

Research Article

Optimal Routing for Heterogeneous Fixed Fleets of Multicompartment Vehicles

Qian Wang,¹ Qingkai Ji,^{1,2} and Chun-Hung Chiu¹

¹ Sun Yat-sen Business School, Sun Yat-sen University, Guangzhou 510275, China

² School of Management, Dalian University of Technology, Dalian 116024, China

Correspondence should be addressed to Qingkai Ji; jorter@foxmail.com

Received 26 February 2014; Accepted 31 March 2014; Published 30 April 2014

Academic Editor: Tsan-Ming Choi

Copyright © 2014 Qian Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a metaheuristic called the reactive guided tabu search (RGTS) to solve the heterogeneous fleet multicompartment vehicle routing problem (MCVRP), where a single vehicle is required for cotransporting multiple customer orders. MCVRP is commonly found in delivery of fashion apparel, petroleum distribution, food distribution, and waste collection. In searching the optimum solution of MCVRP, we need to handle a large amount of local optima in the solution spaces. To overcome this problem, we design three guiding mechanisms in which the search history is used to guide the search. The three mechanisms are experimentally demonstrated to be more efficient than the ones which only apply the known distance information. Armed with the guiding mechanisms and the well-known reactive mechanism, the RGTS can produce remarkable solutions in a reasonable computation time.

1. Introduction

In areas such as delivery of fashion apparel, petroleum distribution, chemical transportation, food delivery, and waste recycling, due to the characteristics of the products, vehicles with multiple separated compartments are indispensable for delivering to (or picking up from) clients who require delivery (or pick-up) of more than one (incompatible) type of product. Routing the vehicles in such multicompartments environments has gained some attention in the vehicle routing problem (VRP) circle [1–3] and it is called the “multicompartment VRP” (MCVRP).

A fixed homogeneous fleet is often assumed in the classic VRP, and also in the MCVRP. However, many researchers (e.g., [4–9]) claim that the fleet should be assumed heterogeneous because of the following reasons. Firstly, even though the fleet is very likely homogeneous in the inception of a distribution company, after a long time in business, the company tends to own heterogeneous vehicles as a result of vehicle market changes (they would buy different vehicles at different costs) and the different depreciation rates of

vehicles [10]. Secondly, it may be wiser for the company to possess/operate a heterogeneous fleet for business flexibility since it can meet different needs of clients.

Heterogeneous fleets are particularly common in multicommodity multicompartment transportation because the physical structure of multicompartment vehicles is more complex than ordinary ones. For example, in the transportation of apparel products, products with different styles and packaging are usually delivered at the same time in one vehicle. Based on the product characteristics, some products are hanged on flexible swing rods and some are packed in boxes. Therefore, the vehicle is reorganized to form multiple separated “compartments” for each type of product. Moreover, products such as gas, chemical, and food require not only separation in multiple compartments, but also special treatments (e.g., temperature, pressure requirements, etc.) during transportation, which imply potential technical differences between vehicles. Therefore, it is natural to consider multicompartment transportation under the setting of heterogeneous fleet.

We formalize the heterogeneous fixed fleet multi-compartment vehicle routing problem (HFFMCVRP) in this article and propose a tabu search-based algorithm called the Reactive Guided Tabu Search (RGTS), which allows a guiding mechanism to collaborate with the well-known reactive mechanism [11], to tackle this problem. The guiding mechanisms employ constantly-updated search history of the appearance of arcs in solutions and the objective values of those solutions in the search to overcome the problem of a large amount of local optima and an induced high diversification. We also change the ways of using the information and compare the resulting different guiding mechanisms to see the delicacy. Experiments on generated HFFMCVRP instances show that (1) the collaboration of guiding mechanism and reactive mechanism significantly improves the efficiency of tabu search, (2) the guiding mechanisms utilizing search history in different ways have similar efficiency, but they all overwhelm the guiding mechanisms without utilizing history information.

In the rest of this paper, we review the literature on two closely related topics, the MCVRP and the HFFVRP, in Section 2. A formal description of the HFFMCVRP is given in Section 3. The outline and details of our proposed algorithm are provided in Section 4. In Section 5, we report computational results and justify the algorithm. Section 6 gives conclusions.

2. Related Literature

Much of the literature regarding the MCVRP focuses on dispatching different petroleum products using tank trucks with isolated compartments (e.g., [12–15]). For environmental issues, multicompartment vehicles are also used in the collection of source-separated waste streams (e.g., [16–18]). Muyldermans and Pang [2] apply a guided local search algorithm to evaluate the cost savings of cocollection over single collection in waste collection. El-Fallahi et al. [1] point out another application involving the distribution of cattle food to farms, and they first use the term “multicompartment VRP” to classify such problems. Believing that “the problem is harder than the VRP and only very small instances can be solved by commercial MIP solvers,” they develop a memetic algorithm (MA) and a tabu search (TS) algorithm to tackle the problem. The MA and TS are evaluated in instances that were generated from the capacitated VRP benchmark instances. We note that although very few MCVRP literature directly focuses on the transportation of chemical and fashion products, similar to other VRP, the MCVRP is a common problem for these and many other areas (see, e.g., [19–21]). Back to the MCVRP, Derigs et al. [3] comprehensively introduce a solver suite of heuristic components covering a broad range of alternative methods for construction, local search, large neighborhood search, and metaheuristics to solve the MCVRP. Comparisons between these components are made in order to determine the best composition of the solver suite. However, in the MCVRP instances generated by bisecting classic VRP instances, these heuristics all failed to produce MCVRP solutions of negligible gaps

with respect to the corresponding best-known VRP solutions (the instance-generating process guarantees that a feasible VRP solution is also feasible to the MCVRP). This suggests that the MCVRP is difficult and there is still room for improvement. Besides, these papers are creditable in that they are devoted to the problem commonly found in multi-commodity distribution, but there is one thing missing, that is, the heterogeneous fleet. Taillard [4] first addresses the heterogeneous fleet routing issue; noticing that the feasibility check and the cost evaluation of a move require more additional effort in the HFFVRP than in the homogeneous VPRs, he uses a column generation method combined with adaptive memory programming (AMP) embedded in a TS, called the heuristic column generation (HCG), to solve the problem. Though coped with powerful solver CPLEX, the HCG fails to produce most of the BKS (best-known solution) unveiled later by other HFFVRP metaheuristics mentioned below.

Tarantilis et al. [22] propose a list-based threshold accepting algorithm (LBTA), and Tarantilis et al. [23] develop a backtracking adaptive threshold accepting algorithm (BATA). Both LBTA and BATA belong to the class of threshold accepting (TA) algorithms [24]. TA explores the solution search space by allowing uphill moves using a threshold value in order to escape premature local optima. In BATA, the threshold value is not only lowered, but it is also raised—or backtracked—depending on failure to find a new solution using the former value; while in LBTA, a list of values is continually updated and the maximum value is used to update the threshold during the search. Tarantilis et al. [25] establish a guided TS (GTS) in which “bone” (chain of nodes) operations rather than node or edge operations are considered. The neighborhood search is then guided by a mechanism that continually modifies the objective function through penalty along undesired edges chosen during iterations. Li et al. [5] propose a record-to-record travel (RTR) algorithm, denoted as the HRTR, for the HFFVRP. The HRTR alternately applies downhill moves and record-to-record travel moves in the inner loop, while in the outer loop the current solution is perturbed by a procedure of removing and reinserting nodes in order to break through the local traps. Li et al. [26] develop a heuristic in which a multistart adaptive memory programming (MAMP) at each iteration constructs multiple provisional solutions from the elite pool. This pool is continuously upgraded by a modified TS. Path relinking is also integrated as an intensification strategy to enhance the performance of the MAMP. Brandão [27] designs an elaborated TS that includes several techniques such as shaking/perturbation, adaptive parameters updating, postoptimization, and strategic oscillation in order to maximize the robustness.

Being the MCVRP or the HFFVRP, or other rich VRP, the algorithms for them are designed to be more and more complicated and specific to cope with the increasing complexities of the rich VRP. Intrinsically, the HFFMCVRP is more complex than the MCVRP and the HFFVRP since it carries the characteristics of both of them, which leaves one little choice but metaheuristics when trying to tackle the HFFMCVRP. Therefore, we propose the RGTS for this task.

3. Description of the HFFMCVRP

Before we present the algorithm, we describe the HFFM-CVRP first.

The HFFMCVRP is defined on the undirected network $G = (L, E)$. Vehicles of the same type share the same characteristics. Customers may place several orders requesting different products, and the orders can be served separately; that is, the customers can be visited more than once. The objective of the HFFMCVRP is to seek the routes with the minimal total transportation cost to serve all the customer orders by assigning the orders to compartments in different vehicles such that all capacities and incompatibilities constraints are met. The HFFMCVRP can be formulated as an integer program as follow:

$$\min \sum_{v \in V} \sum_{i \in L} \sum_{j \in L} \text{cost}^v d_{ij} b_{ijv} \quad (1)$$

s.t.

$$\sum_{j \in L_c} b_{0jv} \leq 1, \quad v \in V, \quad (2)$$

$$\sum_{i \in L} b_{ilv} = \sum_{j \in L} b_{ijv}, \quad v \in V, l \in L, \quad (3)$$

$$u_{0v} = 1, \quad v \in V, \quad (4)$$

$$(n+1) \sum_{i \in L} b_{ijv} \geq u_{jv}, \quad v \in V, j \in L_c, \quad (5)$$

$$b_{ijv} (u_{iv} - u_{jv} + 1) = 0, \quad v \in V, i \in L, j \in L_c, \quad (6)$$

$$\sum_{v \in V} \sum_{j \in L_c} b_{0jv} f_m(v) \leq |V_m|, \quad m \in M, \quad (7)$$

$$\sum_{o \in O} q_o x_{ovc} \leq Q_c, \quad v \in V, c \in C_v, \quad (8)$$

$$\sum_{v \in V} \sum_{c \in C_v} x_{ovc} \leq 1, \quad o \in O, \quad (9)$$

$$\sum_{o \in S_j^o} \sum_{c \in C_v} x_{ovc} \leq |O| \sum_{i \in L} b_{ijv}, \quad v \in V, j \in L_c, \quad (10)$$

$$\sum_{o \in S_p^o} x_{ovc} \leq |O| y_{pvc}, \quad p \in P, v \in V, c \in C_v, \quad (11)$$

$$y_{pvc} = 0, \quad (p, c) \in I_2, v \in V, c \in C_v, \quad (12)$$

$$y_{pvc} + y_{qvc} \leq 1, \quad (p, q) \in I_1, v \in V, c \in C_v, \quad (13)$$

$$b_{ijv} \in \{0, 1\}, \quad i \in L, j \in L, v \in V, \quad (14)$$

$$u_{iv} \in \{0, 1, \dots, n+1\}, \quad i \in L, v \in V, \quad (15)$$

$$x_{ovc} \in \{0, 1\}, \quad o \in O, v \in V, c \in C_v, \quad (16)$$

$$y_{pvc} \in \{0, 1\}, \quad p \in P, v \in V, c \in C_v. \quad (17)$$

Objective (1) aims at minimizing the total travel cost. Constraint (2) ensures that all vehicles depart at most once from depot 0. Constraint (3) ensures that if a vehicle arrives at a location, then it must leave from this location afterward. Constraints (2)-(3) make sure that a vehicle tour must always start and end at depot 0. Constraint (4) enforces the location of depot 0 at position 1. Constraint (5) makes sure that if node j is never visited by vehicle v , its position will be $u_{jv} = 0$. Constraint (6) imposes the condition that the position of node j is just higher than the position of node i , if vehicle v travels from i to j . Constraints (4)-(6) complete the subtour elimination. Constraint (7) sets a limit on the number of vehicles available, where $f_m(v) = 1$ if vehicle v is of type m , otherwise 0. Constraint (8) states that the goods loaded into compartment c on vehicle v must not exceed the compartment capacity Q_c . Constraint (9) ensures that each order is assigned to exactly one compartment of a vehicle. Constraint (10) imposes the restriction that a vehicle must visit customer j if any orders from j are assigned to the vehicle. Constraint (11) makes sure that the total order number of a product in a compartment cannot exceed the total order number of a product. Constraints (12)-(13) describe the incompatibilities and (14)-(17) indicate the decision variables.

4. The Proposed RGTS Algorithm

In our RGTS algorithm, we first obtain an initial feasible solution through a simple procedure. This solution is subsequently improved by a number of well-known operators in the TS framework. In order to seek better quality solutions, we applied a reactive mechanism and a guided mechanism to strengthen the TS.

4.1. Constructing the Initial Solution. First, a void route is generated for each vehicle in the fleet (e.g., for a fleet consisting of three type-A vehicles and two type-B vehicles, five dedicated routes denoted as ‘‘A, A, A, B, B’’ will be generated). Then, following Procedure 1, a *feasible* initial solution can be generated (not necessarily, but for all our tests, it happened to be so) (see Figure 1).

Procedure 1. Construction of the initial solution is as follows.

- (1) Sort all the orders in ascending order according to the demand.
- (2) While there are unserved orders, find the closest one \bar{o} to the depot. According to the relationship I_2 , find $\bar{c} = \arg \min \{Q_c - q_{\bar{o}} \mid Q_c \geq q_{\bar{o}}\}$, and assign \bar{o} to \bar{c} (i.e., add \bar{o} to the end of route $l_{\bar{c}}$ of \bar{c}).
- (3) One by one, insert the sorted orders into route $l_{\bar{c}}$ using least-cost insertion while maintaining route feasibility.
- (4) Update Q_c and repeat steps (2)-(3) until all orders are served.

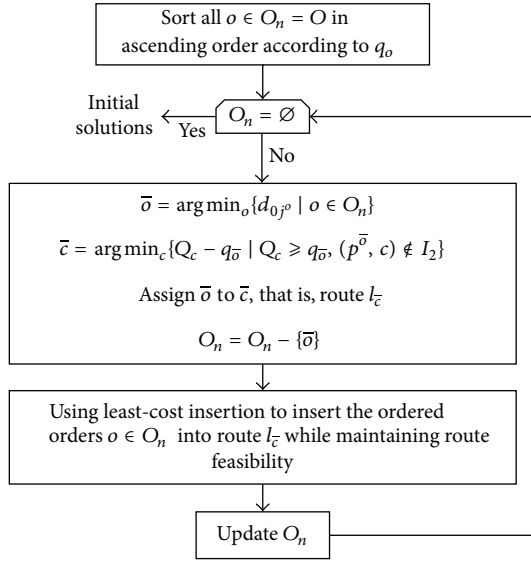


FIGURE 1: Flow chart of Procedure 1.

4.2. Operators and Neighborhood Reduction Strategy. Operators define the local search moves. We implement several well-known operators. Specifically, for an interroute operation, we use 2-opt^* , 1-0 move (*interrelocate*), 1-1 move (*interswap*), and 2-1 move (*three-point move*), while for an innerroute operation, we use *swap*, *relocate*, and 2-opt . Note that the nodes in a route now mean order nodes, not customer nodes.

Due to the high complexity of the VRPs, even metaheuristics may require a significant amount of computation when solving medium/large-scale problems. Accordingly, there is nearly a consensus (e.g., [5, 28]) that one should implement the so-called neighborhood reduction strategy to focus on desirable neighborhoods so as to reduce computation. In our mechanism, a nontabu move is allowed only if it connects edge (o_i, o_j) such that the owner of o_i , customer i , is a neighbor of the owner of o_j , customer j ; that is,

$$d_{ij} \leq \delta \cdot \text{avg}_i, \quad d_{ij} \leq \delta \cdot \text{avg}_j, \quad (18)$$

where δ is a small number used to define the limit of neighborhoods and avg_i is the average distance of all edges (of customer nodes) starting from i .

4.3. Guiding Mechanism. A guiding mechanism is a systematic method that continuously identifies low-quality features and tries to reject them by penalizing them in the objective function so as to guide the search into unexplored domains of the solution space. In routing problems or shortest path problems, it is common to define long edges as low-quality features. A long edge is more likely to be selected according to a utility function and then to be penalized by adding an extra value to its original distance; as a result, this edge can be highly possibly avoided in future search. Voudouris and Tsang [29] first introduce a guiding mechanism into the local search when solving the traveling salesman problem. Every time the local search falls in a local optima S^* , the edge

$\zeta = (i^*, j^*)$ in S^* corresponding to the maximum value of utility function $U(i, j) = d_{ij}/(1 + p_{ij})$ is penalized as $d'_{\zeta} = d_{\zeta} + p_{\zeta} \cdot \lambda$, where p_{ζ} indicates the times that edge ζ has been penalized and λ is an experimentally valued parameter that defines the strength of penalties. Tarantilis et al. [25] introduce another guiding mechanism into the tabu search (the guided tabu search, GTS) and successfully applied it to the HFFVRP. A parameter *guidFreq* is used to switch on/off the guiding mechanism; that is, during every *guidFreq* iterations, an edge in the current solution is selected according to utility function $U(i, j) = d_{ij}/[(1 + p_{ij}) \cdot \text{AVG}_{ij}]$ and penalized as $d'_{\zeta} = d_{\zeta} + p_{\zeta} \cdot \lambda \cdot \text{AVG}_{\zeta}$, where AVG_{ij} is the average length of all edges beginning from locations i and j in the edge set E and d'_{ζ} is only used for $2 \times \text{guidFreq}$ iterations, and the original one is then substituted back. This allows some self-correction in the algorithm because a “bad” edge can have a second chance to become “good,” since “bad” and “good” are defined by the somewhat naïve utility function. Tarantilis et al. [25] claim that their utility function considers not only the distance d_{ij} , but also the relative positions of i and j according to the rest of the customer population (i.e., AVG_{ij}), which leads to a more balanced edge selection.

Edge selection determined by utility function plays a key role in the guiding mechanism because a poor selection would turn guiding into misleading. In the above two guiding mechanisms, such misleading may be avoided by decreasing the utility function values of edges that have been penalized many times before. That is, the edges of higher values of p_{ij} have smaller values of $U(i, j)$, so the selection will not be restricted on a small set of edges. However, except for p_{ij} , other factors in those utility functions (viz., AVG_{ij} and d_{ij}) are predetermined characteristics of the problems, thus containing no information about the evolution of the search that might serve as useful memories. To utilize such renewable information about the evolution of the search, we introduce a matrix $\text{Appr}_{(n'+1) \times (n'+1)}$ to collect information about the appearance of edges in sound solutions. At the beginning, every Appr_{ij} is set to one and is updated each time an operator returns a local optima s^* ; that is, for every edge in s^* , if $f(s_b) < f(s^*)$, renew $\text{Appr}_{ij} = \text{Appr}_{ij} + f(s_b)/f(s^*)$, otherwise $\text{Appr}_{ij} = \min\{\text{Appr}_{ij} - f(s_b)/f(s^*), 1\}$, where s_b is the incumbent best-found solution. By applying $\text{Appr}_{(n'+1) \times (n'+1)}$, we are able to propose a more “just” utility function which utilizes historic search information to evaluate the edges better. Note that in step (2) of Procedure 2, we reset $\text{Appr}_{(n'+1) \times (n'+1)}$ to its default state, a unit matrix, if the search is considered to be in the chaos. Thus, it allows further self-correcting opportunity. It also serves as an important link between the guiding mechanism and the reactive mechanism.

Three utility functions are given in (19). They all contain the matrix *Appr* but differ in using other predetermined characteristics of the problems, d_{ij} and AVG_{ij} (note that distance between two orders is distance between their corresponding customers). Function U_1 can be considered as the most comprehensive function, while U_2 as the purest and U_3 as the modest. In all utility functions, the more often the edge

(i, j) appears in good solutions, the higher the value of $Appr_{ij}$ is, leading to a smaller value of $U(i, j)$, hence a less possibility of edge (i, j) being selected and penalized;

$$\begin{aligned} U_1(i, j) &= \frac{d_{ij}}{(1 + p_{ij}) \cdot AVG_{ij} \cdot Appr_{ij}}, \\ U_2(i, j) &= \frac{1}{(1 + p_{ij}) \cdot Appr_{ij}}, \\ U_3(i, j) &= \frac{d_{ij}}{(1 + p_{ij}) \cdot Appr_{ij}}. \end{aligned} \quad (19)$$

Note that we use the standard penalty term $d'_\zeta = d_\zeta + p_\zeta \cdot \lambda$ for all utility functions. Our proposed guiding mechanism is sketched out in Procedure 2 (see Figure 2).

Procedure 2. The guiding mechanism is as follows.

s^* is the local optima returned by an operator

(1) Update $Appr_{n \times n}$

(2) If $guid = guidFreq$

select edge $\zeta = (i^*, j^*) = \arg \max_{(i,j) \in s^*} U(i, j)$,

penalize edge by $d'_\zeta = d_\zeta + p_\zeta \cdot \lambda$ for $2 \times guidFreq$ iterations

$guid = 0$;

else

$guid = guid + 1$.

4.4. Reactive Mechanism. The reactive TS (RTS) is first proposed by Battiti and Tecchiolli [11]. The primary feature of their RTS is the reactive mechanism for adapting the tabu tenure, or tabu list size TL , to the evolution of the search in order to free the search trajectory from a limited part of the search space, instead of entirely avoiding closed search cycles or repetitions. The RTS has been successfully applied to the VRP, with a few modifications (e.g., [30–32]). They demonstrate the robustness of the RTS and the limiting effect of parameter changes. Therefore, we apply the standard routine of the RTS with an additional modification: turning on/off the neighborhood reduction strategy and resetting a matrix used in the guiding mechanism (step (2) of Procedure 3) to default state. The reactive mechanism is described in Procedure 3 and the parameters are described in parameters section (see Figure 3).

Procedure 3. The reactive mechanism is as follows.

s^* is the local optima returned by an operator.

(1) $TLLastCh = TLLastCh + 1$; $itr = itr + 1$.

If $f(s^*) \in LTL$,

$ReptP = itr$; $itr = 0$; $RCounter = RCounter + 1$;

else

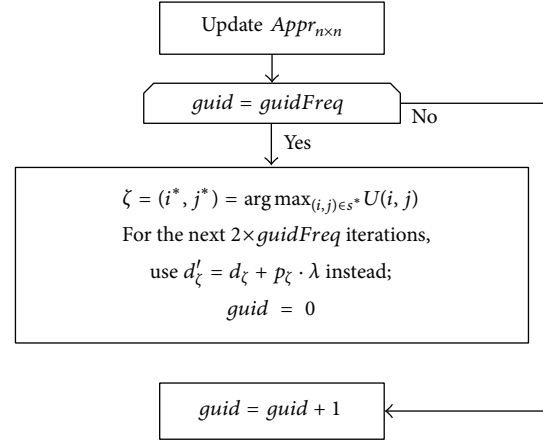


FIGURE 2: Flow chart of Procedure 2.

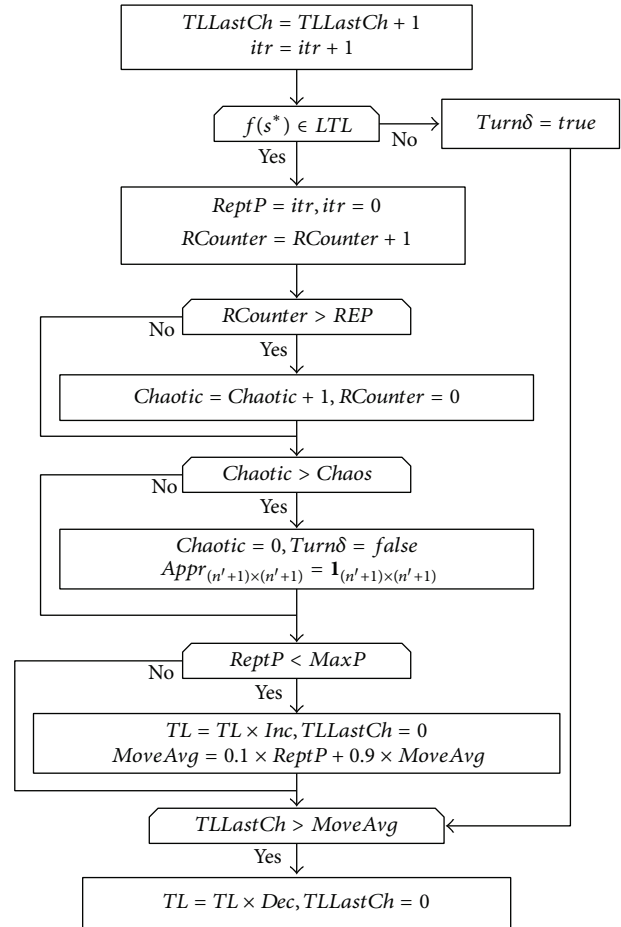


FIGURE 3: Flow chart of Procedure 3.

$Turn\delta = true$; go to (4).

(2) If $RCounter > REP$,

$Chaotic = Chaotic + 1$; $RCounter = 0$;

if $Chaotic > Chaos$,

$Chaotic = 0; Turn\delta = false; Appr_{(n'+1) \times (n'+1)} = \mathbf{1}_{(n'+1) \times (n'+1)}$
 (3) If $ReptP < MaxP$,
 $TL = TL \times Inc; TLLastCh = 0;$
 $MoveAvg = 0.1 \times ReptP + 0.9 \times MoveAvg.$
 (4) If $TLLastCh > MoveAvg$,
 $TL = TL \times Dec; TLLastCh = 0.$

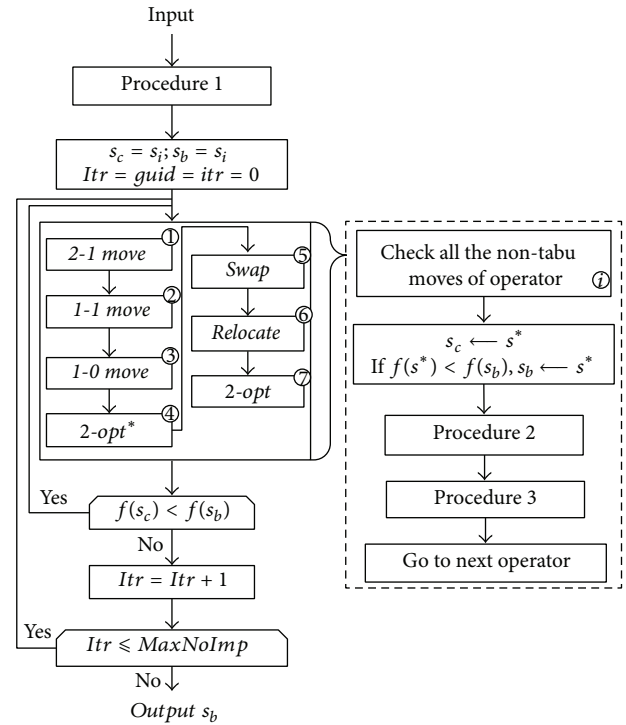
In Procedure 3, note that $f(s) \in LTL$ implies that we use another list LTL containing the objective values of former best-found solutions to help detect repetition; namely, a repetition is detected if $f(s^*)$ has been reached before, even if s^* is a new solution with different structure. In other words, the “repetition” is a harsh one here. In addition to the dynamic change of tabu tenure, the second feature of RTS is the “escape procedure,” which is activated when the search is considered to be drowning in chaos, that is, when $Chaotic > Chaos$. Unlike the escape procedure of Battiti and Tecchiolli [11], which consists of a series of random moves, we use $Turn\delta$ as a switch to turn off the neighborhood reduction strategy in order to free the search from a restricted area of solution space; the neighborhood reduction strategy is turned back on itself if the repetition is not detected in the next iteration. Moreover, the matrix used in the guiding mechanism is set to the default state; that is, $Appr_{(n'+1) \times (n'+1)} = \mathbf{1}_{(n'+1) \times (n'+1)}$, so as to provide more opportunities of self-correction mentioned in last subsection. Now we present the framework of our algorithm in Figure 4.

5. Computational Results

Our proposed algorithm was programmed in Visual Studio 2008 C# and executed on a laptop with Intel (R) Pentium (R) Dual CPU at 1.32 GHz, 1 GB of RAM in Windows XP. To the best of our knowledge, no instances are publicly available for the HFFMCVRP; therefore, we generate test instances based on the existing HFFVRP instances, which themselves are generated from classic VRP instances.

5.1. The HFFMCVRP Instances Sets. The well-known HFFVRP benchmark instances set, the “T-8” set, in Taillard [4], numbered from 13 to 20, is modified to generate our instances. The generating procedure is basically the same as the ones used in El-Fallahi et al. [1] and Muyltermans and Pang [2], which can be summarized as creating two compartments of same capacity or bisecting the capacity of each vehicle and the demand of each customer. As a result, there are two products; each client orders the same amounts of them (consider complementary or correlated products, e.g., two types of petroleum); each vehicle has two compartments; product $i = 1, 2$ must be delivered in compartment $i = 1, 2$ of whichever vehicle. The characteristics of the generated instances (called TC-8) are summarized in Table 1.

The benefit of such generating procedure is that the BKSs of T-8 can serve as benchmarks or target solutions for



s_i initial solution

s_c current solution

s_b best-found solution

s^* local best solution returned by an operator

FIGURE 4: Framework of the RGTS.

algorithms solving TC-8 since feasible T-8 solutions are also feasible TC-8 solutions (the two orders of every customer are shipped together). Besides, the TC-8, though not real life instances, may be representative of them to some extent. For example, in medium sized cities such as Dalian in China, a major oil company typically has around 100 regular customers and 10–20 vehicles, which may be represented by instances 17–20, while the number of customers and number of vehicles in apparel products delivery, food delivery, and waste recycling can be small or big, dependent on the size of the related company/sector (e.g., supermarket or store, common solid waste, or toxic waste). In such cases, instances 13–16 are somewhat representative.

Intuitively, the bisection of a T-8 instance with n customer nodes results in a TC-8 instance with $n' = 2n$ order nodes—in other words, $2n^2 + n$ edges in its corresponding graph while there are only $(n^2 + n)/2$ edges in the graph of the T-8 instance. This fact indicates that a TC-8 instance should have a much larger search space and more local traps than its corresponding T-8 instance.

5.2. Parameter Tuning. Like many other metaheuristics, the RGTS employs a variety of parameters that require a tuning procedure in order to achieve the optimum balance between computational effort and solution quality. Observing the

TABLE 1: Characteristics of TC-8 derived from T-8.

TC-8	n'	Type A Property A	Type B Property B	Type C Property C	Type D Property D	Type E Property E	Type F Property F
13	100	10, 1.0, 4	15, 1.1, 2	20, 1.2, 4	35, 1.7, 4	120, 2.5, 2	200, 3.2, 1
14	100	60, 1.0, 4	80, 1.1, 2	150, 1.4, 1	—	—	—
15	100	25, 1.0, 4	50, 1.6, 3	80, 2, 2	—	—	—
16	100	20, 1.0, 2	40, 1.6, 4	70, 2.1, 3	—	—	—
17	150	25, 1.0, 4	60, 1.2, 4	100, 1.5, 2	175, 1.8, 1	—	—
18	150	10, 1.0, 4	25, 1.3, 4	50, 1.9, 2	75, 2.4, 2	250, 2.9, 1	400, 3.2, 1
19	200	50, 1.0, 4	100, 1.4, 3	150, 1.7, 3	—	—	—
20	200	30, 1.0, 6	70, 1.7, 4	100, 2, 3	—	—	—

n' : number of orders; Type A–F: vehicle types; Property A–F: capacity of each compartment, variable cost per unit of vehicles, and number of vehicles available.

success of RTS in several cases [30–33], we boldly follow the similar parameter settings regarding the reactive mechanism as found in literature. To be prudent, we also carried out extensive tests and finally reached the ones in Table 2. Note that the initial value of TL is set to be $n'/2$ rather than “1” in standard RTS to avoid meaningless adjustment of tabu list size during early-stage iterations. As for the parameters regarding the guiding mechanism, we varied them within relatively wide ranges and tested RGTS’s performance on five randomly generated instances. More specifically, the candidate values of $guidFreq$ are $\{1, 2, \dots, 10\}$, and the candidate values of λ are $\{0.1, 0.2, \dots, 1\}$. Other parameters, after intensive tests, are summarized in Table 2.

TABLE 2: Parameter settings of RGTS.

Parameters	Value
REP	3
$Chaos$	3
Inc	1.1
Dec	0.9
$MaxP$	50
δ	1.0
$MaxNoImp$	50
$GuidFreq$	5
λ	0.1
TL	$n'/2$

5.3. *Results on TC-8.* We have defined three different utility functions above, and now we want to see how they affect the efficiency of the guiding mechanism before we test the RGTS. We conduct comparisons between algorithms with different guiding mechanisms (i.e., different utility functions and penalty terms) and without reactive mechanisms, and the results are shown in Table 3. The first row lists different guiding mechanisms, where “VT” represents the mechanisms using the utility function and penalty term proposed by Voudouris and Tsang [29] and “T” represents the one proposed by Tarantilis et al. [25], while “U1,” “U2,” and “U3” are mechanisms with utility functions U_1 , U_2 , and U_3 , respectively, and the standard penalty term. Other aspects such as the construction procedure, the operators, and the algorithmic routines are defined in Section 4. “OV” stands for the best objective values reached by the algorithms and “CPU” indicates the required running times in seconds.

Table 3 shows that mechanisms “VT” and “T” are outperformed by “U1,” “U2,” and “U3,” which all take $Appr$ into account. This suggests that utilizing the search history information does improve a guiding mechanism’s efficiency. However, the results obtained by “U1,” “U2,” and “U3” are quite close, indicating that once search history information is used to help select undesired edges, edge distance does not matter that much. In fact, it is the “U2” without taking account of any distance information but only search history information returns the best results. After all, edges with long distance are not necessarily undesirable. From now on, we use “U2” without further statements.

To visualize the diversification effect and the effectiveness of the guiding mechanism, we provide Figure 5 to demonstrate the progress of TS with (black line) and without (grey line) the use of the guiding mechanism for Problem 19 of TC-8. It is obvious that without the guiding mechanism, the basic TS (to keep it running longer, the $MaxNoImp$ is set to 100) is trapped after a few iterations and is unable to escape. In contrast, with the guiding mechanism the algorithm seems to be skillful in avoiding local optima and finally leads to a better solution. Similar results are observed in tests of other problems.

The HFFMCVRP is so complex that even the algorithm with “U2” is unable to unveil TC-8 solutions very close to the best-known solutions of the HFFVRP benchmark T-8. So we allow the guiding mechanisms to collaborate with the well-known reactive mechanism (RGTS) and see if such collaboration results in repulse or intimacy between these two mechanisms. In Table 4, we compare RGTS with its reduced versions, that is, TS without either guiding or reactive mechanisms (TS), TS with only reactive mechanism (TS-R), and TS with only guiding mechanism (TS-G).

Table 4 clearly supports that the two mechanisms can coexist and supplement each other. The collaboration of two mechanisms yields the most (average 4.86%) improvement over the basic TS. Note that compared to the BKS of T-8 (see Table 5), the RGTS produces solutions to TC-8 with an average gap 4.06%. Though it is not a small gap, we content with it since TC-8 has quite different structure from T-8

TABLE 3: Comparisons of guiding mechanisms on TC-8.

TC-8	VT		T		U1		U2		U3	
	OV	CPU	OV	CPU	OV	CPU	OV	CPU	OV	CPU
13	1711.68	44	1637.85	58	1588.97	93	1588.97	99	1588.97	85
14	735.16	52	650.91	77	713.75	95	718.90	73	713.75	106
15	1059.11	62	1054.19	65	1043.57	89	1076.14	95	1043.57	91
16	1248.98	41	1207.07	56	1175.01	64	1173.31	82	1263.66	41
17	1170.58	255	1145.69	572	1169.44	166	1169.44	241	1169.44	237
18	2033.98	538	2085.99	452	2036.12	248	1972.22	612	2007.50	610
19	1220.33	1018	1224.31	1170	1197.53	2167	1200.56	1095	1197.53	1746
20	1675.80	359	1711.13	837	1711.00	573	1651.32	1116	1662.58	605
Average	1356.95	296	1339.64	410	1329.42	436	1318.86	426	1330.88	440

TABLE 4: Comparison of RGTS and its reduced versions on TC-8.

TC-8	TS		TS-R			TS-G			RGTS		
	OV	CPU	OV	Gap	CPU	OV	Gap	CPU	OV	Gap	CPU
13	1588.97	81	1686.08	6.11	45	1588.97	0.00	99	1560.97	-1.76	155
14	730.07	103	652.13	-10.68	206	718.9	-1.53	73	625.06	-14.38	148
15	1076.15	118	1077.11	0.09	84	1076.14	0.00	95	1025.76	-4.68	168
16	1225.80	82	1203.39	-1.83	68	1173.31	-4.28	82	1168.25	-4.69	80
17	1169.44	304	1156.89	-1.07	475	1169.44	0.00	241	1114.63	-4.69	725
18	2036.12	329	2013.31	-1.12	421	1972.22	-3.14	612	1939.85	-4.73	908
19	1197.53	1978	1212.17	1.22	790	1200.56	0.25	1095	1186.70	-0.90	2001
20	1680.11	589	1666.95	-0.78	964	1651.32	-1.71	1116	1628.49	-3.07	1345
Average	1338.02	448	1333.50	-1.01	381	1318.86	-1.30	426	1281.21	-4.86	691

Gap (%): gap with respect to the solutions found by the basic TS without reactive and guided mechanisms.

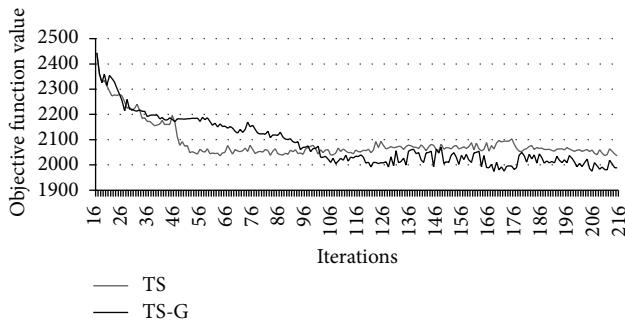


FIGURE 5: Diversification effect of the guiding mechanism.

and is far more complex than T-8; moreover, the BKSs of T-8 are found by several specifically designed algorithms and parameter settings. However, incontestably, there is room for further improvement on the algorithm for the HFFMCVRP.

Since our HFFMCVRP instance set TC-8 is generated from the HFFVRP benchmark set T-8, we wonder how well does our HFFMCVRP algorithm perform in solving T-8. Hence, we apply the RGTS to T-8 with little modification and the same parameter settings, even though it is not intentionally designed for the HFFVRP. According to Table 5, it is fair to say that RGTS is robust, because it unveiled solutions of an average 2.06% gap with regard to the BKSs of T-8 found by four remarkable algorithms. Note that similar

gaps can be observed in the MCVRP literature (see [1–3]); that is, the designed-for-MCVRP algorithms can produce VRP solutions of above 1% gap with respect to the BKSs of the classic VRP in relatively long running times. El Fallahi et al. [1] claimed that the VRP solution is hard to find because of its special structure in a MCVRP context: the two products for each customer must be regrouped in the same trip. Since the HFFVRP is more difficult than the classic VRP because of the heterogeneous fleet, we reckon that 2.06% is expected and acceptable.

6. Conclusions

In this paper, we studied the HFFMCVRP, which is a rich vehicle routing problem commonly found in the practice of multicommodities (e.g., oil, food, fashion, and waste collection) distribution where multicompartment vehicles are indispensable due to safety and health issues. Heterogeneous fleet is almost universal in realistic logistic, and more so when it comes to multicompartment vehicle as the vehicles may have more technical differences. Hence, the HFFMCVRP successfully captures the essence of real life multicommodities distribution. However, not enough attention, which it deserves, has been paid to it.

We propose a hybrid algorithm RGTS to solve the HFFMCVRP. The RGTS systematically integrates a reactive mechanism and a guiding mechanism within the TS framework.

TABLE 5: Comparison of RGTS with HFFVRP algorithms on T-8.

T-8	BKS of T-8	HCG ^a		HRTR ^b		Guided TS ^c		TSA ^d		RGTS		Gap
		Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	OV	CPU	
13	1517.84	0.014	473	0	358	0	116	0	56	1534.72	26	1.11
14	607.53	1.335	575	0	141	0	92	0	55	630.38	57	3.76
15	1015.29	0.155	335	0	166	0	79	0	59	1025.87	48	1.04
16	1144.94	0.796	350	0	188	0	130	0	94	1146.38	61	0.10
17	1061.96	0.926	2245	0	216	0	153	0	206	1071.82	102	0.93
18	1823.58	2.554	2876	0	366	0	252	0	198	1860.01	111	2.00
19	1117.51	0	5833	0.253	404	0.253	327	0.253	243	1172.40	301	4.91
20	1534.17	1.669	3402	0	447	0	479	0	302	1574.24	220	2.61
Average	1227.85	2.614	2011	1228.21	286	1228.21	203	1228.21	152	1251.98	115	2.06
Average Gap	0.000	0.931	—	0.032	—	0.032	—	0.032	—	2.060	—	—

^aHCG: heuristic column generation by Taillard [4] (SucnSparc workstation 50 MHz).

^bHRTR: record-to-record travel algorithm for HFFVRP by Li et al. [5] (Athlon 1 GHz).

^cGuided TS: guided tabu search by Tarantilis et al. [25] (Pentium IV 2.4 GHz).

^dTSA: tabu search algorithm by Brandão [27] (Pentium M 1.4 GHz).

The guiding mechanism is capable of utilizing historic information continuously fed from the search evolution to help more impartially select edges for penalization. Experiments have shown that compared to classic guiding mechanisms employing only predetermined information (i.e., edge distance), our mechanism provided the best results on all generated instances. Moreover, once search history information is used to help select undesired edges, edge distance information may not matter that much. After all, edges with long distance are not necessarily undesirable. To enhance its efficiency, the well-known reactive mechanism is also incorporated in RGTS, and experiments supported that two mechanisms can coexist and supplement each other. The collaboration of two mechanisms yielded an average 4.86% improvement over the basic TS.

For the future research, except for algorithmic issues such as further improvements in the RGTS and designing exact algorithms, one may consider a loading subproblem in the distribution of multiple products in multiple compartments, especially when the compartments are flexible and the products are special (e.g., in shape, form, etc.).

Notations

- L : Node set $\{0, 1, \dots, n\}$ including one depot (node 0)
- L_c : Client set, that is, $L \setminus \{0\}$
- E : Edge set, that is, $\{(i, j) \mid i, j \in L\}$
- d_{ij} : Distance of edge $(i, j) \in E$, $d_{ii} = 0$, $d_{ij} = d_{ji}$, $d_{ij} + d_{jl} \geq d_{il}$
- V : Vehicle set
- M : Vehicle type set
- V_m : Subset of V containing only vehicles of type $m \in M$
- cost^v : Variable cost per unit distance of vehicle $v \in V$
- C_v : Compartment set of vehicle $v \in V$
- Q_c : Capacity of compartment $c \in C_v$, $v \in V$

- P : Product set
- O : Order set
- j^o : Customer who places order $o \in O$, $j^o \in L_c$
- p^o : Product type of order $o \in O$, $p^o \in P$
- q_o : Quantity of order $o \in O$, $q_o > 0$
- S_j^o : Orders placed by customer $j \in L_c$, that is, $\{o \in O \mid j^o = j\}$
- S_p^o : Orders for product $p \in P$, that is, $\{o \in O \mid p^o = p\}$
- I_1 : Set of products incompatibilities, $I_1 \subseteq P \times P$, that is, $(p, q) \in I_1$, indicates that products p and q must not be delivered together in the same compartment
- I_2 : Set of incompatibilities between products and compartments, $I_2 \subseteq P \times C$, that is, $(p, c) \in I_2$, means that products p must not be delivered in compartment c
- b_{ijv} : = 1 if vehicle v travels from node i to j , otherwise = 0
- x_{ovc} : = 1 if order o is assigned to compartment $c \in C_v$, otherwise = 0
- y_{pvc} : = 1 if product p is assigned to compartment $c \in C_v$, otherwise = 0
- u_{iv} : The position of node i in the tour of vehicle v , $u_{iv} = 0$ indicates that node i is never visited by vehicle v .

Parameters

- TL : Tabu list
- LTL : Long tabu list for detecting repetition
- $Turn\delta$: Switch of the neighborhood reduction strategy
- $RCounter$: Counter for repetitions
- REP : Constant threshold for $RCounter$
- $Chaotic$: Counter for $RCounter > REP$
- $Chaos$: Constant threshold for $Chaotic$
- Inc : Percentage increase for TL

<i>Dec</i> :	Percentage decrease for <i>TL</i>
<i>ReptP</i> :	Number of iterations between two consecutive repetitions
<i>MaxP</i> :	Constant threshold for <i>ReptP</i>
<i>TLLastCh</i> :	Number of iterations since the last change of <i>TL</i>
<i>MoveAvg</i> :	Changing threshold for <i>TLLastCh</i>
<i>itr</i> :	Counter used to calculate <i>ReptP</i>
<i>GuidFreq</i> :	Frequency of guiding mechanism
<i>MaxNoImp</i> :	Maximum number of consecutive iterations allowed when the best-found solution has not been improved.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

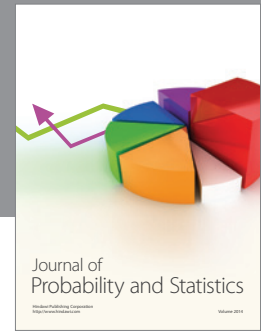
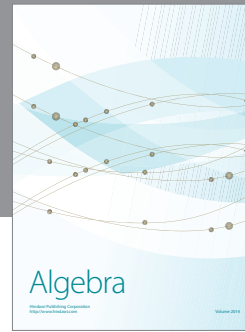
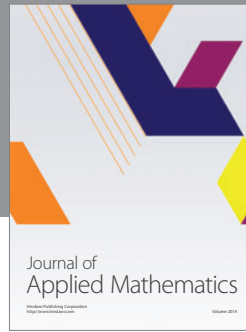
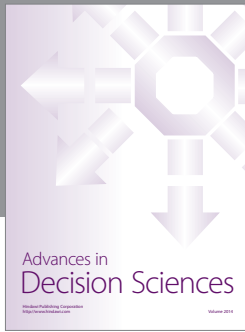
Acknowledgments

This work is supported in part by the Natural Science Foundation of China under Grant no. 70971141, the Natural Science Foundation of Guangdong Province under Grant no. 9151027501000049, the Ministry of Education Humanities and Social Sciences Planned Project under Grant no. 09YGA630156, and the 985 project of Sun Yat-sen University. The authors would like to thank the anonymous referees for their valuable suggestions.

References

- [1] A. E. Fallahi, C. Prins, and R. Wolfler Calvo, "A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem," *Computers and Operations Research*, vol. 35, no. 5, pp. 1725–1741, 2008.
- [2] L. Muyldermans and G. Pang, "On the benefits of co-collection: experiments with a multi-compartment vehicle routing algorithm," *European Journal of Operational Research*, vol. 206, no. 1, pp. 93–103, 2010.
- [3] U. Derigs, J. Gottlieb, J. Kalkoff, M. Piesche, F. Rothlauf, and U. Vogel, "Vehicle routing with compartments: applications, modelling and heuristics," *OR Spectrum*, vol. 33, no. 4, pp. 885–914, 2011.
- [4] E. D. Taillard, "A heuristic column generation method for the heterogeneous fleet VRP," *RAIRO Operations Research*, vol. 33, no. 1, pp. 1–14, 1999.
- [5] F. Li, B. Golden, and E. Wasil, "A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem," *Computers and Operations Research*, vol. 34, no. 9, pp. 2734–2742, 2007.
- [6] B. Golden, A. Assad, L. Levy, and F. Gheysens, "The fleet size and mix vehicle routing problem," *Computers and Operations Research*, vol. 11, no. 1, pp. 49–66, 1984.
- [7] S. Liu, W. Huang, and H. Ma, "An effective genetic algorithm for the fleet size and mix vehicle routing problems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, no. 3, pp. 434–445, 2009.
- [8] X. Li, S. C. H. Leung, and P. Tian, "A multistart adaptive memory-based tabu search algorithm for the heterogeneous fixed fleet open vehicle routing problem," *Expert Systems with Applications*, vol. 39, no. 1, pp. 365–374, 2012.
- [9] M. P. Seixas and A. B. Mendes, "Column generation for a multitrip vehicle routing problem with time windows, driver work hours, and heterogeneous fleet," *Mathematical Problems in Engineering*, vol. 2013, Article ID 824961, 13 pages, 2013.
- [10] A. Hoff, H. Andersson, M. Christiansen, G. Hasle, and A. Løkketangen, "Industrial aspects and literature survey: fleet composition and routing," *Computers and Operations Research*, vol. 37, no. 12, pp. 2041–2061, 2010.
- [11] R. Battiti and G. Tecchiolli, "The reactive tabu search," *INFORMS Journal on Computing*, vol. 6, no. 2, pp. 126–140, 1994.
- [12] F. Cornillier, F. F. Boctor, G. Laporte, and J. Renaud, "A heuristic for the multi-period petrol station replenishment problem," *European Journal of Operational Research*, vol. 191, no. 2, pp. 295–305, 2008.
- [13] P. Avella, M. Boccia, and A. Sforza, "Solving a fuel delivery problem by heuristic and exact approaches," *European Journal of Operational Research*, vol. 152, no. 1, pp. 170–179, 2004.
- [14] G. G. Brown and G. W. Graves, "Real-time dispatch of petroleum tank trucks," *Management Science*, vol. 27, no. 1, pp. 19–32, 1981.
- [15] G. G. Brown, C. J. Ellis, W. G. Graves, and D. Ronen, "Real-time, wide area dispatch of Mobil tank trucks," *Interfaces*, vol. 17, no. 1, pp. 107–120, 1987.
- [16] S. Apotheker, "Curbside collection: complete separation versus commingled collection," *Resource Recycling*, 1990.
- [17] B. Platt and J. Zachary, *Co-Collection of Recyclables and Mixed Waste: Problems and Opportunities*, Institute for Local Self-Reliance, ILSR, Washington, DC, USA, 1992.
- [18] M. Jahre, *Logistics systems for recycling-efficient collection of household waste [Ph.D. thesis]*, Chalmers University of Technology, Göteborg, Sweden, 1995.
- [19] J. Renaud, G. Laporte, and F. F. Boctor, "A tabu search heuristic for the multi-depot vehicle routing problem," *Computers and Operations Research*, vol. 23, no. 3, pp. 229–235, 1996.
- [20] B. L. Golden, A. A. Assad, and E. A. Wasil, "Routing vehicles in the real world: application in the solid waste, beverage, food, dairy, and newspaper industries," in *The Vehicle Routing Problem*, P. Toth and D. Vigo, Eds., vol. 9, pp. 245–286, SIAM, Philadelphia, Pa, USA, 2002.
- [21] Z. Hu, Y. Zhao, and T. M. Choi, "Vehicle routing problem for fashion supply chains with cross-docking," *Mathematical Problems in Engineering*, vol. 2013, Article ID 362980, 10 pages, 2013.
- [22] C. D. Tarantilis, C. T. Kiranoudis, and V. S. Vassiliadis, "A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem," *Journal of the Operational Research Society*, vol. 54, no. 1, pp. 65–71, 2003.
- [23] 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*, vol. 152, no. 1, pp. 148–158, 2004.
- [24] G. Dueck, "Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing," *Journal of Computational Physics*, vol. 90, no. 1, pp. 161–175, 1990.
- [25] C. D. Tarantilis, E. E. Zachariadis, and C. T. Kiranoudis, "A guided tabu search for the heterogeneous vehicle routing problem," *Journal of the Operational Research Society*, vol. 59, no. 12, pp. 1659–1673, 2008.

- [26] X. Li, P. Tian, and Y. P. Aneja, "An adaptive memory programming metaheuristic for the heterogeneous fixed fleet vehicle routing problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 6, pp. 1111–1127, 2010.
- [27] J. Brandão, "A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem," *Computers and Operations Research*, vol. 38, no. 1, pp. 140–151, 2011.
- [28] B. Almada-Lobo, J. F. Oliveira, and M. A. Carravilla, "Production planning and scheduling in the glass container industry: a VNS approach," *International Journal of Production Economics*, vol. 114, no. 1, pp. 363–375, 2008.
- [29] C. Voudouris and E. Tsang, "Guided local search and its application to the traveling salesman problem," *European Journal of Operational Research*, vol. 113, no. 2, pp. 469–499, 1999.
- [30] I. H. Osman and N. A. Wassan, "A reactive tabu search metaheuristic for the vehicle routing problem with back-hauls," *Journal of Scheduling*, vol. 5, no. 4, pp. 263–285, 2002.
- [31] N. A. Wassan, "A reactive tabu search for the vehicle routing problem," *Journal of the Operational Research Society*, vol. 57, no. 1, pp. 111–116, 2006.
- [32] N. A. Wassan, A. H. Wassan, and G. Nagy, "A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries," *Journal of Combinatorial Optimization*, vol. 15, no. 4, pp. 368–386, 2008.
- [33] N. A. Wassan and I. H. Osman, "Tabu search variants for the mix fleet vehicle routing problem," *Journal of the Operational Research Society*, vol. 53, no. 7, pp. 768–782, 2002.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

