Information Sciences 406–407 (2017) 146–169

Contents lists available at ScienceDirect

Information Sciences

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

Multi-vehicle selective pickup and delivery using metaheuristic algorithms

Chuan-Kang Ting*, Xin-Lan Liao, Yu-Hsuan Huang, Rung-Tzuo Liaw

Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 62102, Taiwan

ARTICLE INFO

Article history:
Received 17 December 2015
Revised 21 January 2017
Accepted 3 April 2017
Available online 8 April 2017

Keywords:
Selective pickup and delivery problem
Multiple vehicles
Tabu search
Genetic algorithm
Scatter search

ABSTRACT

The pickup and delivery problem (PDP) addresses real-world problems in logistics and transportation, and establishes a critical class of vehicle routing problems. This study presents a novel variant of the PDP, called the multi-vehicle selective pickup and delivery problem (MVSPDP), and designs three metaheuristic algorithms for this problem. The MVSPDP aims to find the minimum-cost routes for a fleet of vehicles collecting and supplying commodities, subject to the constraints on vehicle capacity and travel distance. The problem formulation features relaxing the requirement of visiting all pickup nodes and enabling multiple vehicles for achieving transportation efficiency. To solve the MVSPDP, we propose three metaheuristic algorithms: tabu search (TS), genetic algorithm (GA), and scatter search (SS). A fixed-length representation is presented to indicate the varying number of vehicles used and the selection of pickup nodes. Furthermore, we devise four operators for TS, GA, and SS to handle the selection of pickup nodes, number of vehicles used, and their routes. The experimental results indicate that the three metaheuristic algorithms can effectively solve the MVSPDP. In particular, TS outperforms GA and SS in solution quality and convergence speed. In addition, the problem formulation produces substantially lower transportation costs than the PDP does, thus validating the utility of the MVSPDP.

1. Introduction

The pickup and delivery problem (PDP) belongs to a critical class of vehicle routing problems and arises in several real-world logistic problems, such as the delivery of packages and letters. In the PDP, customers are classified into pickup nodes and delivery nodes, which represent customers providing and demanding commodities, respectively. The PDP is used to find the shortest route that satisfies the requests of customers. Comprehensive surveys of the PDP are presented in [7,50,51,61]. Parragh et al. [50,51] divided PDP scenarios into transportation between the depot and customers as well as conveyance among customers. The first type involves commodities picked from and delivered to the depot. This one-to-many-to-one (1-M-1) PDP is applicable to reverse logistics in simultaneously managing product distribution from the storehouse and material collection for remanufacture [33,34,55,56]. The second type implements the one-to-one (1-1) PDP structure, through which commodities are transferred between paired pickup and delivery nodes (e.g., dial-a-ride system [18] and message transmission in mobile networks [64]). This type can also be applied in many-to-many (M-M) transportation to supply a set of delivery nodes with commodities collected from numerous pickup nodes [1,8,9,24,37].

* Corresponding author.
E-mail address: ckting@cs.ccu.edu.tw (C.-K. Ting).

http://dx.doi.org/10.1016/j.ins.2017.04.001

In most PDPs, equal amounts of total supply and total demand are assumed, which implicitly imposes a constraint on visiting all customers [7]. Ting and Liao [61] recently proposed the selective pickup and delivery problem (SPDP) by relaxing the constraint that the vehicle must visit all nodes in the PDP. Instead of collecting commodities from all pickup nodes, the SPDP enables the selection of pickup nodes from which to collect sufficient commodities for all delivery nodes. Such a relaxation in visiting all pickup nodes can substantially reduce the transportation cost for cases that focus on satisfying the demand of delivery nodes. The selectability of pickup nodes is particularly useful for the real-world applications dealing with the stock and supply of commodities. For example, in the city bike rental service, arranging the routes for the trucks to transport bikes to the popular spots is a key issue for redistribution of bikes. The SPDP facilitates reducing the transportation cost by picking up bikes from only some stations, rather than visiting all, and then delivering them to the demanded areas. Compared with the PDP, the selectability of pickup nodes in the SPDP permits unequal amounts of total supply and total demand and enables *all* delivery nodes to be supplied efficiently with an adequate number of commodities collected from *some* pickup nodes by using a single vehicle.

This study extends the SPDP in the number of vehicles to improve transportation efficiency. Specifically, we propose a novel problem formulation, called the *multi-vehicle selective pickup and delivery problem* (MVSPDP), aiming to minimize the transportation cost for multiple vehicles to supply all delivery nodes with commodities obtained from some selected pickup nodes, given that each vehicle exhibits identical limits on vehicle capacity and travel distance. Using multiple vehicles can effectually enhance transportation efficiency; however, this extension induces additional constraints, objectives, and logistic characteristics and, therefore, poses considerable challenges to solving the problem. First, two additional constraints must be addressed in the MVSPDP: one constraint requires the vehicle load to be nonnegative and lower than the vehicle capacity along the route; the other constraint imposes the maximum travel distance for workload balance or the refueling requirements of each vehicle. Second, the number of vehicles in use plays an essential role in the resultant routes. Determining the appropriate number of vehicles to use and arranging their routes are key issues in solving the MVSPDP. Third, the selectability of pickup nodes can substantially reduce the cost of the route but increase the problem complexity [61]. An increase in the number of vehicles, nevertheless, compounds the difficulty in selecting suitable pickup nodes for optimal routes.

To solve the MVSPDP, we develop three metaheuristic algorithms: tabu search (TS) [29], genetic algorithm (GA) [39], and scatter search (SS) [28]. These three algorithms have been proved to be effective in resolving numerous search and optimization problems [15,30,46,49]. The key challenge in these algorithms is to design effective operators that select pickup nodes, organize visiting orders, and allocate multiple vehicles to minimize the total transportation cost in accordance with the two constraints. In this study, we design the components and operators for the three metaheuristic algorithms to solve the MVSPDP:

- The representation for candidate solutions enables indicating the varying number of vehicles used and the selection of pickup nodes by using a fixed-length string.
- An evaluation function considering the solution feasibility is proposed to handle the constraints.
- An initialization method based on the sweep algorithm is developed to improve search efficiency.
- Four operators are devised to deal with route planning, the selection of pickup nodes, and the number of vehicles adopted.

In this study, a series of experiments is conducted to examine and compare the performance of the three metaheuristic algorithms in solving the MVSPDP. In addition, the experiments are conducted to investigate the effects of deterministic and adaptive control on the probability of performing a specific operator. The remainder of this paper is organized as follows: Section 2 presents the formulation of the MVSPDP; Section 3 elucidates the proposed TS, GA, and SS methods; and Section 4 provides the experimental results. Finally, Section 5 draws the conclusions of this study.

2. The multi-vehicle selective pickup and delivery problem

This study presents a novel variant of the PDP, called the multi-vehicle selective pickup and delivery problem (MVSPDP). The MVSPDP aims for the shortest routes for a fleet of vehicles that transport commodities from *some* pickup nodes to *all* delivery nodes, subject to the constraints on vehicle capacity and travel distance. The vehicles are regulated to start from and return to the depot. The following subsection presents the problem formulation of the MVSPDP. In addition, we derive a bound of solution values for the MVSPDP.

2.1. Problem formulation

Let $G = (V, E)$ be an undirected complete graph with vertex set $V = (v_0, \ldots, v_n)$ and edge set $E = \{(v_i, v_j) \mid v_i, v_j \in V, v_i \neq v_j\}$, in which each edge (v_i, v_j) has a cost $c_{ij} > 0$ and $c_{ij} = c_{ji}$. The MVSPDP involves one depot v_0 and n customer nodes $v_1, \ldots, v_n$. Let d_i denote the demand of node v_i. According to the demand, the vertex set V is divided into a set of *pickup* nodes $V^+ = \{v_i \mid v_i \in V, d_i > 0\}$, a set of *delivery* nodes $V^- = \{v_i \mid v_i \in V, d_i < 0\}$, and the *depot* v_0 with $d_0 = 0$. All m vehicles are assumed to exhibit the same capacity Q and maximal travel distance permitted R.

Resolving the MVSPDP requires selecting pickup nodes and arranging visiting orders such that multiple vehicles supply all delivery nodes with sufficient commodities. Let binary decision variable $x_{ijk} = 1$ if the kth vehicle travels from v_i to v_j in a

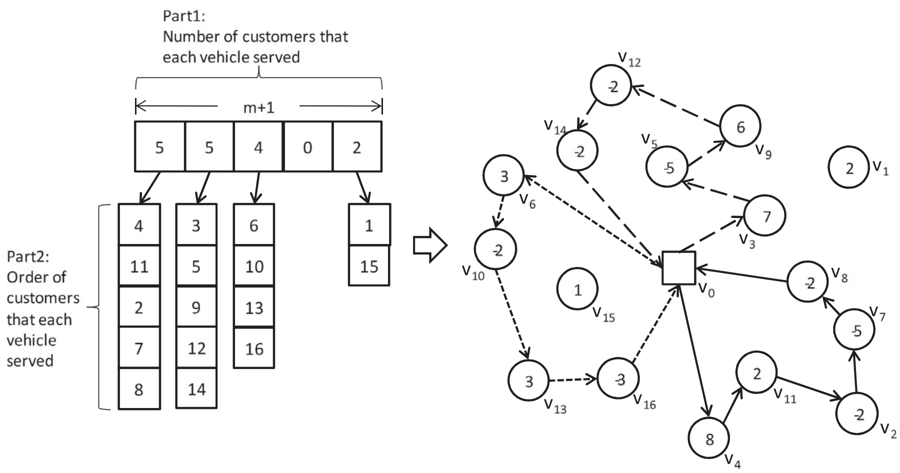


Fig. 1. An example representation for an MVSPDP instance with $n = 16$ customer nodes and $m = 4$ vehicles. The number inside circles denotes demand d_i. The solid, dashed, and dotted lines represent the routes of three vehicles.

directed path; otherwise, $x_{ijk} = 0$. Another binary variable y_{ik} is associated with the visiting of a vehicle to a particular node: $y_{ik} = 1$ if the kth vehicle visits v_i, and $y_{ik} = 0$ otherwise. Let $V^{\pm} = V^+ \cup V^-$, the objective and constraints of the MVSPDP are formulated as follows:

$$\min \sum_{k=1}^{m} \sum_{v_i, v_j \in V} c_{ij} x_{ijk} \tag{1}$$

s.t.

$$\sum_{v_j \in V} x_{ijk} = \sum_{v_j \in V} x_{jik} = y_{ik}, \ \forall v_i \in V, k \in \{1, \ldots, m\} \tag{2}$$

$$\sum_{k=1}^{m} y_{ik} = 1, \ \forall v_i \in V^- \tag{3}$$

$$\sum_{k=1}^{m} y_{ik} \le 1, \ \forall v_i \in V^+ \tag{4}$$

$$\sum_{k=1}^{m} y_{0k} \le m \tag{5}$$

$$\sum_{v_i, v_j \in V} c_{ij} x_{ijk} \le R, \ \forall k \in \{1, \ldots, m\} \tag{6}$$

$$y_{sk} \le \sum_{v_i \in S} \sum_{v_j \notin S} x_{ijk}, \ \forall S \subseteq V^{\pm}, v_s \in S, k \in \{1, \ldots, m\} \tag{7}$$

$$z_{Sk} = \lfloor \frac{\sum_{v_i, v_j \in S} x_{ijk}}{\sum_{v_i \in S} y_{ik} - 1} \rfloor \sum_{v_j \in S} x_{0jk}, \ \forall S \subseteq V^{\pm}, k \in \{1, \ldots, m\} \tag{8}$$

$$0 \le z_{Sk} \sum_{v_i \in S} d_i y_{ik} \le Q, \ \forall S \subseteq V^{\pm}, k \in \{1, \ldots, m\} \tag{9}$$

Constraint (2) restricts incoming and outgoing flows to be equal and gives the value of decision variable y_{ik}. Constraint (3) guarantees that each delivery node can be visited exactly once, and constraint (4) enables selectability of pickup nodes. The number of available vehicles is limited to m as specified in (5). Constraint (6) further limits the maximal travel distance R permitted for each vehicle. The subtour elimination for each route is presented in constraint (7), which requires that each subset of customers assigned to a vehicle has at least one flow going out of that subset. Regarding (8), the floor part considers the connection of nodes in the subset S; that is, the number of traversed edges in a route equals the number of visited nodes minus one, i.e., $\sum_{v_i, v_j \in S} x_{ijk} = \sum_{v_i \in S} y_{ik} - 1$. Hence, the variable $z_{Sk} = 1$ if there exists an incoming flow issued from v_0 to the subsequent connected route of the kth vehicle; otherwise, $z_{Sk} = 0$. For example, considering the route of the third vehicle in Fig. 1, i.e., $v_0 \rightarrow v_6 \rightarrow v_{10} \rightarrow v_{13} \rightarrow v_{16} \rightarrow v_0$, the connected nodes $S = \{v_6, v_{10}\}$, $\{v_6, v_{10}, v_{13}\}$, and $\{v_6, v_{10}, v_{13}, v_{16}\}$ have $\sum_{v_i, v_j \in S} x_{ij3} = \sum_{v_i \in S} y_{i3} - 1 = 1$, 2, and 3, respectively, which give $z_{S3} = 1$ in that $x_{063} = 1$. By contrast, the unconnected nodes in the route, e.g. $S = \{v_{10}, v_{16}\}$, result in $z_{S3} = 0$ due to $\sum_{v_i,v_j\in S} x_{ij3} < \sum_{v_i\in S} y_{i3} - 1$ as well as $\sum_{v_j\in S} x_{0j3} = 0$. Accordingly, constraint (9) confines the vehicle load within capacity Q.

The MVSPDP relaxes the requirement for visiting all pickup nodes and uses multiple vehicles. This problem formulation is pertinent to real-world logistic applications, particularly for enabling some suppliers (pickup nodes) to satisfy the demands of all customers (delivery nodes) by using multiple vehicles. Furthermore, the selectability of pickup nodes can substantially reduce the transportation cost in the PDP. Some problems are related to but differ from the SPDP. The single-commodity vehicle routing problem with pickup and delivery service (1-VRPPD) [43] uses multiple vehicles to solve the one-commodity pickup-and-delivery problem [37]. The 1-VRPPD involves locating minimum-cost routes to transport homogeneous commodities among unpaired customers, according to the constraints that each node must be visited exactly once and the depot can balance the total demand. Falcon et al. [25] applied this problem to manage carrier-based coverage repair in wireless sensor networks, in which pickup nodes are optional. The principal difference between these problems and the MVSPDP lies in the depot capability and unitary demand. The linehaul-feeder vehicle routing problem with virtual depots [12] enables selectability of reloading spots; virtual depots refer to public parking lots in which a large vehicle dispatches commodities to small vehicles for relaying delivery in narrow streets. By contrast, in the MVSPDP, each node is visited at most once to collect a sufficient number of commodities and serve all of the delivery customers.

Furthermore, the MVSPDP can be viewed as a combination of the SPDP and the vehicle routing problem (VRP); therefore, the properties of both the SPDP and VRP must be considered in resolving the MVSPDP. For example, the VRP accounts for the number of vehicles and depots [23,40], time windows [6,32,42,59,66], workload balance [36], and dynamics [44]. In 1-1 transportation, the precedence constraint guarantees that a requested pickup node is visited before reaching the designated destination [4,47]. Additional constraints on the duration of a single route and the number of passengers disembarking at a site exist in some industries; for example, oil companies are subject to refueling requirements and security concerns on the production platform [63]. The multiple requests in emergency transportation operations and the control of automated guided vehicle dispatching in manufacturing create additional constraints or objectives regarding the passenger ride time, occupancy rate, and resource required for service quality [26,48,52]. In addition to load splitting, which allows multiple vehicles to serve mutual customers in the VRP, pairing transportation with transfer opportunity enables goods to be stored temporarily in transshipment nodes, which are visited several times by multiple vehicles to satisfy requests [20,53]. The 1-M-1 PDP is a generalization of the capacitated vehicle routing problem and presents various limitations on solution types [7], including Hamiltonian [5,13,31], delivery-first pickup-second [62,65], and lasso tours [38]. Zhu et al. [67] recently considered the 1-M-1 PDP with three objectives and dynamic requests. Regarding intermodal shipments using containers, individual trips are permitted when containers are fully loaded, and a delivery trip and a pickup trip can be merged if the time and container size are compatible [58]. Noteworthily, compensation for the travel cost is considered when selecting pickups in the 1-M-1 structure [35], whereas selection of pickups is executed in the MVSPDP to collect a sufficient number of commodities for supplying delivery nodes.

2.2. Bound from the minimum k-degree center tree

This subsection attempts to bound the values of solutions to the MVSPDP. Christofides et al. [16] derived a bound for the multiple traveling salesman problem (M-TSP), which can also serve as a lower bound for the VRP because the VRP can be viewed as an M-TSP with additional constraints. In light of the fact that MVSPDP is a generalization of the VRP, we use their method to compute the bound for the MVSPDP. Specifically, a solution to the M-TSP is decomposed into three sets, i.e., S_t, S_0, and S_1, where S_t consists of the edges forming a k-degree center tree (k-DCT) in the solution, and S_0 and S_1 comprise the remaining edges connected to v_0 and those between customer nodes, respectively. A k-DCT is a tree whose branches connect to all vertices in G and the degree of v_0 is k. The number of available vehicles limits the maximal number of edges in S_0, i.e., $|S_0| \le m$, leading to $k = 2m - |S_0|$. Let binary decision variable $\xi^t_{ij} = 1$ if (v_i, v_j) is in the edge set S_t, and $\xi^t_{ij} = 0$ otherwise; in addition, let ξ^0_{ij} and ξ^1_{ij} denote such binary decision variables for S_0 and S_1, respectively. According to Christofides et al. [16], the M-TSP can be formulated as follows:

$$\min \sum_{v_i,v_j\in V} c_{ij}(\xi^t_{ij} + \xi^0_{ij} + \xi^1_{ij}) \tag{10}$$

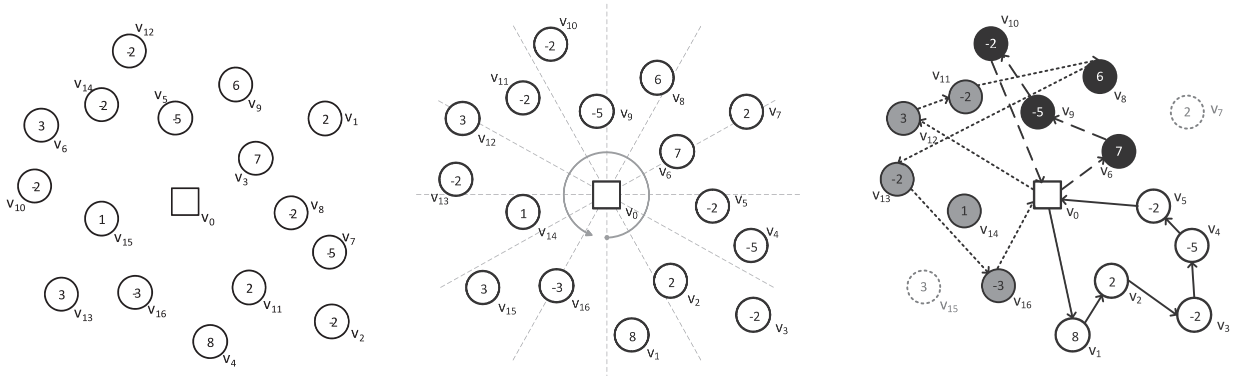
s.t.

$$\sum_{v_i\in S, v_j\notin S} \xi^t_{ij} \ge 1, \quad \forall S \subset V, S \ne \emptyset \tag{11}$$

$$\sum_{v_j\in V} \xi^t_{0j} = 2m - y \tag{12}$$

$$\sum_{v_i,v_j\in V} \xi^t_{ij} = n \tag{13}$$

$$\sum_{v_j\in V} \xi^0_{0j} = y, \quad y \in \mathbb{Z}^* \tag{14}$$



(a) An instance of MVSPDP. (b) Sorting by polar-coordinate angle. (c) An initial solution.

Fig. 2. An example initialization for an MVSPDP instance with $n = 16$, $\hat{m}' = 3$, and $\delta = 7.67$. The number inside circles denotes demand d_i. The grounds of the circles indicate respective partitions. The solid, dashed, and dotted lines represent the routes of three vehicles.

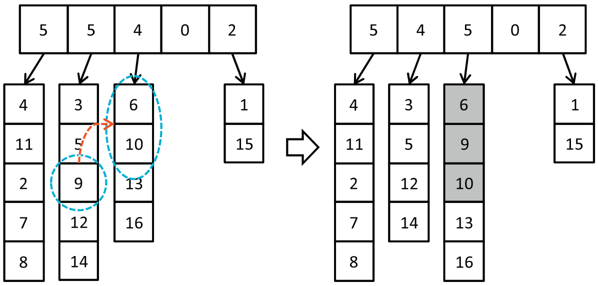


Fig. 3. Example of relocation. The dotted circles indicate the node and target place chosen for relocation.

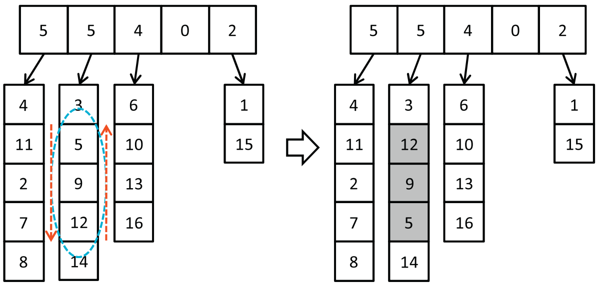


Fig. 4. Example of inversion. The dotted circle indicates the part to be inverted.

$$\sum_{v_i, v_j \in V^{\pm}} \xi_{ij}^1 = m - y \tag{15}$$

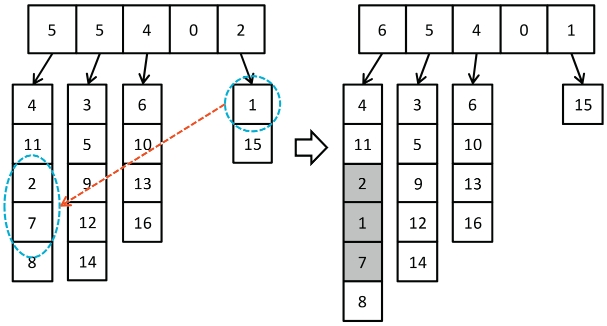
$$\sum_{v_j \in V} (\xi_{ij}^t + \xi_{ij}^0 + \xi_{ij}^1) = 2, \quad \forall v_i \in V^{\pm} \tag{16}$$

Constraint (11) guarantees the connectivity of the k-DCT, in which the degree of v_0 is limited by constraint (12). Constraints (13), (14), and (15) specify the numbers of edges required for S_t, S_0, and S_1, respectively. Constraint (16) confines the degree of each vertex in an M-TSP to two, except for v_0.

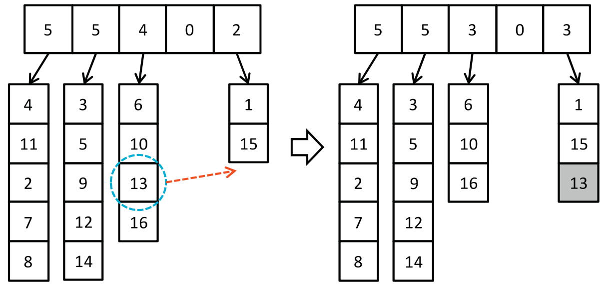
The Lagrangian relaxation produces three decomposed problems P_t, P_0, and P_1 for each candidate value of latent variable y and introduces constraint (16) into a general objective:

$$g^{\kappa}(\boldsymbol{p}, y) = \sum_{v_i, v_j \in V} (c_{ij} + p_i + p_j)\xi_{ij}^{\kappa} - 2\sum_{v_i \in V^{\pm}} p_i\,, \tag{17}$$

where $\boldsymbol{p} = (p_1, \ldots, p_n)$ represents the non-negative penalties associated with the n customer nodes, and $\kappa \in \{t, 0, 1\}$ denotes the index of ξ_{ij}^t, ξ_{ij}^0, and ξ_{ij}^1. For a given value of y, let $g^t(\boldsymbol{p}, y)$ be the value of optimal solution to P_t defined by (17), (11), (12), and (13); let $g^0(\boldsymbol{p}, y)$ be the value of optimal solution to P_0 defined by (17) and (14); and let $g^1(\boldsymbol{p}, y)$ be the value of



(a) Selecting a pickup node. The dotted circle indicates the pickup node selected by the operator.



(b) Unselecting a pickup node. The dotted circle indicates the pickup node unselected by the operator.

Fig. 5. Example of selection: a) selecting a pickup node; b) unselecting a pickup node.

optimal solution to P_1 defined by (17) and (15). A lower bound to the M-TSP can be computed by

$$\max_{y \le m} \max_{\boldsymbol{p} \ge 0} \left(g^t(\boldsymbol{p}, y) + g^0(\boldsymbol{p}, y) + g^1(\boldsymbol{p}, y) \right). \tag{18}$$

Note that this lower bound is valid for the VRP and the MVSPDP, albeit omitting the constraints on the vehicle load and travel distance. Restated, the values of feasible solutions to these two problems must be no smaller than this lower bound; otherwise, contradiction will occur during the approximation of the decomposed problems.

3. Metaheuristics for the MVSPDP

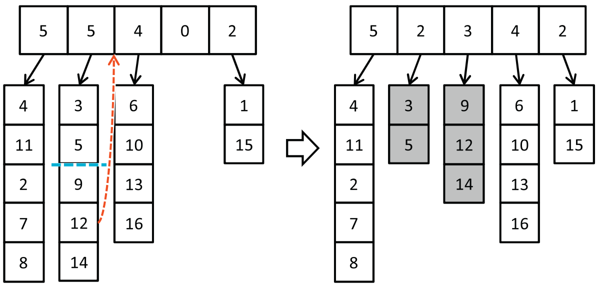
This study develops three metaheuristic algorithms for the MVSPDP: tabu search (TS), genetic algorithm (GA), and scatter search (SS). A fixed-length representation is presented for the three algorithms to accommodate the MVSPDP. TS is initialized using the modified sweep algorithm, whereas GA and SS are initialized by combining the modified sweep algorithm with random initialization to increase the population diversity.

In addition, we propose four operators serving as the neighborhood functions in TS, mutation operators in GA, and improvement operators in SS. Using multiple search operators in metaheuristic algorithms has shown to be helpful for exploring different structures in a problem and achieved considerable successes [14,54,57,60]. In view of the benefits, this study devises four operators for the three algorithms to deal with the visiting order, selection of pickup nodes, and the number of vehicles adopted.

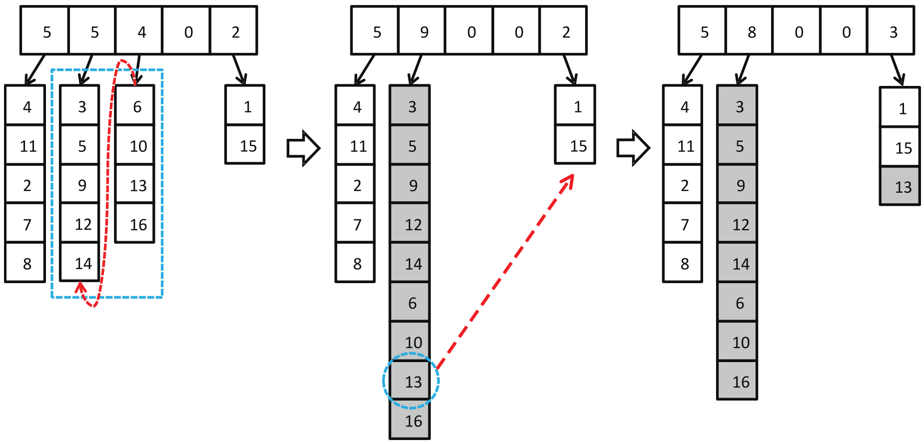
The designs of the proposed algorithms are detailed in the subsequent sections.

3.1. Tabu search

Tabu search is a metaheuristic algorithm known for its effectiveness in combinatorial optimization problems. This metaheuristic algorithm uses an explicit memory structure to record the search trajectory. Based on recorded information, TS guides the search considering both intensification and diversification. In TS, the search is processed using a series of moves from a solution to its neighbor, where the neighborhood is generally defined by a distance function, such as the Hamming distance. The tabu-versus-aspiration strategy controls the moves: On the one hand, TS uses the tabu list to record the forbidden moves and prevents the search from being mired in the local optima by indicating certain moves as *tabu*. The tabu tenure affects the level of restriction on choosing the forbidden moves; long tabu tenure guides the search to explore



(a) Adding a route. The dashed line indicates the cut point.



(b) Deleting a route. The dashed square indicates the routes selected to merge.

Fig. 6. Example of addition/deletion.

unvisited territory, encouraging diversification. On the other hand, the aspiration criterion provides the opportunity to override the tabu restriction. That is, the aspiration criterion enables superior neighboring solutions to be chosen, despite the restriction of tabu moves. This criterion supports the ability of intensification for the search.

Designing TS involves solution representation, an evaluation function, initialization, a neighborhood function, and an adaptive search phase. The following subsections describe our designs of TS for the MVSPDP .

3.1.1. Representation and evaluation function

This study presents a representation for candidate solutions to the MVSPDP based on the chromosome representation of Tan et al. [57]. In addition to indicating the visiting orders of nodes for vehicles, the proposed representation considers selecting pickup nodes in the MVSPDP. Specifically, the representation consists of two parts: The first part comprises $m+1$ nonnegative integers, of which the first m integers indicate the number of nodes for m vehicles to visit, and the final integer provides the number of unselected pickup nodes. In the example representation in Fig. 1, the first value “5” indicates that the first vehicle must visit five nodes, while the fourth value “0” signifies that the fourth vehicle is unused and, therefore, has no corresponding route. The second part of the representation indicates the visiting order of nodes for each vehicle. The final sequence records the unselected pickup nodes, which are not visited by any vehicle. Noteworthily, although the $m+1$ sequences are variable-length, the sum of their lengths is fixed to the number of nodes n. Hence, the proposed representation applies a fixed-length encoding, which is capable of indicating the varying number of vehicles used and the selection of pickup nodes.

In TS, the evaluation function determines the promising directions and guides the search toward the optima. The proposed evaluation function considers the objective of the MVSPDP and deals with infeasible solutions. Formally, the evaluation function for a solution $\boldsymbol{x}$ is defined as

$$f(\boldsymbol{x}) = \sum_{k=1}^{m} \sum_{v_i, v_j \in V} c_{ij} x_{ijk} + \alpha p_d + \beta \bar{c} p_l, \tag{19}$$

where $\bar{c}$ is the average cost among all edges, and α and β denote the weights for penalizing the constraint violation on loading and distance; both weights are set to 2 in this study. The penalty for violating load constraint p_l counts the number of times that vehicles are overloaded or have insufficient commodities on board, and penalty p_d accounts for the travel

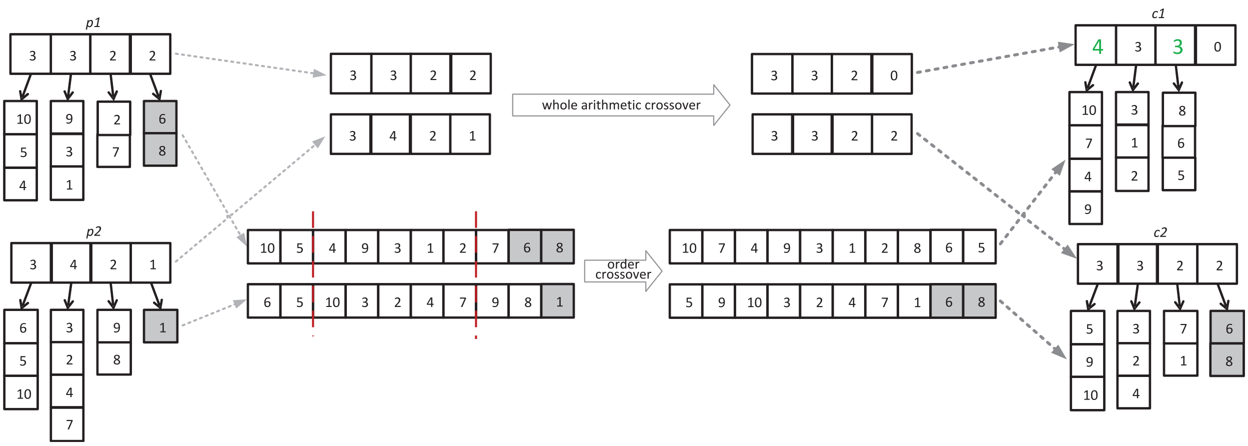


Fig. 7. Crossover for the proposed GA: whole arithmetic crossover for the first part and order crossover for the second part. The gray squares denote the unselected pickup nodes, and the dashed lines indicate the cut points for order crossover. The green numbers in the resultant offspring highlight the changes in the number of visited nodes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

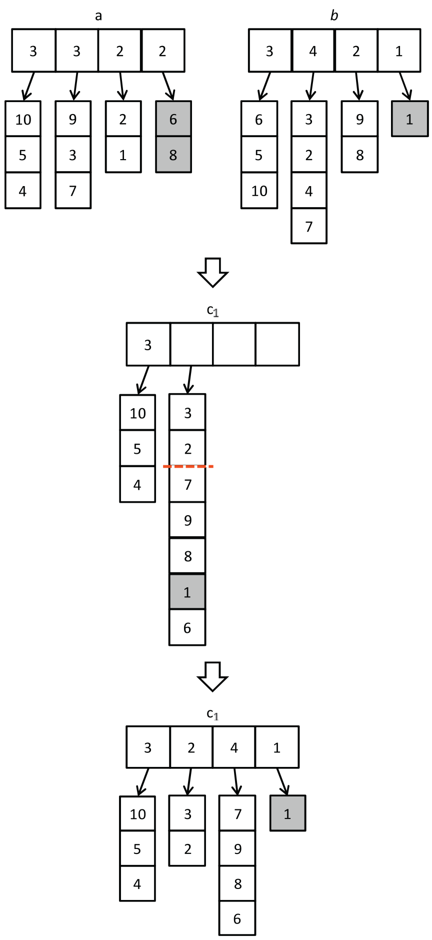


Fig. 8. Example of SS creating a solution from a and b. The dotted line indicates the cut point.

Table 1
MVSPDP test instances.

Instance	Modified from	m	Q	R	#customers	
					Pickup $\|V^+\|$	Delivery $\|V^-\|$
mvspdp50a	vrpnc01	6	85	180	26	24
mvspdp50b	vrpnc06	7	85	260	27	23
mvspdp75a	vrpnc02	12	85	190	39	36
mvspdp75b	vrpnc07	13	85	230	41	34
mvspdp100a	vrpnc03	10	105	280	49	51
mvspdp100b	vrpnc08	10	95	290	44	56
mvspdp100c	vrpnc12	14	140	300	46	54
mvspdp100d	vrpnc14	11	115	1100	46	54
mvspdp150a	vrpnc04	16	130	280	69	81
mvspdp150b	vrpnc09	16	100	300	73	77
mvspdp199a	vrpnc05	22	110	300	96	103
mvspdp199b	vrpnc10	22	105	300	91	108
mvspdp120a	vrpnc11	10	110	350	62	58
mvspdp120b	vrpnc13	13	65	800	65	55
mvspdp200	kelly05	6	450	3000	98	102
mvspdp240	kelly01	13	366	1083	128	112
mvspdp280	kelly06	11	450	2500	142	138
mvspdp320	kelly02	13	350	1500	172	148
mvspdp360	kelly07	12	450	2166	188	172
mvspdp400	kelly03	13	450	2400	202	198
mvspdp440	kelly08	15	600	2000	231	209
mvspdp480	kelly04	14	500	2666	238	242

Table 2
Parameter setting.

Parameter	Value
(a) Tabu search.	
Representation	Integer + order
Initialization	Modified sweep algorithm
Neighborhood function	Relocation, inversion, selection, addition/deletion
Neighborhood search	Random
Tabu tenure	Adaptive
Neighborhood Size	200
Termination	1500 iterations
#Evaluations per iteration	200
Total #evaluations	300,000
(b) Genetic algorithm.	
Representation	Integer + order
Initialization	Modified sweep algorithm (50%), random (50%)
Recombination (p_c)	Whole arithmetic (0.9), order crossover (0.9)
Mutation (p_m)	Relocation (0.6), inversion (0.6), selection ($1/\|V^+\|$), addition/deletion ($1/m$)
Parent selection	2-tournament
Survival selection	$\mu + \lambda$
Population size	100
Termination	3000 generations
#Evaluations per generation	100
Total #evaluations	300,000
(c) Scatter search.	
Representation	Integer + order
Initialization	Modified sweep algorithm (50%), random (50%)
Population size	25
Reference size	5
Improvement method	Hill climbing (HC)
HC neighborhood function	Relocation, inversion, selection, addition/deletion
HC neighborhood size	100
HC termination	10 iterations
Termination	60 generations
#Evaluations per generation	5010
Total #evaluations	325,625

Table 3
The average route length (avg.), feasible rate (feas.), and average running time (time) for the TS, GA, and SS using the *fixed* probability of applying the four operators on the MVSPDP instances with $\gamma = 0$. Boldface marks the best result in each instance.

Instance	TS-fixed			GA-fixed			SS-fixed		
	Avg.	Feas. (%)	Time (s)	Avg.	Feas. (%)	Time (s)	Avg.	Feas. (%)	Time (s)
mvspdp50a	**473**	90	0.16	**475**	**93**	2.15	**473**	73	0.19
mvspdp50b	**434**	**93**	0.21	444	**93**	3.12	444	87	0.24
mvspdp75a	**622**	**93**	0.19	**642**	77	4.95	668	83	0.22
mvspdp75b	**568**	**97**	0.23	**572**	90	6.15	601	**97**	0.25
mvspdp100a	**723**	**83**	0.28	757	53	8.35	810	80	0.29
mvspdp100b	**716**	**93**	0.30	739	**93**	8.53	745	90	0.33
mvspdp100c	**656**	87	0.34	**629**	90	9.34	**625**	**93**	0.58
mvspdp100d	529	**100**	0.40	**500**	**100**	8.51	522	**100**	0.61
mvspdp150a	**833**	**87**	0.34	921	80	17.51	995	**87**	0.47
mvspdp150b	**853**	**97**	0.35	886	87	17.40	985	83	0.47
mvspdp199a	**938**	**97**	0.40	1090	83	28.85	1169	50	0.84
mvspdp199b	**978**	**90**	0.40	1142	87	29.21	1149	87	0.85
mvspdp120a	**1186**	**97**	0.25	1623	50	11.98	1372	37	0.46
mvspdp120b	**693**	**93**	0.50	825	80	12.12	800	90	0.76
mvspdp200	**8440**	**17**	0.77	11,109	3	24.16	–	0	–
mvspdp240	**6584**	**90**	0.40	–	0	–	–	0	–
mvspdp280	**10,518**	**100**	0.63	–	0	–	–	0	–
mvspdp320	**10,359**	**100**	0.52	–	0	–	–	0	–
mvspdp360	**12,745**	**97**	0.59	–	0	–	–	0	–
mvspdp400	**14,954**	**97**	0.70	–	0	–	–	0	–
mvspdp440	**15,751**	**93**	0.58	–	0	–	–	0	–
mvspdp480	**17,798**	**90**	0.64	–	0	–	24,690	3	5.43

Table 4
The average route length (avg.), feasible rate (feas.), and average running time (time) for the TS, GA, and SS using the *fixed* probability of applying the four operators on the MVSPDP instances with $\gamma = 32$. Boldface marks the best result in each instance.

Instance	TS-fixed			GA-fixed			SS-fixed		
	Avg.	Feas. (%)	Time (s)	Avg.	Feas. (%)	Time (s)	Avg.	Feas. (%)	Time (s)
mvspdp50a	**404**	**97**	0.15	**406**	93	2.36	**407**	**97**	0.17
mvspdp50b	**365**	**100**	0.20	**363**	**100**	3.08	373	**100**	0.21
mvspdp75a	**496**	**93**	0.19	**500**	**93**	6.07	510	87	0.22
mvspdp75b	**442**	**93**	0.23	**453**	**93**	6.07	**448**	87	0.25
mvspdp100a	**564**	**100**	0.27	**558**	**100**	8.61	**570**	93	0.31
mvspdp100b	**603**	**100**	0.28	**602**	93	7.94	611	**100**	0.33
mvspdp100c	515	**97**	0.43	514	**97**	8.51	**503**	90	0.45
mvspdp100d	464	**100**	0.49	**452**	**100**	7.90	461	**100**	0.44
mvspdp150a	**666**	**100**	0.34	690	97	17.33	**673**	**100**	0.43
mvspdp150b	**664**	**100**	0.37	690	90	17.02	679	93	0.43
mvspdp199a	**781**	**97**	0.39	814	90	28.96	797	**97**	0.55
mvspdp199b	**812**	**100**	0.39	862	93	27.93	837	90	0.58
mvspdp120a	**775**	**100**	0.25	807	93	12.00	**793**	93	0.34
mvspdp120b	**563**	**100**	0.55	**579**	**100**	12.62	625	**100**	0.49
mvspdp200	**6190**	**100**	0.70	7112	93	21.64	6982	97	0.84
mvspdp240	**4529**	**97**	0.38	5804	83	30.53	6943	63	1.18
mvspdp280	**7560**	**97**	0.82	10,074	90	37.50	9841	90	1.30
mvspdp320	**6512**	**97**	0.57	9354	73	50.43	10,825	53	1.95
mvspdp360	**8643**	**93**	0.54	13,725	70	61.40	13,602	10	2.48
mvspdp400	**9299**	**100**	0.76	15,611	87	73.16	24,569	23	3.24
mvspdp440	**10,066**	**97**	0.64	17,579	57	89.72	11,913	93	3.37
mvspdp480	**12,320**	**93**	0.82	21,865	57	99.08	15,179	**93**	3.42

distance exceeding the limitation R. That is,

$$p_l = \sum_{k=1}^{m} \sum_{S \subseteq V^{\pm}} \ell_{Sk}, \tag{20}$$

$$p_d = \sum_{k=1}^{m} \max \left\{ 0, \sum_{v_i, v_j \in V} c_{ij} x_{ijk} - R \right\}, \tag{21}$$

Table 5
The average route length (avg.), feasible rate (feas.) in percentage, and average running time (time) in seconds, for the TS, GA, and SS using the *adaptive* probability of applying the four operators, in comparison with CPLEX, on the MVSPDP instances with $\gamma = 0$. Boldface marks the best result in each instance.

Instance	TS-adaptive			GA-adaptive			SS-adaptive			CPLEX	
	Avg.	Feas.	Time	Avg.	Feas.	Time	Avg.	Feas.	Time	Length	Time
mvspdp50a	**477**	73	0.15	**468**	**97**	1.84	482	87	0.17	480	289408
mvspdp50b	**433**	**100**	0.20	**437**	**100**	2.99	447	97	0.23	428	321872
mvspdp75a	**623**	77	0.17	688	73	3.57	**667**	**83**	0.22	760	1218342
mvspdp75b	**562**	**93**	0.21	635	80	5.50	590	**93**	0.24	767	1345694
mvspdp100a	**720**	73	0.24	815	17	8.15	**806**	**80**	0.29	–	–
mvspdp100b	**715**	93	0.25	778	**100**	7.99	747	93	0.35	–	–
mvspdp100c	**621**	87	0.46	714	37	7.99	**614**	**90**	0.46	–	–
mvspdp100d	508	**100**	0.65	**499**	**100**	8.16	519	**100**	0.70	–	–
mvspdp150a	**827**	**97**	0.29	1207	23	16.19	**1080**	77	0.48	–	–
mvspdp150b	**808**	**100**	0.31	1258	10	16.54	911	73	0.68	–	–
mvspdp199a	**898**	**100**	0.35	1636	10	26.96	**1232**	43	0.85	–	–
mvspdp199b	**927**	**97**	0.36	1713	13	26.33	1291	57	1.03	–	–
mvspdp120a	**1152**	**100**	0.30	1680	10	11.66	1565	57	0.54	–	–
mvspdp120b	**629**	**100**	0.76	1336	30	11.84	767	**100**	0.77	–	–
mvspdp200	**8288**	**17**	0.46	–	0	–	–	0	–	–	–
mvspdp240	**6490**	**93**	0.40	–	0	–	–	0	–	–	–
mvspdp280	**10,367**	**100**	0.45	–	0	–	–	0	–	–	–
mvspdp320	**10,112**	**93**	0.49	–	0	–	–	0	–	–	–
mvspdp360	**12,373**	**100**	0.53	–	0	–	–	0	–	–	–
mvspdp400	**14,296**	**100**	0.55	–	0	–	–	0	–	–	–
mvspdp440	**15,228**	**100**	0.55	–	0	–	–	0	–	–	–
mvspdp480	**17,357**	**97**	0.65	–	0	–	22,260	3	4.07	–	–

Table 6
The average route length (avg.), feasible rate (feas.) in percentage, and average running time (time) in seconds, for the TS, GA, and SS using the *adaptive* probability of applying the four operators, in comparison with CPLEX, on the MVSPDP instances with $\gamma = 32$. Boldface marks the best result in each instance.

Instance	TS-adaptive			GA-adaptive			SS-adaptive			CPLEX	
	Avg.	Feas.	Time	Avg.	Feas.	Time	Avg.	Feas.	Time	Length	Time
mvspdp50a	**405**	90	0.14	**399**	**97**	3.43	**407**	90	0.18	361	298319
mvspdp50b	369	**100**	0.18	**363**	**100**	3.45	373	**100**	0.22	343	327830
mvspdp75a	**496**	90	0.18	536	**97**	6.11	509	90	0.23	458	1376643
mvspdp75b	**453**	**100**	0.18	471	97	6.33	**458**	97	0.25	392	1485938
mvspdp100a	**560**	93	0.24	596	**97**	9.36	574	93	0.31	–	–
mvspdp100b	**595**	97	0.24	629	93	9.05	609	**100**	0.33	–	–
mvspdp100c	**515**	**93**	0.31	561	83	9.10	505	90	0.61	–	–
mvspdp100d	**459**	**100**	0.44	**456**	**100**	9.50	**458**	**100**	0.64	–	–
mvspdp150a	**654**	93	0.29	824	**100**	18.41	685	97	0.41	–	–
mvspdp150b	**645**	90	0.31	804	**100**	19.04	679	97	0.41	–	–
mvspdp199a	**761**	90	0.34	1078	**100**	31.78	801	100	0.55	–	–
mvspdp199b	**793**	93	0.36	1157	97	30.83	829	**100**	0.61	–	–
mvspdp120a	**785**	**100**	0.24	1041	93	14.82	**784**	**100**	0.45	–	–
mvspdp120b	**528**	**100**	0.43	847	**100**	15.42	605	**100**	0.61	–	–
mvspdp200	**5937**	**100**	0.54	7164	**100**	30.26	6758	97	0.76	–	–
mvspdp240	**4481**	**97**	0.37	5849	93	45.25	6433	63	1.29	–	–
mvspdp280	**7297**	90	0.60	10,034	**100**	57.91	8730	80	1.21	–	–
mvspdp320	**6341**	97	0.39	8674	**100**	78.93	9713	47	1.88	–	–
mvspdp360	**8391**	97	0.50	13,333	**100**	98.19	16,780	3	2.64	–	–
mvspdp400	**8938**	**97**	0.63	14,025	**97**	112.96	24,180	30	3.23	–	–
mvspdp440	**9677**	**97**	0.54	17,557	67	147.43	10,976	93	3.03	–	–
mvspdp480	**12,017**	**100**	0.70	20,471	73	134.87	13,348	90	2.75	–	–

where

$$\ell_{Sk} = \begin{cases} 0 & 0 \le z_{Sk} \sum_{v_i \in S} d_i y_{ik} \le Q, \\ 1 & \text{otherwise}. \end{cases} \tag{22}$$

The evaluation function (19) considers route length as well as the penalties for violating the constraints on vehicle load and maximal travel distance. Accordingly, TS tends to locate the shortest feasible routes for a fleet.

3.1.2. Initialization

Beyond random initialization, this study proposes an improved initialization method based on the sweep algorithm. The sweep algorithm has been widely used for dispatching vehicles [27]. It assigns disjoint sets of customers to vehicles according to the polar-coordinate angle, and the routes are subsequently improved by the Lin-Kernighan algorithm [41], yielding a near-optimal solution to the single-depot vehicle-dispatch problem. Given that the number of pickup nodes is variable due to their selectability in the MVSPDP, we perform such an assignment on only the *delivery* nodes. In addition, we estimate the delivery requests served for a vehicle. First, the number $\hat{s}$ of selected pickup nodes can be approximated as follows:

$$\hat{s} = \left(1 - \frac{\sum_{v_i \in V} d_i}{\sum_{v_j \in V^+} d_j}\right) \mid V^+ \mid, \tag{23}$$

where $\frac{\sum_{v_i \in V} d_i}{\sum_{v_j \in V^+} d_j}$ is the ratio of surplus commodities to the total supply. The number of required vehicles $\hat{m}$ can then be estimated by

$$\hat{m} = \left\lfloor \frac{(\hat{s} + |V^-|) \cdot \bar{c}}{R - \bar{c}} + 0.5 \right\rfloor, \tag{24}$$

where the numerator $(\hat{s} + |V^-|) \cdot \bar{c}$ provides an approximate travel distance for visiting the selected pickup nodes and all delivery nodes, and the denominator $(R - \bar{c})$ represents the maximal travel distance permitted for a vehicle, where $\bar{c}$ is subtracted because the route in the numerator omits one edge (i.e., the edge from the final node to the depot). Accordingly, the estimated number of delivery requests served by a vehicle can be determined using

$$\delta = \frac{\sum_{v_i \in V^-} |d_i|}{\hat{m}'}, \tag{25}$$

where $\hat{m}' \in [\min\{\hat{m}, m\}, m]$ is randomly chosen to diversify the initial state.

To generate an initial solution, we sort the vertices according to the polar-coordinate angle and randomly select one vertex as the starter. Following the sorting order, delivery nodes are sequentially assigned to a vehicle until the number of required commodities exceeds δ. The final node is then randomly determined to be included in the route or a new route served by the next vehicle. This assignment separates the sorted vertices into partitions. Accordingly, each vehicle begins at the first pickup node in its partition and adds subsequent nodes (pickup or delivery) in ascending order of polar-coordinate angle. The pickup nodes are stacked in the waiting list according to the sorting order. Given that a nonnegative vehicle load must be maintained, a pickup node is popped from the waiting list whenever a negative vehicle load occurs. Fig. 2 illustrates the proposed initialization, where v_4 in Fig. 2a is selected as the starter, producing sorted vertices as shown in Fig. 2b. Fig. 2c presents three partitions, namely $\{v_1, \ldots, v_5\}$, $\{v_6, \ldots, v_{10}\}$, and $\{v_{11}, \ldots, v_{16}\}$, in which v_5 and v_{11} are the final nodes assigned: v_5 is included in the first route, but v_{11} is allocated to the subsequent vehicle. Additionally, v_{12} is the first pickup node of the third partition, but its supply is insufficient before visiting v_{13}; therefore, we pop v_8 from the stack to keep the vehicle load nonnegative, causing crosses in the initial routes. Although it may bring about infeasible solutions for the travel distance and vehicle load, using δ and polar-coordinate sorting can effectively generate promising initial solutions.

3.1.3. Neighborhood functions

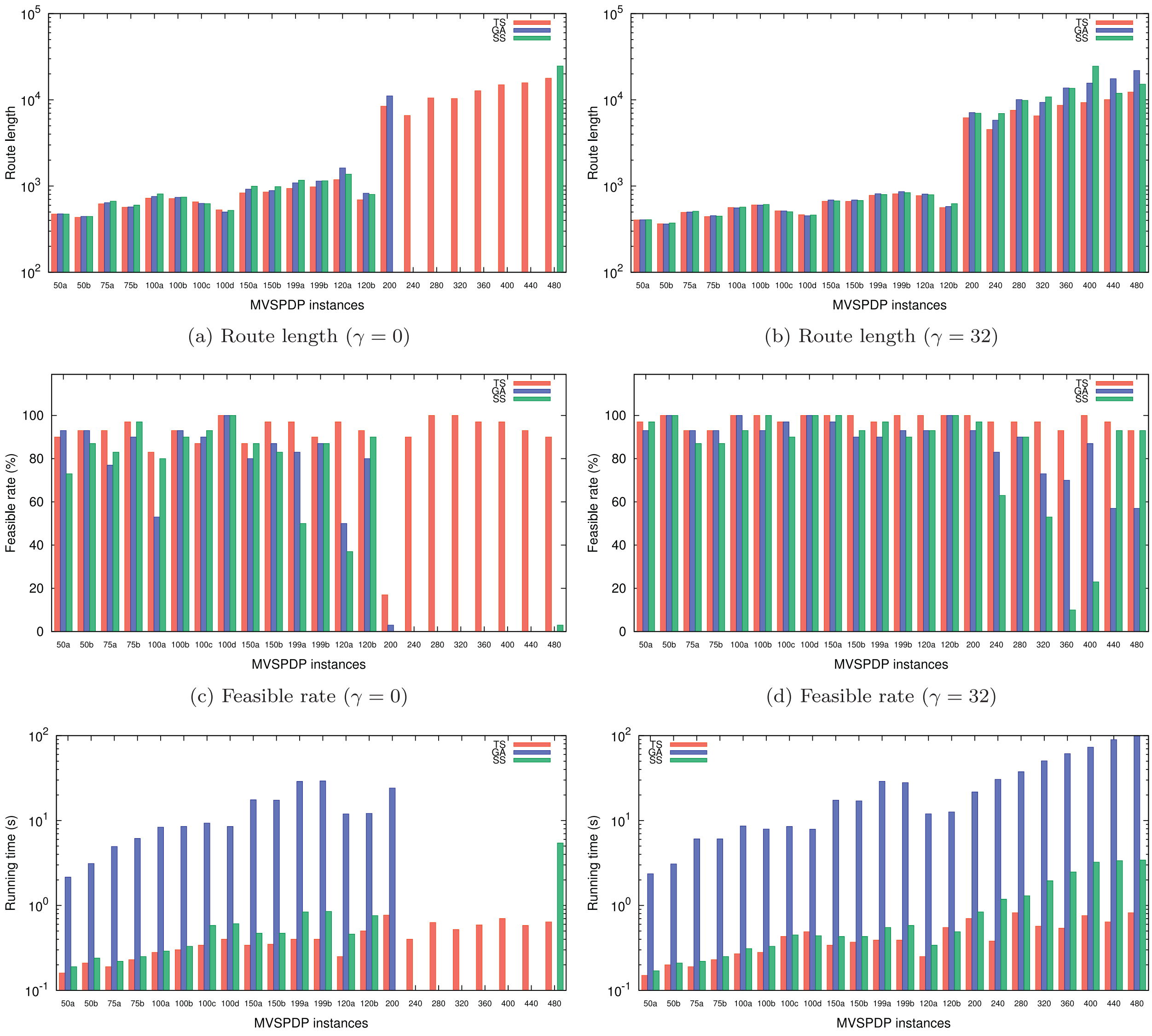
In this study, we devise four operators to construct the neighbors of the current solution. The four operators address different aspects of the MVSPDP: The relocation operator and the inversion operator account for inter-route and intra-route variations, respectively. The selection operator processes the selection of pickup nodes in the MVSPDP. Furthermore, the addition/deletion operator adjusts the number of vehicles used.

For generating the neighbors, the proposed TS randomly performs one of the four operators on the current solution and repeats this procedure until the predetermined neighborhood size is achieved. Two methods are proposed for determining the probability of an operator being selected: fixed probability and adaptive probability control. More details on the four operators are presented below.

Relocation. This operator produces an inter-route variation by moving a randomly chosen node to a previous or subsequent route. The destination is limited to keep the sequence of the polar angles of nodes as complete as possible. Fig. 3 illustrates an application of the relocation operator in which node v_9 is randomly chosen and relocated to the subsequent route.

Inversion. The inversion operator produces an intra-route variation that inverts a random partial route to alter the visiting order. An example of this operator is presented in Fig. 4. The chosen partial route $v_5 \to v_9 \to v_{12}$ is inverted into $v_{12} \to v_9 \to v_5$.

Selection. The selection operator addresses the key factor in the MVSPDP: the selection of pickup nodes. The operator is designed to randomly move a pickup node among the $m + 1$ sequences. The selection operator randomly chooses a pickup node from the final sequence and then inserts this node into one of the first m routes as a selected pickup node. In addition,



(a) Route length ($\gamma = 0$) (b) Route length ($\gamma = 32$)

(c) Feasible rate ($\gamma = 0$) (d) Feasible rate ($\gamma = 32$)

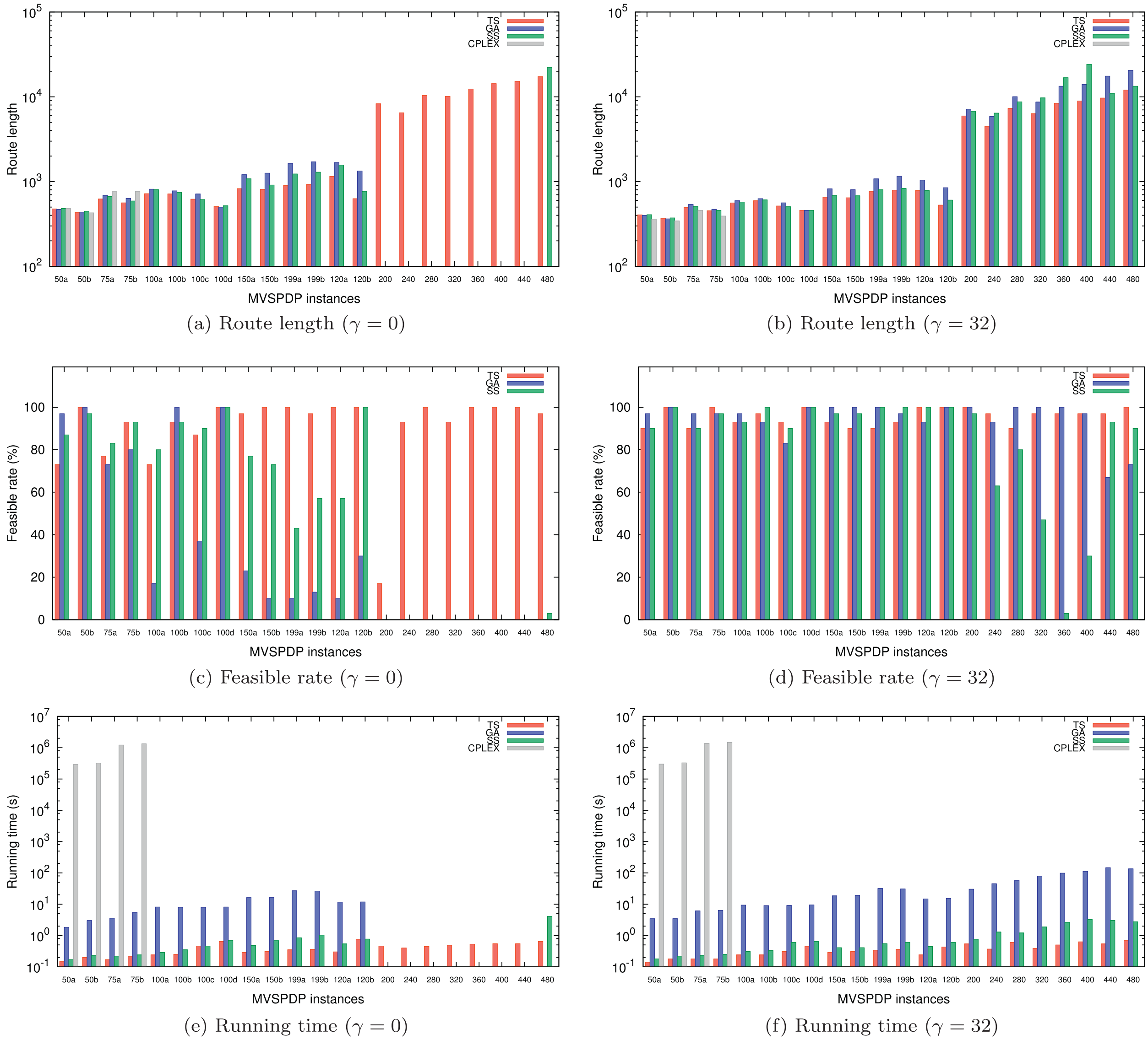
(e) Running time ($\gamma = 0$) (f) Running time ($\gamma = 32$)

Fig. 9. Comparison of average route length, feasible rate, and average running time for the TS, GA, and SS using the *fixed* probability of applying the four operators on the MVSPDP instances with $\gamma = 0$ (left) and 32 (right).

this operator may unselect a pickup node by moving it from the first m routes to the final sequence. Note that in the representation, the first m sequences indicate the routes for m vehicles while the last sequence consists of the unselected pickup nodes. As illustrated in Fig. 5, node v_1 is selected and inserted between v_2 and v_7; by contrast, pickup node v_{13} is unselected as it is moved to the final sequence.

Addition/Deletion. The number of vehicles used affects the routes and total distance traveled by all vehicles in the MVSPDP. This study develops the addition/deletion operator to adjust the number of vehicles used; whether vehicles are added or deleted is determined at uniform random. When adding a vehicle, the operator randomly selects a vehicle and then splits its route into two parts: the first part is the route for the selected vehicle, and the second part is separated as a route for a new vehicle. For example, the new route in Fig. 6a is separated from the second route.

When deleting a vehicle, the operator merges a randomly selected route with its adjacent route in the first part of the representation. However, the merger may be infeasible because of vehicle overload. To address this issue, we propose a heuristic to trim unnecessary pickup nodes. More specifically, this method iteratively removes the last visited pickup node by moving it to the sequence of unselection if this elimination can keep the number of remaining commodities nonnegative. In other words, the demand of removed pickup nodes must be less than or equal to $\sum_{v_i \in V} d_i y_{ik}$. Fig. 6b illustrates the deletion operation. Since the merged route has five remaining commodities, the deletion operator removes v_{13} with $d_{13} = 3$ to reduce



(a) Route length ($\gamma = 0$) (b) Route length ($\gamma = 32$)
(c) Feasible rate ($\gamma = 0$) (d) Feasible rate ($\gamma = 32$)
(e) Running time ($\gamma = 0$) (f) Running time ($\gamma = 32$)

Fig. 10. Comparison of average route length, feasible rate, and average running time for the TS, GA, and SS using the *adaptive* probability of applying the four operators, in comparison with CPLEX, on the MVSPDP instances with $\gamma = 0$ (left) and 32 (right).

the number of unnecessary pickup nodes. This procedure terminates at v_6 because $d_6 = 3$, which exceeds $\sum_{v_i \in V} d_i y_{ik} = 2$ at that time. The proposed method reduces the probability of overload and implicitly shortens the merged route.

3.1.4. Adaptive search phase

The proposed TS comprises two search phases: *normal* and *intensification* phases. Each phase manipulates the recency-based short-term memory to achieve a particular strategy. In the TS, the tabu list records only the edges added or removed by neighborhood functions instead of the entire solution. The list is used to forbid the change of these edges during tabu tenure to increase diversification. The tabu tenure is randomly initialized within $\{5, \ldots, 15\}$ in this study. Additionally, the intensification phase halves the tabu tenure for the TS to exploit the neighborhood of elite solutions.

The search phase is adaptively controlled according to variable ϕ. The TS conducts intensification only if ϕ is below the threshold θ; otherwise, it performs a normal search. The intensification phase empties tabu list and halves the tabu tenure, and then restarts the search with the best-so-far solution. The value of ϕ is adjusted according to counter r, which increases as the best solution improves, and decreases otherwise. Specifically, ϕ is varied by

$$\phi = \begin{cases} \phi \cdot \kappa & \text{if } r = -15 \text{ and } \phi > 0.2 \\ \phi / \kappa & \text{if } r = 20 \text{ and } \phi < 0.8 \\ \phi & \text{otherwise} \end{cases} \tag{26}$$

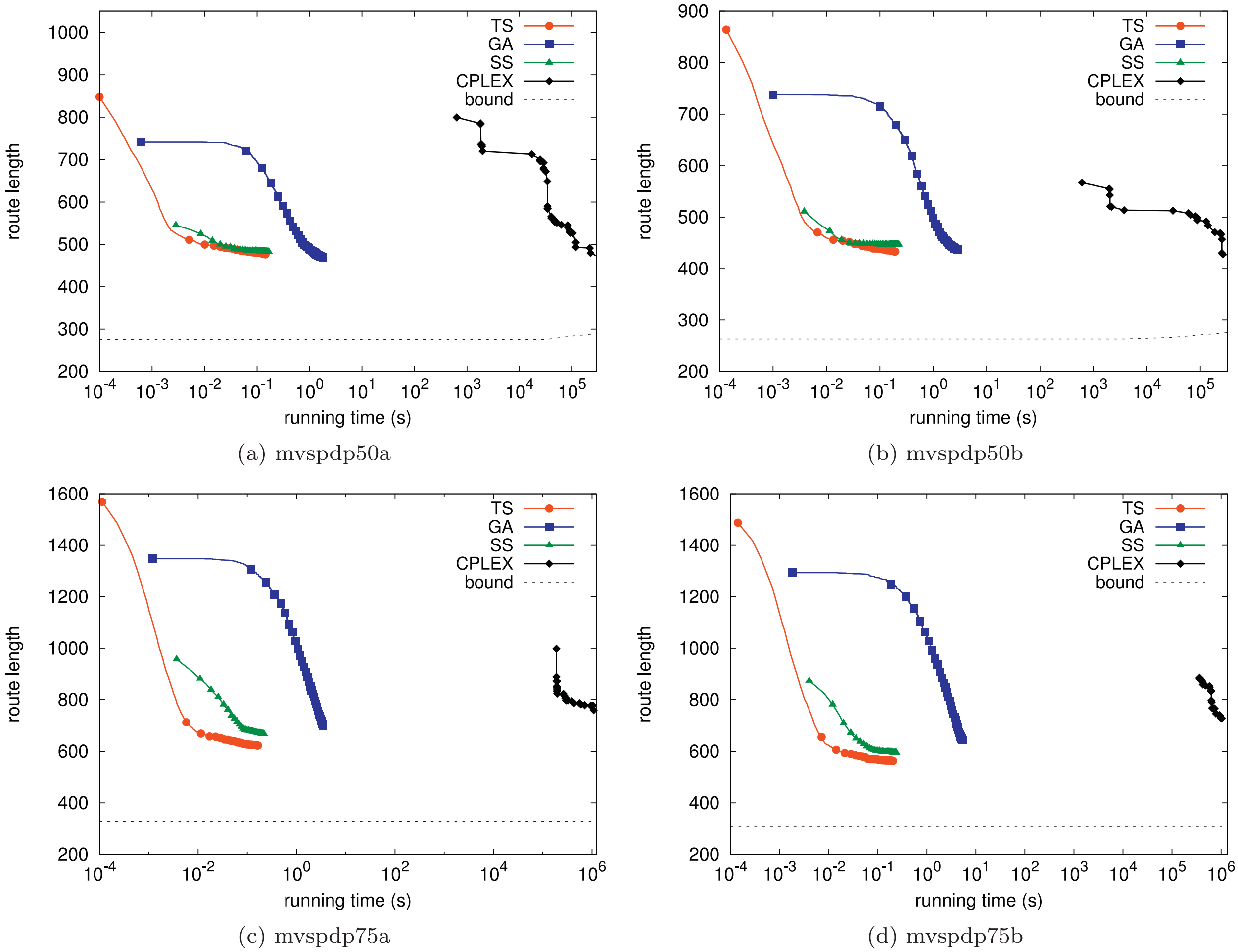


Fig. 11. Anytime behavior of the average route length over 30 runs of TS, GA, and SS using the adaptive probability, in comparison with CPLEX and its bound, on four MVSPDP instances with $\gamma = 0$.

As the search stagnates so long that r reaches -15, the value of ϕ decreases due to the positive coefficient $\kappa < 1$. If this situation consecutively occurs τ times, then the search is regarded as frozen, and thus we inactivate r and ϕ for t_f generations. In addition, the tabu tenure is doubled in subsequent iterations to encourage exploring the unvisited regions and escape from the local optima. In this study, the coefficients κ and ϕ are empirically set to 0.8 and 0.4, respectively; θ is fixed to 0.2; $\tau = \max\{2, \lfloor n/50 + 0.5 \rfloor\}$; and the frozen time t_f is 50 generations.

In summary, the adaptive search phase adjusts tabu tenure in response to the search progress for a balance between exploitation and exploration. Algorithm 1 shows the procedure of the proposed TS.

3.2. Genetic algorithm

Genetic algorithm is well known for its effectiveness in global optimization. Generally, GA uses a set (population) of candidate solutions (individuals or chromosomes) that can exchange information with each other by using genetic operators to explore the solution space. The operations in GA include selection, crossover, and mutation. The selection operator chooses individuals from the population to generate new candidate solutions (offspring) through crossover and mutation. The crossover operator exchanges and recombines the genetic information of selected individuals to produce offspring, and the mutation slightly changes offspring to provide diversity. In GA, the 'survival of the fittest' principle is applied in selecting individuals with highest fitness values to survive into the next generation.

This study involves developing a GA for the MVSPDP. The designs focus on representation, fitness function, and genetic operators. The details on the proposed GA are described in the subsequent sections.

3.2.1. Representation and fitness function

Individuals in the GA are represented using the representation proposed for the TS. Restated, an individual is composed of two parts: the first part indicates the use of vehicles, and the second part represents their respective routes, i.e., visit-

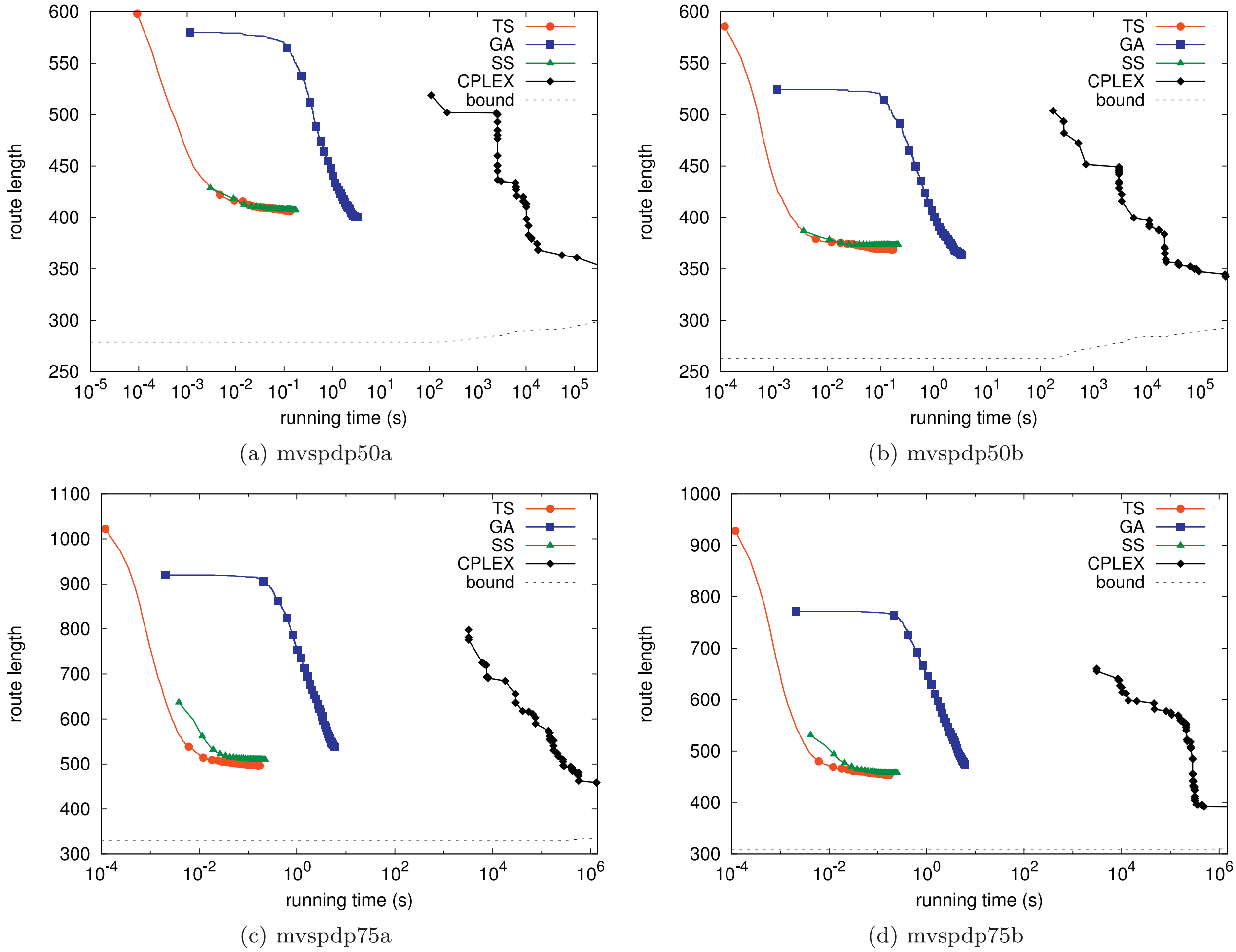


Fig. 12. Anytime behavior of the average route length over 30 runs of TS, GA, and SS using the adaptive probability, in comparison with CPLEX and its bound, on four MVSPDP instances with $\gamma = 32$.

ing orders of nodes (cf. Fig. 1). The half-half strategy is adopted for population initialization: the TS initialization operator generates half of the initial population; the other half is generated at random to provide diversity. The fitness function uses (19), which accounts for the route length and the constraints on vehicle capacity and travel distance in the MVSPDP.

3.2.2. Genetic operators

The genetic operators in the GA include selection, crossover, and mutation. The selection operators, including parent selection and survival selection, are associated with the fitness values of individuals. This study adopts 2-tournament parent selection because of its recognized performance [2,22]. The 2-tournament selection randomly picks two individuals from the population and selects the better individual as a parent. Performing this selection twice yields a pair of parents to be used in subsequent reproduction (i.e., crossover and mutation).

The crossover operator generates offspring by exchanging the information of two selected candidate solutions. In the light of the representation of individuals, we adopt the whole arithmetic crossover [45] with $\alpha = 0.8$ on the integer string of the first part and conduct the order crossover [21] on the merged routes in the second part. Note that the offspring inherit the selection of pickup nodes from the second part of their parents. However, the sum of resultant integer strings may exceed or fall short of the total number of customers. To address this issue, the proposed crossover operator entails selecting a random gene and reducing or increasing its value by 1 until reaching the requirement. The merged route is then divided among vehicles according to the designated numbers in the integer string. Fig. 7 demonstrates the proposed crossover operator with which the offspring obtain new routes as well as a diverse selection of pickup nodes.

Moreover, this study utilizes the four neighborhood functions of TS as the mutation operators for the GA. The relocation and inversion operators are used to produce inter-route and intra-route variations, respectively. The selection operator alters the selection of pickup nodes in the MVSPDP. The addition/deletion operator can increase or decrease the number of vehicles

Algorithm 1: Tabu search for the MVSPDP.

```
Input: MVSPDP instance
Output: solution x*

initialize(x) ;                                          // modified sweep algorithm
x* ← x ;                                                 // x*:best-so-far solution
tenure ← rand(5, 15);
phase ← Normal;
r, c, t ← 0, φ ← 0.4;
repeat
    N ← neighborhood(x);
    repeat
        x′ ← best of N;
        N ← N\{x′};
    until f(x′) < f(x*) or (xΔx′) ∉ TabuList;
    adjust tenure according to phase;
    update TabuList with xΔx′ and tenure;                // Δ:symmetric difference
    x ← x′;
    if f(x) < f(x*) then x* ← x, r ← r + 1;
    else r ← r − 1;
    adjust φ according to r;                             // Eq. (26)
    if c = τ then
        t ← t_f;
        tenure ← 2 · tenure;
    t ← t − 1;
    if φ ≤ θ and t < 0 then                              // intensification phase
        if phase = Normal then
            x ← x*;
            TabuList ← ∅;
        phase ← Intensification;
        c ← c + 1;
    else                                                 // normal phase
        phase ← Normal;
        c ← 0;
        if t = 0 then r ← 0, φ ← 0.4;
until terminated;
```

used. Each operator has a specific mutation rate, which is predetermined or adapted during the run. Section 4 examines the effects of these two strategies.

The selection-crossover-mutation process is repeated until the offspring population is generated. The survival selection then picks out the fitter individuals for the subsequent generation. This study uses $(\mu + \lambda)$ survival selection, in which the fittest μ individuals are selected from the union of μ parents and λ offspring to survive into the next generation.

3.3. Scatter search

Scatter search manipulates the population of solutions and uses a combination method to create new solutions [30]. The key components of SS are listed as follows.

- Diversification generation: A collection of diverse solutions is created using arbitrary or seed solutions.
- Improvement: A solution is transformed into one or more enhanced solutions.
- Reference set update: This procedure develops and maintains a reference set containing numerous best solutions found.
- Subset generation: A subset of solutions is generated from the reference set as the basis for solution combination.
- Solution combination: The subset of generated solutions is combined into one or more solutions.

Table 7
The average route length (avg.) and feasible rate (feas.) for the GA using the fixed and adaptive probability of applying the four operators. Column "time inc." lists the increment (or decrement) in the running time of GA-adaptive over GA-fixed. Boldface indicates the higher feasible rate and significantly shorter average route length according to a one-tailed t-test with confidence level $\alpha = 0.05$. The dash indicates that no feasible solution is found in 30 runs.

Instance	$\gamma = 0$					$\gamma = 32$				
	GA-fixed		GA-adaptive			GA-fixed		GA-adaptive		
	Ave.	Feas. (%)	Ave.	Feas. (%)	Time inc. (%)	Ave.	Feas. (%)	Ave.	Feas. (%)	Time inc. (%)
mvspdp50a	475	93	468	**97**	−14.42	406	93	399	**97**	45.34
mvspdp50b	444	93	437	**100**	−4.17	363	**100**	363	**100**	12.01
mvspdp75a	**642**	**77**	688	73	−27.88	**500**	93	536	**97**	0.66
mvspdp75b	**572**	**90**	635	80	−10.57	**453**	93	471	**97**	4.28
mvspdp100a	**757**	**53**	815	17	−2.40	**558**	**100**	596	97	8.71
mvspdp100b	**739**	93	778	**100**	−6.33	**602**	**93**	629	**93**	13.98
mvspdp100c	**629**	**90**	714	37	−14.45	**514**	**97**	561	83	6.93
mvspdp100d	500	**100**	499	**100**	−4.11	452	**100**	456	**100**	20.25
mvspdp150a	**921**	**80**	1207	23	−7.54	**690**	97	824	**100**	6.23
mvspdp150b	**886**	**87**	1258	10	−4.94	**690**	90	804	**100**	11.87
mvspdp199a	**1090**	**83**	1636	10	−6.55	**814**	90	1078	**100**	9.74
mvspdp199b	**1142**	**87**	1713	13	−9.86	**862**	93	1157	**97**	10.38
mvspdp120a	1623	**50**	1680	10	−2.67	**807**	**93**	1041	**93**	23.50
mvspdp120b	**825**	**80**	1336	30	−2.31	**579**	**100**	847	**100**	22.19
mvspdp200	**11,109**	**3**	–	0		7112	93	7164	**100**	39.83
mvspdp240	–	0	–	0		5804	83	5849	**93**	48.21
mvspdp280	–	0	–	0		10,074	90	10,034	**100**	54.43
mvspdp320	–	0	–	0		9354	73	**8674**	**100**	56.51
mvspdp360	–	0	–	0		13,725	70	**13,333**	**100**	59.92
mvspdp400	–	0	–	0		15,611	87	**14,025**	**97**	54.40
mvspdp440	–	0	–	0		17,579	57	17,557	**67**	64.32
mvspdp480	–	0	–	0		21,865	57	**20,471**	**73**	36.12

3.3.1. Diversification generation and improvement

The representation and evaluation function of SS follow those of the proposed TS and GA. In addition, the diversification generation method uses the half-half strategy, in which both the modified sweep algorithm and random initialization are adopted, as the proposed GA does. The improvement method is based on hill climbing (HC), for which the neighborhood function employs the four operators of TS presented in Section 3.1.3.

3.3.2. Solution combination

The combination method exchanges the information collected during the search. The proposed combination method is used to create two solutions from a pair of routes in the reference set to retain their characteristics. Fig. 8 illustrates the procedure for creating a solution. The solution c_1 inherits the first segment (i.e., the route of the first vehicle) from solution a and sequentially fills the remaining genes without duplicates from solution b. The split procedure is then applied to cut the solution and determine the number of nodes that each vehicle serves. The other solution is generated similarly but in an opposite direction, that is, by copying the first segment of b and filling the remainder from a.

In this study, the split procedure is based on the saving heuristic [17]. The cost (distance) before linking two nodes v_i and v_j is $c_{0i} + c_{i0} + c_{0j} + c_{j0}$, whereas the cost after linking them is $c_{0i} + c_{ij} + c_{j0}$. Thus, the benefit s from aggregation of v_i and v_j is

$$s_{ij} = c_{i0} + c_{0j} - c_{ij}. \tag{27}$$

The edges are linked according to the amount of saved cost. However, the constraints on travel distance (6) and vehicle capacity (9) may cause an individual to be cut into too many segments when using the split procedure; consequently, the generated solutions become too similar. To address this problem, the split procedure uses two alternative constraints,

$$2\left\lfloor \sum_{v_i \in V^-} \frac{d_i}{|V^-|} \right\rfloor \le z_{Sk} \sum_{v_i \in S} d_i y_{ik} \le Q \tag{28}$$

$$\sum_{v_i, v_j \in V} c_{ij} x_{ijk} \le R - \frac{n \cdot \bar{c}}{50} \tag{29}$$

to relax the lower bound of the vehicle load and tighten the upper bound of the travel distance, respectively.

3.3.3. Reference set

The reference set is updated through an operation similar to the selection operation of GA, which chooses solutions to generate new solutions. In SS, the reference set *Ref* is updated half from the original solutions and half from the newly

Table 8
The average route length (avg.) and feasible rate (feas.) for the TS using the fixed and adaptive probability of applying the four operators. Column "time inc." lists the increment (or decrement) in the running time of TS-adaptive over TS-fixed. Boldface indicates the higher feasible rate and significantly shorter average route length according to a one-tailed t-test with confidence level $\alpha = 0.05$. The dash indicates that no feasible solution is found in 30 runs.

Instance	$\gamma = 0$					$\gamma = 32$				
	TS-fixed		TS-adaptive			TS-fixed		TS-adaptive		
	Ave.	Feas. (%)	Ave.	Feas. (%)	Time inc. (%)	Ave.	Feas. (%)	Ave.	Feas. (%)	Time inc. (%)
mvspdp50a	473	**90**	477	73	−6.25	404	**97**	405	90	−6.67
mvspdp50b	434	93	433	**100**	−4.76	365	**100**	369	**100**	−10.00
mvspdp75a	622	**93**	623	77	−10.53	496	**93**	496	90	−5.26
mvspdp75b	568	**97**	562	93	−8.70	**442**	93	453	**100**	−21.74
mvspdp100a	723	**83**	720	73	−14.29	564	**100**	560	93	−11.11
mvspdp100b	716	**93**	715	**93**	−16.67	603	**100**	**595**	97	−14.29
mvspdp100c	656	**87**	**621**	**87**	35.29	515	**97**	515	93	−27.91
mvspdp100d	529	**100**	**508**	**100**	62.50	464	**100**	459	**100**	−10.20
mvspdp150a	833	87	827	**97**	−14.71	666	**100**	**654**	93	−14.71
mvspdp150b	853	97	**808**	**100**	−11.43	664	**100**	**645**	90	−16.22
mvspdp199a	938	97	**898**	**100**	−12.50	781	**97**	**761**	90	−12.82
mvspdp199b	978	90	**927**	**97**	−10.00	812	**100**	**793**	93	−7.69
mvspdp120a	1186	97	1152	**100**	20.00	775	**100**	785	**100**	−4.00
mvspdp120b	693	93	**629**	**100**	52.00	563	**100**	**528**	**100**	−21.82
mvspdp200	8440	**17**	8288	**17**	−40.26	6190	**100**	**5937**	**100**	−22.86
mvspdp240	6584	90	6490	**93**	0.00	4529	**97**	4481	**97**	−2.63
mvspdp280	10,518	**100**	10,367	**100**	−28.57	7560	**97**	**7297**	90	−26.83
mvspdp320	10,359	**100**	**10,112**	93	−5.77	6512	**97**	**6341**	**97**	−31.58
mvspdp360	12,745	97	**12,373**	**100**	−10.17	8643	93	**8391**	**97**	−7.41
mvspdp400	14,954	97	**14,296**	**100**	−21.43	9299	**100**	**8938**	97	−17.11
mvspdp440	15,751	93	**15,228**	**100**	−5.17	10,066	**97**	**9677**	**97**	−15.63
mvspdp480	17,798	90	**17,357**	**97**	1.56	12,320	93	**12,017**	**100**	−14.63

generated solutions. Specifically, from the original solutions, the b_1 best and $\frac{|Ref|}{2} - b_1$ randomly-selected solutions are included in the reference set. Similarly, the remaining half of the reference set is composed of the b_2 best and $\frac{|Ref|}{2} - b_2$ randomly-selected solutions generated by the combination method.

4. Experimental results

This study conducts a series of experiments to evaluate the proposed TS, GA, and SS for the MVSPDP. The test instances are modified from the Capacitated VRP Instances [11], ranging from 50 to 480 customer nodes. In modifying these instances for the MVSPDP, we change the demands of some random nodes to be negative, subject to the total demand being non-negative. Table 1 lists the information of instances used in the experiments, where the total number of customers is indicated in the instance name. In addition, we introduce parameter γ as a gain in supply for each pickup node, i.e., $d_i' = d_i + \gamma$ for all $v_i \in V^+$, to investigate the influence of the selectability of pickup nodes, where the vehicle capacity is increased by 2γ as well. The test instances can be downloaded via http://cilab.cs.ccu.edu.tw/MVSPDP.zip. Table 2 lists the parameter setting for TS, GA, and SS in the experiments. Each algorithm is tested with 30 independent runs, considering its stochastic nature. The experiments are performed on Windows 7 platform and Intel i7-920 machines.

First, we investigate the performance of each algorithm using fixed and adaptive control strategies for the probability of executing four operations, namely relocation, inversion, selection, and addition/deletion. In the experiments of fixed probability, the neighborhood function of TS and SS randomly conducts one of the four proposed operations with equal probability, whereas GA uses commonly suggested mutation rates for the four operators and enables more than one operator to be used in a mutation operation. Tables 3 and 4 list the results of TS, GA, and SS on the MVSPDP instances with $\gamma = \{0, 32\}$, where the dash indicates that no feasible solution is found in 30 runs. The feasible rate represents the percentage of runs achieving feasible solutions among the 30 runs. Boldface on the feasible rate marks the best result for the instance, whilst boldface on the average route length denotes that the algorithm significantly outperforms the second best method according to a one-tailed t-test with confidence level $\alpha = 0.05$, and multiple boldfaced results infer a nonsignificant difference between them. Fig. 9 further compares the results of the three metaheuristic algorithms. The experimental results show that, as the number of nodes increases, TS outperforms both GA and SS in the average route length and feasible rate. Precisely, TS excels with statistical significance in 17 and 11 out of 22 instances for $\gamma = 0$ and 32, respectively. Compared with TS, GA and SS hardly yield feasible solutions for the MVSPDP instances with > 200 nodes and $\gamma = 0$. Furthermore, TS uses lowest computation time, whereas GA requires more than 10 times computation time of TS and SS. These outcomes indicate the effectiveness and efficiency of the proposed TS for resolving the MVSPDP.

Second, this study proposes adaptive control over the probability of performing each operator. For TS and SS, this probability is determined according to the proportion of times that a particular operator is selected as the neighborhood function

Table 9
The route length (avg.) and feasible rate (feas.) for the SS using the fixed and adaptive probability of applying the four operators. Column "time inc." lists the increment (or decrement) in the running time of SS-adaptive over SS-fixed. Boldface indicates the higher feasible rate and significantly shorter average route length according to a one-tailed t-test with confidence level $\alpha = 0.05$. The dash indicates that no feasible solution is found in 30 runs.

Instance	$\gamma = 0$					$\gamma = 32$				
	SS-fixed		SS-adaptive			SS-fixed		SS-adaptive		
	Ave.	Feas. (%)	Ave.	Feas. (%)	Time inc. (%)	Ave.	Feas. (%)	Ave.	Feas. (%)	Time inc. (%)
mvspdp50a	**473**	73	482	**87**	−10.53	407	**97**	407	90	5.88
mvspdp50b	444	87	447	**97**	−4.17	373	**100**	373	**100**	4.76
mvspdp75a	668	**83**	667	**83**	0.00	510	87	509	**90**	4.55
mvspdp75b	601	**97**	590	93	−4.00	448	87	458	**97**	0.00
mvspdp100a	810	**80**	806	**80**	0.00	570	**93**	574	**93**	0.00
mvspdp100b	745	90	747	**93**	6.06	611	**100**	609	**100**	0.00
mvspdp100c	625	**93**	614	90	−20.69	503	**90**	505	**90**	35.56
mvspdp100d	522	**100**	519	**100**	14.75	461	**100**	458	**100**	45.45
mvspdp150a	**995**	**87**	1080	77	2.13	673	**100**	685	97	−4.65
mvspdp150b	985	**83**	911	73	44.68	679	93	679	**97**	−4.65
mvspdp199a	1169	**50**	1232	43	1.19	797	97	801	**100**	0.00
mvspdp199b	**1149**	**87**	1291	57	21.18	837	90	829	**100**	5.17
mvspdp120a	1372	37	1565	**57**	17.39	793	93	784	**100**	32.35
mvspdp120b	800	90	767	**100**	1.32	625	**100**	605	**100**	24.49
mvspdp200	–	0	–	0		6982	**97**	**6758**	**97**	−9.52
mvspdp240	–	0	–	0		6943	**63**	**6433**	**63**	9.32
mvspdp280	–	0	–	0		9841	**90**	**8730**	80	−6.92
mvspdp320	–	0	–	0		10,825	**53**	**9713**	47	−3.59
mvspdp360	–	0	–	0		13,602	**10**	16,780	3	6.45
mvspdp400	–	0	–	0		24,569	23	24,180	**30**	−0.31
mvspdp440	–	0	–	0		11,913	**93**	**10,976**	**93**	−10.09
mvspdp480	24,690	**3**	**22,260**	**3**	−25.05	15,179	**93**	**13,348**	90	−19.59

during the search process. The lower bound of the probability is set at 0.05 to prevent stagnation, and the probability for each operator is 0.25 during the frozen time in TS. As for GA, after an offspring is generated, we adaptively adjust the probability by multiplying the mutation rate by the ratio of its fitness to the less fit parent. Additionally, the mutation rates of relocation and inversion operators are limited within [0.3, 0.9], and the rates for the selection and addition/deletion operators are $[1/2|V^+|, 2/|V^+|]$ and $[1/2m, 2/m]$, respectively. Tables 5 and 6 summarize the performance of TS, GA, and SS with the adaptive probability of applying each operator on the MVSPDP instances with $\gamma = \{0, 32\}$.

Fig. 10 compares the route length, feasible rate, and average running time of the three metaheuristic algorithms. Similar to the tendency observed regarding fixed probability, TS significantly outperforms GA and SS in 14 and 17 out of 22 instances for $\gamma = 0$ and 32, respectively. Moreover, the experimental results indicate the considerable advantages of TS over GA and SS in running time and feasible rate for solving the MVSPDP. Based on the superior performance, the TS with adaptive probability is a favorable method for the MVSPDP. Considering the search behavior of metaheuristic algorithms, the local search methods like TS are good at locating the optima in a region, whereas the global search methods such as GA excel in identifying the promising regions for global optima. In this regard, the preferable performance of TS implies the advantage of local search methods to solve the MVSPDP. In addition, the comprehensive survey of Cordeau et al. [19] indicated that a considerable number of tabu search algorithms have been proposed for the VRP, while the use of adaptive memory procedure initiated the study of population search heuristics for the VRP [10]. Baker and Ayechew [3] stated that, although GA performs well, it is inferior to tabu search in terms of solution quality on the VRP. As the proposed MVSPDP is a variant of the VRP, these empirical validations are referable for the MVSPDP; in fact, our experimental results are consistent with these findings.

Next, we compare the anytime behavior of the proposed TS, GA, and SS in terms of route length against time. The comparison further includes the mixed integer programming solver IBM ILOG CPLEX, which presents a performance baseline for the MVSPDP. A termination criterion of six million nodes traversed is set for the CPLEX due to long running time. Figs. 11 and 12 depict the progress of route length against running time for the test algorithms on four MVSPDP instances with $\gamma = \{0, 32\}$. The results show that TS achieves the fastest convergence, and SS converges faster than GA does. In addition, the three proposed methods, i.e., TS, SS, and GA, are much more efficient than CPLEX. The figures show the closeness of the obtained route lengths to the gradually increasing lower bound of CPLEX. However, the long running time of CPLEX ($>$ 400 h) hinders the experiments from getting the optimal solution and lower bound. Tables 5 and 6 compare the route length obtained from the four test algorithms. According to the results, CPLEX can mostly gain shorter routes than the three proposed algorithms on the small instances; nonetheless, its running time is at least 10^6 times longer than TS and SS, and 10^5 times longer than GA. By contrast, the three proposed algorithms, especially the TS, can yield competitive results in reasonable time.

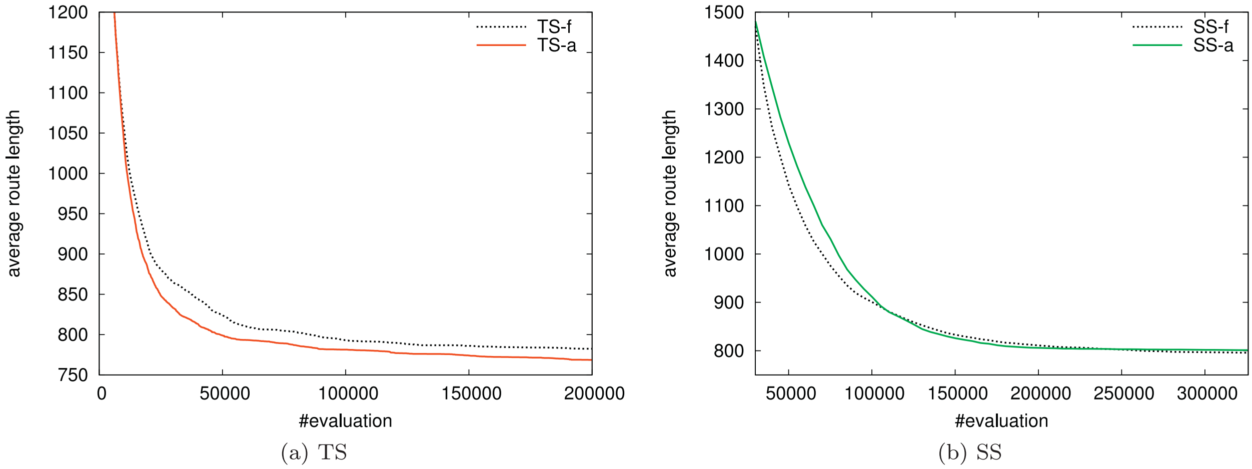


Fig. 13. Anytime behavior of the average route length over 30 runs of TS and SS using the fixed probability (f) and adaptive probability (a) of applying the four operators on mvspdp199a with $\gamma = 32$.

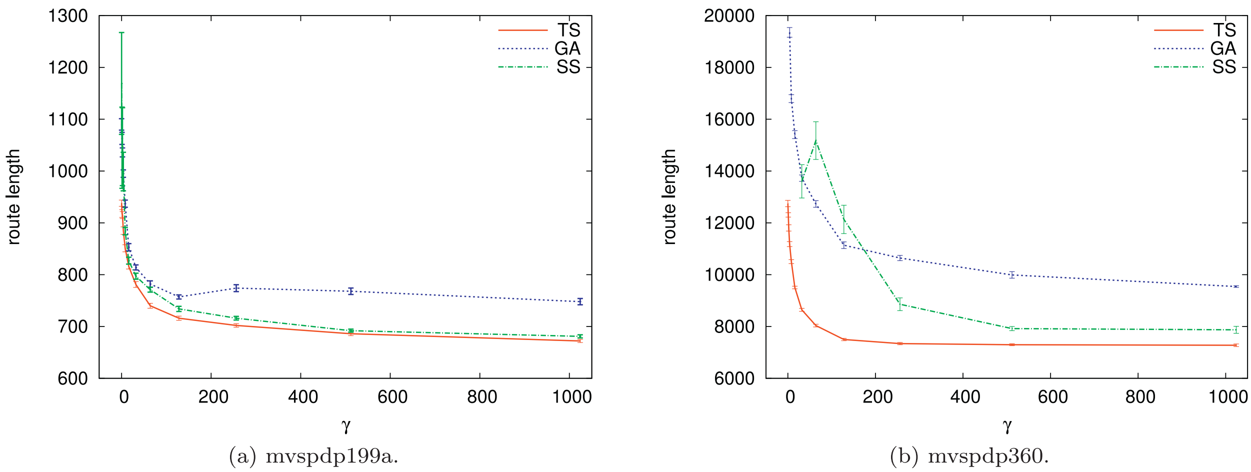
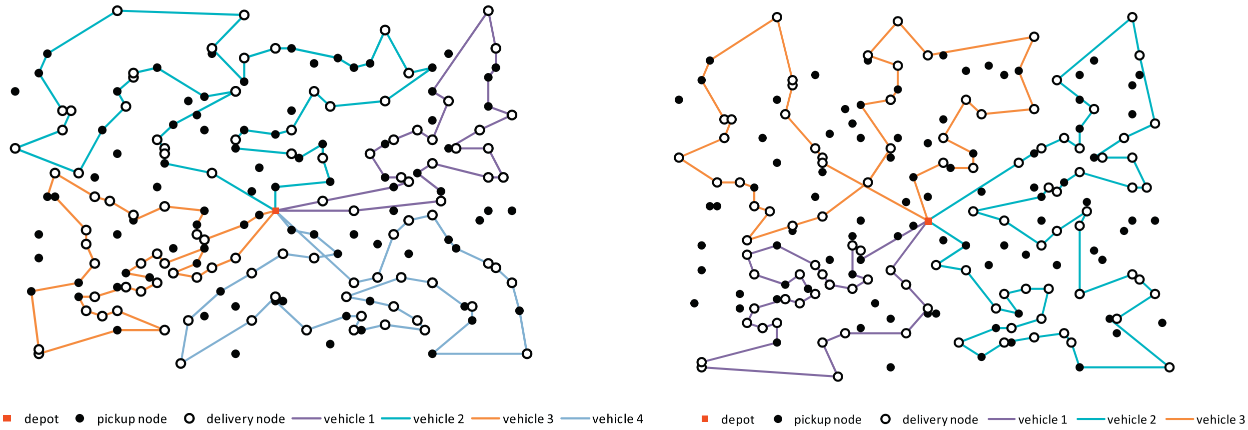


Fig. 14. The average route lengths obtained from TS, GA, and SS using the adaptive probability of applying the four operators at different gain values γ.

Tables 7–9 compare the results of fixed and adaptive probability to investigate their effects on the three algorithms. According to Table 7, the GA with fixed probability surpasses that with adaptive probability in route length, but is inferior in feasible rate. By contrast, Tables 8 and 9 show that adaptively tuning the probability of executing operations for TS and SS can significantly shorten the route length on instances involving substantial numbers of nodes and high gains. Notably, TS with adaptive probability is computationally less time-consuming than that with fixed probability in most cases. Both TS and SS with adaptive probability achieve similar feasible rates to their corresponding versions with fixed probability. In addition, the anytime behavior of TS and SS, shown in Fig. 13, demonstrates that adaptive probability effectively improves the convergence speed of TS. These results validate the benefits of employing adaptive probability in TS and SS, in which the adaptive control mechanism facilitates selecting promising operators in the course of search process.

Finally, this study experiments with different gain values to examine the influence of the selectability of pickup nodes. Fig. 14 presents only the results on mvspdp199a and mvspdp360 due to space limitation. The results indicate that GA and SS infrequently find feasible solutions on mvspdp360 when γ is low, whereas the proposed TS achieves the shortest routes and yields the lowest standard deviation among the test algorithms. The route length generally decreases as γ increases, confirming that the flexibility in selecting pickup nodes intensifies when the commodities outnumber those required. The selectability of pickup nodes in the MVSPDP expands the search space of VRP. In other words, the sparsity of feasible solutions and the low gain γ raise the difficulty of searching for the optima; however, as γ increases, shorter routes could be achieved by selecting the required pickup nodes.

Fig. 15 illustrates the routes obtained from TS on mvspdp199a with $\gamma = \{0, 32\}$. The resultant routes validate the benefits of the MVSPDP: the route length and the number of selected pickup nodes decrease as γ increases. Additionally, the selectability reduces the number of vehicles used (Fig. 15b). These results indicate that relaxing the constraint on visiting all nodes provides economic solutions for real-world applications that focus on supplying all of the customers demanding



(a) $\gamma = 0$, route length: 820.63. (b) $\gamma = 32$, route length: 714.44

Fig. 15. The route obtained from TS on mvspdp199a with $\gamma = \{0, 32\}$. The square denotes the depot, solid circles denote pickup nodes, and hollow circles denote delivery nodes.

commodities. The solutions may contain "crosses" in routes, which seem to be undesirable in the sense of shortest path. Nevertheless, the crosses are necessary for the optimal solutions in the SPDP and the MVSPDP because of the constraints on vehicle capacity and nonnegative vehicle loads.

5. Conclusions

This study presents the multi-vehicle selective pickup and delivery problem (MVSPDP), which aims for the shortest routes for a fleet of vehicles to collect and supply commodities in accordance with constraints regarding vehicle capacity and travel distance. The problem formulation relaxes the requirement for visiting all nodes and uses multiple vehicles. The MVSPDP is pertinent to real-world logistic applications that focus on supplying the demands of all customers from a certain number of providers by using multiple vehicles.

To resolve the MVSPDP, this study proposed three metaheuristic algorithms: TS, GA, and SS. The fixed-length representation enables indicating the varying number of vehicles and the selection of pickup nodes. Four neighborhood functions are introduced to TS, each of which focuses on one critical aspect of solving the MVSPDP. In addition, these functions are used as mutation operators in GA and as improvement methods for SS. Moreover, we proposed an initialization method based on the sweep algorithm for the MVSPDP.

A series of experiments is conducted to examine the proposed TS, GA, and SS. The experimental results indicate the utility of the four operators in selecting pickup nodes and arranging the visiting order for the MVSPDP, which leads to shorter routes than does the PDP. In addition, the results validate that the three metaheuristic algorithms can effectively solve the MVSPDP; in particular, TS outperforms both GA and SS in route length and feasible rate on most of the instances at various gain values. Moreover, TS requires less running time to achieve the solutions than the other two algorithms do. According to these outcomes, TS with the adaptive probability of applying the four operators is a preferable method for resolving the MVSPDP.

Future studies include some directions. First, this study focuses on design of TS, GA, and SS; nonetheless, other established algorithms, such as ant colony optimization (ACO) and particle swarm optimization (PSO), are also promising for the MVSPDP. Improving the selection of appropriate pickup nodes and arrangement of routes can also be considered. Second, the MVSPDP can be extended to various transportation scenarios and requirements, such as partial pickup and delivery, time windows, and multiple objectives. Partial pickup and delivery enables each customer to be served by multiple vehicles. The MVSPDP with time windows requires customers to be served at their preferred periods. The multi-objective MVSPDP involves additional objectives, such as the number of vehicles and waiting time. In addition, the costs of unselecting pickup customers and adding vehicles can be further taken into account. Third, formal analysis on the bound and behavior of metaheuristic algorithms for the MVSPDP is a challenging yet fruitful topic for future work.

Acknowledgment

The authors would like to thank the editor and reviewers for their valuable comments and suggestions. This work was supported by the National Science Council of Taiwan, under contract NSC100-2628-E-194-009-MY3.

References

[1] S. Anily, J. Bramel, Approximation algorithms for the capacitated traveling salesman problem with pickups and deliveries, Nav. Res. Logist. 46 (6) (1999) 654–670.
[2] T. Back, D.B. Fogel, Z. Michalewicz (Eds.), Handbook of Evolutionary Computation, IOP Publishing Ltd, 1997.
[3] B.M. Baker, M.A. Ayechew, A genetic algorithm for the vehicle routing problem, Comput. Oper. Res. 30 (5) (2003) 787–800.
[4] R. Baldacci, E. Bartolini, A. Mingozzi, An exact algorithm for the pickup and delivery problem with time windows, Oper. Res. 59 (2) (2011) 414–426.
[5] M. Battarra, G. Erdoğan, G. Laporte, D. Vigo, The traveling salesman problem with pickups, deliveries, and handling costs, Transp. Sci. 44 (3) (2010) 383–399.
[6] A.K. Beheshti, S.R. Hejazi, A novel hybrid column generation-metaheuristic approach for the vehicle routing problem with general soft time window, Inf. Sci. 316 (2015) 598–615.
[7] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, G. Laporte, Static pickup and delivery problems: A classification scheme and survey, TOP 15 (1) (2007) 1–31.
[8] C. Bordenave, M. Gendreau, G. Laporte, Heuristics for the mixed swapping problem, Comput. Oper. Res. 37 (1) (2010) 108–114.
[9] C. Bordenave, M. Gendreau, G. Laporte, A branch-and-cut algorithm for the preemptive swapping problem, Networks 59 (4) (2012) 387–399.
[10] B. Bozkaya, E. Erkut, G. Laporte, A tabu search heuristic and adaptive memory procedure for political districting, Eur. J. Oper. Res. 144 (1) (2003) 12–26.
[11] Capacitated VRP Instances, http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/, 2015.
[12] H.K. Chen, H.W. Chou, C.F. Hsueh, T.Y. Ho, The linehaul-feeder vehicle routing problem with virtual depots, IEEE Trans. Autom. Sci. Eng. 8 (4) (2011) 694–704.
[13] P. Chen, H.K. Huang, X.Y. Dong, An ant colony system based heuristic algorithm for the vehicle routing problem with simultaneous delivery and pickup, in: Proceedings of the 2007 IEEE Conference on Industrial Electronics and Applications (ICIEA 2007), 2007, pp. 136–141.
[14] X. Chen, Y.S. Ong, A conceptual modeling of meme complexes in stochastic search, IEEE Trans. Syst. Man Cybern. – Part C 42 (5) (2012) 612–625.
[15] R. Chiong, T. Weise, Z. Michalewicz, Variants of Evolutionary Algorithms for Real-World Applications, Springer, 2012.
[16] N. Christofides, A. Mingozzi, P. Toth, Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations, Math. Program. 20 (1) (1981) 255–282.
[17] G. Clarke, J.W. Wright, Scheduling of vehicles from a central depot to a number of delivery points, Oper. Res. 12 (4) (1964) 568–581.
[18] J.-F. Cordeau, G. Laporte, S. Ropke, Recent models and algorithms for one-to-one pickup and delivery problems, in: B. Golden, S. Raghavan, E. Wasil (Eds.), The Vehicle Routing Problem: Latest Advances and New Challenges, Operations Research/Computer Science Interfaces Series, vol.43, Springer US, 2008, pp. 327–357.
[19] J.-F. Cordeau, G. Laporte, M.W. Savelsbergh, D. Vigo, Vehicle routing, in: Handbooks in Operations Research and Management Science, vol.14, Elsevier, 2007, pp. 367–428.
[20] C.E. Cortés, M. Matamala, C. Contardo, The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method, Eur. J. Oper. Res. 200 (3) (2010) 711–724.
[21] L. Davis, Applying adaptive algorithms to epistatic domains., in: Proceedings of the International Joint Conference on Artificial Intelligence, vol.85, 1985, pp. 162–164.
[22] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Natural Computing, Springer-Verlag, 2003.
[23] B. Eksioglu, A.V. Vural, A. Reisman, The vehicle routing problem: A taxonomic review, Comput. Ind. Eng. 57 (4) (2009) 1472–1483.
[24] G. Erdoğan, J.-F. Cordeau, G. Laporte, A branch-and-cut algorithm for solving the non-preemptive capacitated swapping problem, Discrete Appl. Math. 158 (15) (2010) 1599–1614.
[25] R. Falcon, X. Li, A. Nayak, I. Stojmenovic, The one-commodity traveling salesman problem with selective pickup and delivery: An ant colony approach, in: Proceedings of the 2010 IEEE Congress on Evolutionary Computation (CEC 2010), 2010, pp. 1–8.
[26] T. Garaix, C. Artigues, D. Feillet, D. Josselin, Optimization of occupancy rate in dial-a-ride problems via linear fractional column generation, Comput. Oper. Res. 38 (10) (2011) 1435–1442.
[27] B.E. Gillett, L.R. Miller, A heuristic algorithm for the vehicle-dispatch problem, Oper. Res. 22 (2) (1974) 340–349.
[28] F. Glover, Heuristics for integer programming using surrogate constraints, Dec. Sci. 8 (1) (1977) 156–166.
[29] F. Glover, M. Laguna, Tabu Search, Kluwer Academic Publishers, 1997.
[30] F. Glover, M. Laguna, R. Martí, Fundamentals of scatter search and path relinking, Control cybern. 39 (3) (2000) 653–684.
[31] F.P. Goksal, I. Karaoglan, F. Altiparmak, A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery, Comput. Ind. Eng. 65 (1) (2013) 39–53.
[32] Y.-J. Gong, J. Zhang, O. Liu, R.-Z. Huang, H.-H. Chung, Y.-H. Shi, Optimizing the vehicle routing problem with time windows: A discrete particle swarm optimization approach, IEEE Trans. Syst. Man Cybern. Part C 42 (2) (2012) 254–267.
[33] I. Gribkovskaia, G. Laporte, One-to-many-to-one single vehicle pickup and delivery problems, in: B. Golden, S. Raghavan, E. Wasil (Eds.), The Vehicle Routing Problem: Latest Advances and New Challenges, Operations Research/Computer Science Interfaces, vol.43, Springer US, 2008, pp. 359–377.
[34] I. Gribkovskaia, G. Laporte, A. Shyshou, The single vehicle routing problem with deliveries and selective pickups, Comput. Oper. Res. 35 (9) (2008) 2908–2924.
[35] G. Gutiérrez-Jarpa, G. Desaulniers, G. Laporte, V. Marianov, A branch-and-price algorithm for the vehicle routing problem with deliveries, selective pickups and time windows, Eur. J. Oper. Res. 206 (2) (2010) 341–349.
[36] K. Haridass, J. Valenzuela, A.D. Yucekaya, T. McDonald, Scheduling a log transport system using simulated annealing, Inf. Sci. 264 (2014) 302–316.
[37] H. Hernández-Pérez, J.J. Salazar-González, The one-commodity pickup-and-delivery travelling salesman problem, in: M. Jünger, G. Reinelt, G. Rinaldi (Eds.), Combinatorial Optimization — Eureka, You Shrink!, Lecture Notes in Computer Science, vol.2570, Springer Berlin Heidelberg, 2003, pp. 89–104.
[38] A. Hoff, I. Gribkovskaia, G. Laporte, A. Løkketangen, Lasso solution strategies for the vehicle routing problem with pickups and deliveries, Eur. J. Oper. Res. 192 (3) (2009) 755–766.
[39] J. Holland, Adaptation in Natural and Artificial Systems., University of Michigan Press, 1975.
[40] H.C.W. Lau, T.M. Chan, W.T. Tsui, W.K. Pang, Application of genetic algorithms to solve the multidepot vehicle routing problem, IEEE Trans. Autom. Sci. Eng. 7 (2) (2010) 383–392.
[41] S. Lin, B.W. Kernighan, An effective heuristic algorithm for the traveling-salesman problem, Oper. Res. 21 (2) (1973) 498–516.
[42] J. Luo, X. Li, M.-R. Chen, H. Liu, A novel hybrid shuffled frog leaping algorithm for vehicle routing problem with time windows, Inf. Sci. 316 (2015) 266–292.
[43] G. Martinovic, I. Aleksi, A. Baumgartner, Single-commodity vehicle routing problem with pickup and delivery service, Math. Problems Eng. 2008 (2008), doi:10.1155/2008/697981. Article ID 697981.
[44] M. Mavrovouniotis, S. Yang, Ant algorithms with immigrants schemes for the dynamic vehicle routing problem, Inf. Sci. 294 (2015) 456–477.
[45] Z. Michalewicz, Genetic Algorithms+ Data Structures= Evolution Programs, Springer, 1996.
[46] Z. Michalewicz, D.B. Fogel, How to Solve It: Modern Heuristics, Springer, 2004.
[47] Y. Nagata, S. Kobayashi, Guided ejection search for the pickup and delivery problem with time windows, in: P. Cowling, P. Merz (Eds.), Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science, vol.6022, Springer Berlin Heidelberg, 2010, pp. 202–213.
[48] D. Naso, B. Turchiano, Multicriteria meta-heuristics for AGV dispatching control based on computational intelligence, IEEE Trans. Syst. Man Cybern. Part B 35 (2) (2005) 208–226.
[49] I.H. Osman, J.P. Kelly, Meta-heuristics: Theory and applications, Springer, 1996.
[50] S.N. Parragh, K.F. Doerner, R.F. Hartl, A survey on pickup and delivery problems—Part I: Transportation between customers and depot, Journal für Betriebswirtschaft 58 (2008) 21–51.

[51] S.N. Parragh, K.F. Doerner, R.F. Hartl, A survey on pickup and delivery problems—Part II: Transportation between pickup and delivery locations, Journal für Betriebswirtschaft 58 (2008) 81–117.
[52] S.N. Parragh, K.F. Doerner, R.F. Hartl, X. Gandibleux, A heuristic two-phase solution approach for the multi-objective dial-a-ride problem, Networks 54 (4) (2009) 227–242.
[53] H.L. Petersen, S. Ropke, The pickup and delivery problem with cross-docking opportunity, in: J.W. Böse, H. Hu, C. Jahn, X. Shi, R. Stahlbock, S. Voß (Eds.), Computational Logistics, Lecture Notes in Computer Science, vol.6971, Springer Berlin Heidelberg, 2011, pp. 101–113.
[54] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417.
[55] P. Sasikumar, G. Kannan, Issues in reverse supply chains, part II: Reverse distribution issues — an overview, Int. J. Sustainable Eng. 1 (4) (2008) 234–249.
[56] P. Sasikumar, G. Kannan, Issues in reverse supply chain, part III: Classification and simple analysis, Int. J. Sustainable Eng. 2 (1) (2009) 2–27.
[57] K.C. Tan, C.Y. Cheong, C.K. Goh, Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation, Eur. J. Oper. Res. 177 (2) (2007) 813–839.
[58] K.C. Tan, Y.H. Chew, L.H. Lee, A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing porblems, Eur. J. Oper. Res. 172 (3) (2006) 855–885.
[59] K.C. Tan, Y.H. Chew, L.H. Lee, A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows, Comput. Optim. Appl. 34 (1) (2006) 115–151.
[60] E. Teymourian, V. Kayvanfar, G.M. Komaki, M. Zandieh, Enhanced intelligent water drops and cuckoo search algorithms for solving the capacitated vehicle routing problem, Inf. Sci. (2015).
[61] C.K. Ting, X.L. Liao, The selective pickup and delivery problem: Formulation and a memetic algorithm, Int. J. Prod. Econ. 141 (1) (2013) 199–211.
[62] P. Toth, D. Vigo, VRP with backhauls, in: P. Toth, D. Vigo (Eds.), The vehicle routing problem, SIAM Monographs on Discrete Mathematics and Applications 9, Society for Industrial and Applied Mathematics, 2002, pp. 195–224.
[63] N. Velasco, P. Dejax, C. Guéret, C. Prins, A non-dominated sorting genetic algorithm for a bi-objective pick-up and delivery problem, Eng. Optim. 44 (3) (2012) 305–325.
[64] H.A. Waisanen, D. Shah, M.A. Dahleh, A dynamic pickup and delivery problem in mobile networks under information constraints, IEEE Trans. Automat. Control 53 (6) (2008) 1419–1433.
[65] X.P. Wang, H.Y. Cao, A dynamic vehicle routing problem with backhaul and time window, in: Proceedings of the 2008 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI 2008), 1, 2008, pp. 1256–1261.
[66] E.T. Yassen, M. Ayob, M.Z.A. Nazri, N.R. Sabar, Meta-harmony search algorithm for the vehicle routing problem with time windows, Inf. Sci. 325 (2015) 140–158.
[67] Z. Zhu, J. Xiao, S. He, Z. Ji, Y. Sun, A multi-objective memetic algorithm based on locality-sensitive hashing for one-to-many-to-one dynamic pick-up-and-delivery problem, Inf. Sci. 329 (2016) 73–89.