.
- [54] M. M. Solomon, [Algorithms for the vehicle routing and scheduling problems with time window constraints](#), *Operations Research*, **35** (1987), 254–265.
- [55] A. Subramanian, L. M. d. A. Drummond, C. Bentes, L. S. Ochi and R. Farias, A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery, *Computers & Operations Research*, **37** (2010), 1899–1911.
- [56] A. Subramanian, P. H. V. Penna, E. Uchoa and L. S. Ochi, [A hybrid algorithm for the heterogeneous fleet vehicle routing problem](#), *European Journal of Operational Research*, **221** (2012), 285–295.
- [57] A. Subramanian, E. Uchoa, A. A. Pessoa and L. S. Ochi, [Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery](#), *Optimization Letters*, **7** (2013), 1569–1581.
- [58] É. D. Taillard, [A heuristic column generation method for the heterogeneous fleet VRP](#), *RAIRO-Operations Research*, **33** (1999), 1–14.
- [59] E.-G. Talbi, A taxonomy of hybrid metaheuristics, *Journal of heuristics*, **8** (2002), 541–564.
- [60] C. Blum, J. Puchinger, G. Raidl and A. Roli, [Hybrid metaheuristics](#), *Hybrid Optimization, Springer Optim. Appl.*, Springer, New York, **45** (2011), 305–335.
- [61] C. D. Tarantilis, C. T. Kiranoudis and V. S. Vassiliadis, [A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem](#), *European Journal of Operational Research*, **152** (2004), 148–158.
- [62] A. S. Tasan and M. Gen, [A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries](#), *Computers & Industrial Engineering*, **62** (2012), 755–761.
- [63] P. Toth and D. Vigo, [The Vvehicle Routing Problem](#), SIAM Monographs on Discrete Mathematics and Applications, 9. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002.
- [64] X. F. Wang, The heterogeneous fleet location routing problem with simultaneous pickup and delivery and overloads, *Discrete Contin. Dyn. Syst. Ser. S*, **12** (2019), 1147–1166.
- [65] N. A. Wassan and I. H. Osman, [Tabu search variants for the mix fleet vehicle routing problem](#), *Journal of the Operational Research Society*, **53** (2002), 768–782.

- [66] C. Waters, [Expanding the scope of linear programming solutions for vehicle scheduling problems](#), *Omega*, **16** (1988), 577–583.
- [67] Y. Zhang, M. Y. Qi, W.-H. Lin and L. X. Miao, [A metaheuristic approach to the reliable location routing problem under disruptions](#), *Transportation Research Part E: Logistics and Transportation Review*, **83** (2015), 90–110.

Received November 2018; revised July 2019.

E-mail address: bkececi@baskent.edu.tr

E-mail address: fulyaal@gazi.edu.tr

E-mail address: ikara@baskent.edu.tr