Int. J. Production Economics 141 (2013) 199–211

Contents lists available at SciVerse ScienceDirect

Int. J. Production Economics

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

The selective pickup and delivery problem: Formulation and a memetic algorithm

Chuan-Kang Ting*, Xin-Lan Liao

Department of Computer Science and Information Engineering, National Chung Cheng University, Chia-Yi 62102, Taiwan

ARTICLE INFO

Article history:
Received 16 November 2011
Accepted 7 June 2012
Available online 26 June 2012

Keywords:
Pickup and delivery
Logistics
Evolutionary computation
Memetic algorithm

ABSTRACT

The pickup and delivery problem addresses the real-world issues in logistic industry and establishes an important category of vehicle routing problems. The problem is to find the shortest route to collect and distribute commodities under the assumption that the total supply and the total demand are in equilibrium. This study presents a novel problem formulation, called the selective pickup and delivery problem (SPDP), by relaxing the constraint that all pickup nodes must be visited. Specifically, the SPDP aims to find the shortest route that can supply delivery nodes with required commodities from some pickup nodes. This problem can substantially reduce the transportation cost and fits real-world logistic scenarios. Furthermore, this study proves that the SPDP is NP-hard and proposes a memetic algorithm (MA) based on genetic algorithm and local search to resolve the problem. A novel representation of candidate solutions is designed for the selection of pickup nodes. The related operators are also devised for the MA; in particular, it adapts the 2-opt operator to the sub-routes of the SPDP for enhancement of visiting order. The experimental results on several SPDP instances validate that the proposed MA can significantly outperform genetic algorithm and tabu search in terms of solution quality and convergence speed. In addition, the reduced route lengths on the test instances and the real-world application to rental bikes distribution demonstrate the benefit of the SPDP in logistics.

1. Introduction

The pickup and delivery problem (PDP) arises in many real-world cases such as logistics and robotics. This problem consists of several nodes classified as *pickup* customers and *delivery* customers. The former supplies while the latter demands a number of commodities. The goal of the PDP is to find the shortest route such that the requirement of each customer can be satisfied. Solving this problem concerns vehicle routing and commodity distribution. The PDP has been proved to be NP-hard. Several variants of the PDP consider different requirements for pickup and delivery customers, assumptions about the transportation scenario, and constraints on the transportation capacity. Berbeglia et al. (2007) conducted a comprehensive survey of PDP formulations and classified them into one-to-one, one-to-many-to-one, and many-to-many schemes.

Some real-world applications focus on supplying the demands of delivery customers. The constraint of visiting *all* pickup customers can, therefore, be relaxed by gathering sufficient commodities from *some* pickup customers. Such a relaxation can substantially reduce the transportation cost and still satisfy the demands of delivery customers. An example application is distributing rental bikes for city traveling, which is greatly promoted in tourism nowadays. The key is to arrange a route for the vehicle (truck) to transport bikes to the rental stations that have reservations and to the popular areas around the city. In this case, visiting all rental stations to pick up bikes is unnecessary; instead, picking up bikes from some rental stations and delivering them to the demanded places will be much more efficient.

This study formulates a new problem, called the *selective pickup and delivery problem* (SPDP), considering the above scenario. Distinguished from the PDP, the proposed SPDP holds two features: First, it relaxes the requirement for visiting all pickup nodes. Second, the SPDP imposes an additional constraint on the vehicle load. For the example of distributing bikes, the SPDP is to find the shortest route that can deliver all demanded bikes without visiting all pickup nodes. Furthermore, it avoids the impractical situation that a vehicle attempts to supply a delivery node with the number of bikes more than its load at some station or to hold a load exceeding its capacity. According to the classification of Berbeglia et al. (2007), the SPDP is of many-to-many scheme, where each node serves as either a source (pickup) or a destination (delivery) of commodities; and the commodities collected from pickup nodes can supply any delivery nodes.

* Corresponding author.
E-mail addresses: ckting@cs.ccu.edu.tw (C.-K. Ting), lhl99p@cs.ccu.edu.tw (X.-L. Liao).

0925-5273/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ijpe.2012.06.009

To resolve the SPDP, this study proposes a memetic algorithm (MA) based on genetic algorithm and local search. Memetic algorithm is a blooming dialect of evolutionary algorithm (EA). In addition to *Darwinism*, MA implements *Lamarckian* or *Baldwinian* theory by integrating a local enhancement, such as local search and repair operator, into the canonical EA. This integration significantly improves the exploitation ability of EA and has been widely shown to provide superior solution quality and high convergence speed (Hart et al., 2004; Le et al., 2009; Meuth et al., 2009; Ong et al., 2006, 2007, 2010; Sinha et al., 2004). The proposed MA adopts a novel representation of candidate solutions to simultaneously deal with the selection of pickup nodes and the visiting order of nodes. A modified 2-opt operator is presented to improve the arrangement of visiting order. The fitness function, furthermore, helps to handle the constraint of vehicle load along the route. This study conducts a series of experiments to examine the MA performance on the SPDP. Furthermore, we apply the proposed method to a real-world rental bikes distribution problem.

The remainder of this paper is organized as follows. Section 2 reviews related work on the PDP. Section 3 presents the formal formulation of the SPDP and proof of NP-hardness. Section 4 sheds light on the proposed MA. The experimental results are presented and discussed in Section 5. Finally, we draw conclusions and recommend the directions for future work in Section 6.

2. Related work

The PDP aims for a minimum-cost route to distribute resources among nodes, including pickup nodes supplying commodity and delivery nodes requiring commodity. This problem can be viewed as a synthesis of the vehicle routing problem (VRP) and object distribution like the knapsack problem. In this study, we classify the PDP formulations according to the attributes of transportation, node, vehicle, and commodity. Table 1 summarizes many variants of PDP and their differences in the setting of these attributes. The first classification criterion—scheme—is proposed in the comprehensive survey of Berbeglia et al. (2007), where the PDP is categorized into one-to-one, one-to-many-to-one, and many-to-many schemes. The main difference among these three schemes is transportation endpoint (Parragh et al., 2008a,b): One-to-many-to-one schemes deliver commodities from the depot to linehaul customers and from backhaul customers to the depot (Gribkovskaia and Laporte, 2008), while one-to-one and many-to-many schemes deal with transportation between customers (Cordeau et al., 2008). Second, the setting of nodes in the PDP is associated with selectivity, depot supply/demand, and time window. Selectivity of nodes relaxes the requirement of visiting all vertices, depot supply/demand indicates whether the depot supplies or demands commodities, and time window limits the time for vehicles to visit nodes. Third, the considerations to vehicles include vehicle capacity and the number of available vehicles. Finally, regarding the properties of commodity, some PDP variants assume that there exists only one type (homogeneous) of commodities to be delivered while some consider multiple types (heterogeneous). The setting of transfer enables temporary stock of commodities (or passengers) in transshipment nodes. The fragment allows for partial delivery of commodities.

The one-to-many-to-one (1-M-1) scheme involves two types of commodities that originate from and terminate at the depot respectively. This scheme is widely used to deal with the issues in reverse logistics (Gonzalez-Torre et al., 2004; Mutha and

Table 1
Classification of PDP variants. (Sel: selectivity, Dep: depot supply/demand, TW: time window, Cap: capacity, Num: number, Ho: homogeneity, Fr: fragment, Tr: transfer).

Scheme	Node			Vehicle		Commodity			Variants
	Sel	Dep	TW	Cap	Num	Ho	Fr	Tr	
1-M-1		+			1				**TSPB** (Gendreau et al., 1996)
		+		+	1				**TSPPD** (Gendreau et al., 1999; Berbeglia and Hahn, 2009)
									SVRPPD (Gribkovskaia et al., 2007)
		+		+	*				**VRPB** (Toth and Vigo, 1997; Garcia-Najera, 2012)
									VRPPD (Gribkovskaia et al., 2001; Hoff et al., 2009)
									VRPSPD (Ai and Kachitvichyanukul, 2009; Çatay, 2010; Subramanian et al., 2010, 2011; Goksal et al., in press)
		+	+	+	*				**VRPBTW** (Duhamel et al., 1997)
									FDPPTW (Wang and Chen, in press)
	+	+		+	1				**SVRPDSP** (Gribkovskaia et al., 2008)
	+	+	+	+	*				**VRPDSPTW** (Gutiérrez-Jarpa et al., 2010)
1-1					1				**PDTSP** (Renaud et al., 2000, 2002)
									TSPPD (Dumitrescu et al., 2010)
									TSPPDF (Erdoğan et al., 2009; Cordeau et al., 2010a)
									TSPPDL (Cordeau et al., 2010b; Tu et al., 2010)
				+	1				***m*-PDTSP** (Hernández-Pérez and Salazar-González, 2009)
				+	*				**Ship Routing Problem** (Pang et al., 2011)
			+	+	1				**S-DARP** (Heilporn et al., 2011)
			+	+	*				**PDPTW** (Cheung et al., 2008; Ropke and Cordeau, 2009; Baldacci et al., 2011)
									VRPPD-G (Psaraftis, 2011)
			+		*			+	**PDPTW** (Shang and Cuff, 1996)
			+	+	*			+	**PDPT** (Cortés et al., 2010)
			+	+	*		+		**m-TSPTW** (Zhang et al., 2009, 2011)
M-M		+		+	1	+			***k*-Delivery TSP** (Chalasani and Motwani, 1999)
									CTSPPD (Anily and Bramel, 1999)
									1-PDTSP (Hernández-Pérez and Salazar-González, 2003, 2004a, 2004b; Hernández-Pérez et al., 2009; Zhao et al., 2009; Louveaux and Salazar-González, 2009)
	+	+		+	1	+			**1-TSP-SELPD** (Falcon et al., 2010)
		+		+	1			+	**SP** (Anily and Hassin, 1992; Bordenave et al., 2009; Anily et al., 2011)
									MSP (Bordenave et al., 2010)
									NCSP (Erdoğan et al., 2010)
	+			+	1	+			**SPDP**

Pokharel, 2009). An example application is the delivery of soft drink, where vehicles need to deliver full containers to customers and retrieve empty containers back to the depot. The 1-M-1 PDP varies in the solution types such as Hamiltonian (Ai and Kachitvichyanukul, 2009; Berbeglia and Hahn, 2009; Çatay, 2010; Gendreau et al., 1999; Goksal et al., in press; Subramanian et al., 2010, 2011; Wang and Chen, in press), delivery-first pickup-second (Duhamel et al., 1997; Garcia-Najera, 2012; Gendreau et al., 1996; Toth and Vigo, 1997), lasso coupling aforementioned types (Gribkovskaia et al., 2001; Hoff et al., 2009), and the times of visiting customers (Gribkovskaia et al., 2007). Some 1-M-1 PDP formulations (Gribkovskaia et al., 2008; Gutiérrez-Jarpa et al., 2010) enable selection of pickup nodes for additional revenue against delivery cost.

In the one-to-one (1-1) scheme, paired pickup and delivery nodes are ordinarily considered, and each commodity has its designate source and destination. Its applications include shipping cargoes (Pang et al., 2011), dial-a-ride system (Heilporn et al., 2011), etc. The variants of 1-1 PDP impose different constraints such as vehicle capacity, number of vehicles, and time window (see Table 1). In addition, some formulations assume that a batch of request can be fragmented (Zhang et al., 2009, 2011) and commodities can be transferred among vehicles (Cortés et al., 2010; Shang and Cuff, 1996).

The many-to-many (M-M) scheme assumes that the vehicle collects commodities from many pickup nodes and supplies them to many delivery nodes. Anily and Bramel (1999) proposed the capacitated traveling salesman problem with pickups and deliveries (CTSPPD) that consists of n pickup customers and n delivery customers, each of which provides or demands one unit of commodity. The CTSPPD follows the general framework of equal amount of total supply and total demand (Berbeglia et al., 2007). Hernández-Pérez and Salazar-González (2003) formulated the one-commodity pickup-and-delivery traveling salesman problem (1-PDTSP). The goal of the 1-PDTSP is to find a minimum-cost route for a vehicle with a fixed capacity to serve all customers. Specially, the 1-PDTSP assumes that the depot provides or consumes an amount of commodities so as to balance total demand. The vehicle capacity is a key issue of the 1-PDTSP. Due to the characteristic of depot, the 1-PDTSP with a large enough or even infinite capacity coincides with the TSP. The swapping problem (SP) (Anily and Hassin, 1992) allows commodities to originate from a collection of pickup nodes and terminate at a collection of delivery nodes, subject to the type of commodities. This problem has been applied to robotic arm and stacker crane operations (Anily et al., 2011; Anily and Hassin, 1992; Bordenave et al., 2009, 2010; Erdoğan et al., 2010). In addition, Falcon et al. (2010) employed the many-to-many scheme to the carrier-based coverage repair problem in wireless sensor networks.

The proposed SPDP belongs to the many-to-many scheme. Table 1 shows that the SPDP is a novel variant of PDP. The difference between SPDP and other variants of many-to-many PDP lies in the problem setting for nodes and commodities. For example, a major difference between the 1-PDTSP and the SPDP is that the depot in the 1-PDTSP provides commodity whereas that in the SPDP does not. In addition, both the CTSPPD and the 1-PDTSP do not consider selectivity of pickup nodes.

3. The selective pickup and delivery problem

The proposed formulation of the SPDP breaks the assumption of demand equilibrium and relaxes the requirement for visiting all pickup nodes. Let $G=(V,E)$ be a complete graph with vertex set $V=(v_0,\ldots,v_n)$ and edge set $E=\{(v_i,v_j)|v_i,v_j\in V, v_i\neq v_j\}$, in which each edge (v_i,v_j) has a cost[1] $c_{ij}>0$. Each node v_i provides or demands a number d_i of commodities. The node v_0 serves as the starter (depot) and has $d_0=0$; the other nodes are classified into two sets: $V^+=\{v_i|v_i\in V, d_i>0\}$ of pickup nodes and $V^-=\{v_i|v_i\in V, d_i<0\}$ of delivery nodes.

The SPDP seeks for a minimum-cost route for a vehicle to gain commodities from *some* pickup nodes and supply the demands of *all* delivery nodes, subject to the constraint that the vehicle load at each node should be non-negative and never exceed the vehicle capacity Q. Restated, the objective of the SPDP is to find a permutation $\boldsymbol{p}=(v_0,v_{(1)},v_{(2)},\ldots,v_{(m)})$ such that the overall cost is minimum, where m is the total number of selected pickup nodes and delivery nodes, and $v_{(i)}$ represents the ith visiting node. Formally, let x_{ij} be the decision variable with

$$x_{ij}=\begin{cases}1 & \text{vehicle travels from } v_i \text{ to } v_j\\ 0 & \text{otherwise}\end{cases}$$

The SPDP can be formulated as the following integer linear programming model:

$$\min \sum_{v_i,v_j\in V} c_{ij}x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{v_i\in V} x_{ij}=\sum_{v_i\in V} x_{ji}\leq 1, \quad \forall v_j\in V^+ \tag{2}$$

$$\sum_{v_i\in V} x_{ij}=\sum_{v_i\in V} x_{ji}=1, \quad \forall v_j\in V^-\cup\{v_0\} \tag{3}$$

$$\sum_{v_i,v_j\in S} x_{ij}\leq |S|-1, \quad \forall S\subseteq V\backslash\{v_0\} \tag{4}$$

$$0\leq \ell_{(t)}\leq Q, \quad \forall t\in\{1,\ldots,m\} \tag{5}$$

$$x_{ij}\in\{0,1\} \tag{6}$$

The objective function (1) minimizes the traveling cost subject to load constraint (5), where $\ell_{(t)}$ denotes vehicle load at $v_{(t)}$ along the visiting order, i.e., $\ell_{(t)}=\ell_{(t-1)}+d_{(t)}$ with $\ell_{(0)}=0$. Inequality (2) enables selection of pickup nodes; in addition, (2) and (3) guarantee that the selected pickup nodes and the delivery nodes are visited exactly once. Constraint (4) eliminates sub-tours among customers and (6) indicates binary variables.

This formulation relaxes the requirement of visiting all nodes by allowing *selectivity* of pickup nodes. This selectivity severely increases the complexity of the SPDP, specifically, $\mathcal{O}(2^{|V^+|}\cdot m!)$. Note that this complexity is higher than that of an n-city TSP by approximately $2^{|V^+|}$ times. Moreover, the SPDP imposes a constraint on the composition of routes: All delivery nodes must be visited and supplied with enough commodities ($\ell_{(t)}\geq 0$); meanwhile, the vehicle load should not exceed its capacity ($\ell_{(t)}\leq Q$). In practice, the SPDP is highly relevant to the logistic applications that focus on supplying demands of all customers (delivery nodes) from some providers (pickup nodes).

Proposition 1. *The SPDP is NP-hard.*

Proof. Firstly, a decision version is introduced to determine whether there exists a feasible route such that the total cost is less than a predefined cost C. By checking the load constraints along the planned route and recording the cost to compare with C, the decision version of the SPDP takes $\mathcal{O}(n)$ for solution verification. The polynomial time verifiability confirms SPDP $\in$ NP.

[1] The cost between nodes is ordinarily defined as their distance. This study also adopts distance metric as cost.

Next, we reduce the TSP to the SPDP given that the TSP is proved to be NP-hard. The reduction aims to prove the TSP is a special case of the SPDP, where the starter is a duplicate of the assumed single pickup node. Given an SPDP instance $G=(V,E)$, assume $v_p \in V$ is the *only one* pickup node and the cost of an edge incident to starter v_0 equals that of an edge incident to v_p, i.e., $c_{0i}=c_{pi}, \forall v_i \in V$. Here the constraints of SPDP $\sum_{v_i \in V} d_i \geq 0$ and $Q \geq d_p$ need to hold. Note that v_0 must be followed by v_p for feasibility in loading. In essence, v_0 can be viewed as a duplicate of v_p and a feasible route $(v_0, v_p, v_{(1)}, \ldots, v_{(n)})$ turns out to be $(v_p, v_{(1)}, \ldots, v_{(n)})$. This reduction transforms a TSP instance into an SPDP instance by choosing an arbitrary node as the single pickup node v_p and making starter v_0 a duplicate of v_p. In other words, there exists a feasible route in the SPDP such that the total cost is less than C if and only if the corresponding TSP has a route at the cost less than C. Fig. 1 provides an example.

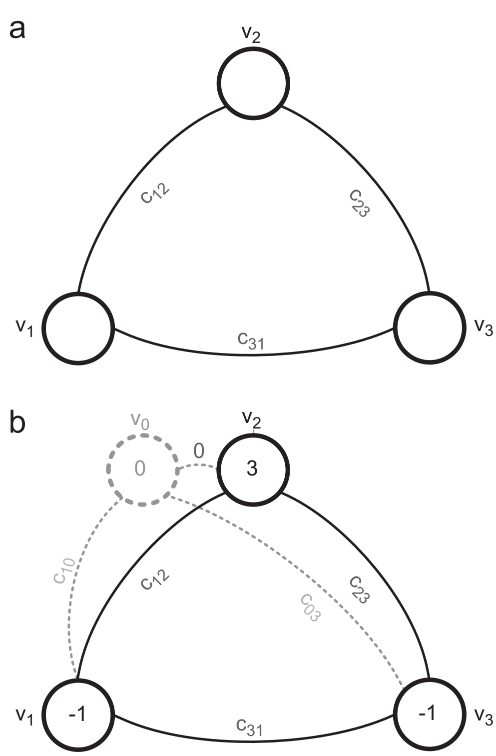

Fig. 1. Example reduction of the TSP to the SPDP. (a) A TSP instance. Each edge is associated with a cost indicated aside. (b) The corresponding SPDP instance ($Q \geq 3$) using the proposed reduction method. The figure inside a circle represents demand d_i, and v_2 is the only one pickup node. The edge cost (dashed lines) incident to v_0 equals that incident to v_2.

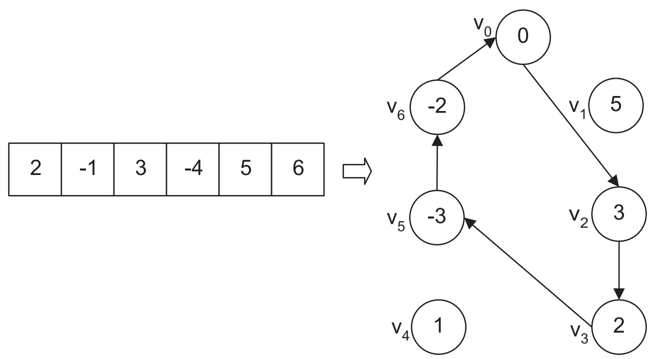

Fig. 2. An example representation for an SPDP instance with $n=6$ nodes. The number inside a circle denotes demand d_i.

Since the decision version of the SPDP can be verified in polynomial time and there exists a polynomial-time reduction transforming the TSP to the SPDP, the decision version of the SPDP is NP-complete. The SPDP is at least as hard as its decision version; hence, the SPDP is proved to be NP-hard. □

4. The proposed memetic algorithm

This study proposes an MA to solve the SPDP. Algorithm 1 presents the framework of the proposed MA. The MA implements the GA scheme and additionally adopts a local search operator

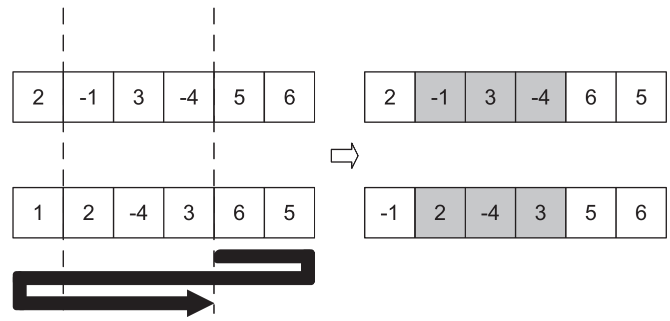

Fig. 3. Example of OX operation. The black arrow indicates the direction of determining offspring genes.

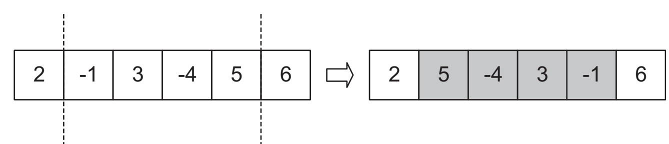

Fig. 4. Example of inversion mutation.

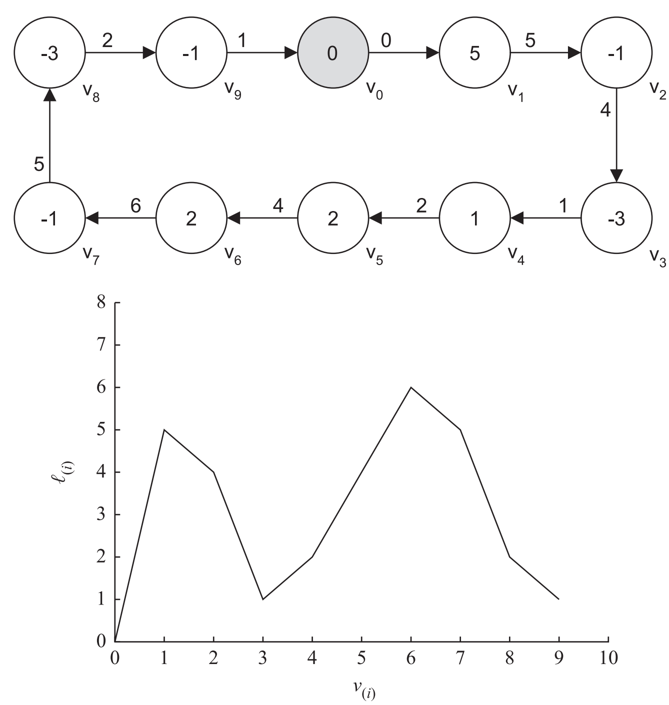

Fig. 5. An example route (top) and the corresponding variation of vehicle load $\ell_{(i)}$ (bottom). The number inside a circle represents demand $d_{(i)}$ and the number above an edge denotes load $\ell_{(i)}$.

designed for the SPDP. Following the GA scheme, the MA *represents* candidate solutions as chromosomes. The fitness function evaluates the quality (*fitness*) of candidate solutions (chromosomes). Evolutionary algorithms, such as GA and MA, manipulate a set (*population*) of chromosomes to search for the optimal solution. After initializing the population, MA embarks on the evolutionary process. First, the *selection* operator picks two chromosomes from the population to serve as *parents*. The *crossover* operator then exchanges the information between these two parents to produce their *offspring*. A predetermined crossover rate defines the probability of performing crossover. Analogously, *mutation* is performed with a probability, called mutation rate, to alter slightly some genes in the offspring. Afterward, the local search is performed to improve the chromosome.

Table 2
Parameter setting in the experiments.

	MA	GA	TS
Representation		Modified permutation	
Initialization		Random	
Population/neighborhood size		500	
Mutation/neighborhood function		Bit-flip ($p_m = 1/\lvert V^+\rvert$) + inversion ($p_m = 1.0$)	
Selection	Binary tournament	Binary tournament	–
Crossover	Order crossover ($p_c = 1.0$)	Order crossover ($p_c = 1.0$)	–
Survival	$(\mu+\lambda)$	$(\mu+\lambda)$	–
Local search	2-opt	–	–
Tabu tenure	–	–	10
Termination (#generations)	3000	10 000	10 000

Table 3
Average route lengths and average numbers (after slash marks) of selected pickup nodes obtained from MA for different capacities Q and gain γ values.

Instance	Q	$\gamma = 100$	$\gamma = 200$	$\gamma = 300$	$\gamma = 400$
n100mosA(91)/42	200	5354.16/2.10	–	–	–
	400	5274.23/2.17	5268.42/2.00	5177.06/1.00	–
	600	5232.19/2.20	5215.64/2.00	5167.87/1.00	5170.98/1.00
	1000	5243.14/2.27	5189.47/2.00	5167.87/1.00	5167.87/1.00
n100mosB(92)/47	200	5365.42/3.00	–	–	–
	400	5272.99/3.00	5231.55/2.00	5200.73/1.00	–
	600	5261.24/3.00	5236.76/2.00	5166.53/1.00	5172.61/1.00
	1000	5263.58/3.00	5185.84/2.00	5164.74/1.00	5160.39/1.00
n200mosA(181)/94	200	8837.58/5.00	–	–	–
	400	7646.87/5.03	7715.88/3.00	7638.32/2.00	–
	600	7560.72/5.07	7515.59/3.00	7467.33/2.03	7581.82/2.00
	1000	7549.46/5.00	7457.90/3.00	7418.36/2.00	7418.96/2.00
n200mosB(184)/88	200	9523.21/5.07	–	–	–
	400	8329.73/5.00	8433.09/3.00	8411.98/2.00	–
	600	8289.96/5.00	8233.01/3.00	8205.01/2.03	8292.87/2.00
	1000	8211.91/5.00	8187.13/3.00	8125.90/2.00	8149.09/2.03
n300mosA(277)/141	200	12 033.74/7.17	–	–	–
	400	9613.65/7.00	9612.50/4.00	9741.63/3.00	–
	600	9321.98/7.03	9289.82/4.00	9281.69/3.00	9228.90/2.00
	1000	9239.73/7.03	9148.37/4.00	9123.31/3.00	9104.33/2.00
n300mosB(279)/138	200	12 884.81/7.47	–	–	–
	400	9807.71/7.17	9816.74/4.03	9984.83/3.00	–
	600	9476.11/7.23	9377.53/4.00	9420.21/3.00	9376.53/2.03
	1000	9412.23/7.20	9270.51/4.00	9206.01/3.00	9189.70/2.00
n400mosA(358)/172	200	21 590.76/9.93	–	–	–
	400	13 303.94/9.57	12 940.68/5.07	13 306.66/4.00	–
	600	11 802.30/9.40	11 447.40/5.03	11 841.61/4.00	11 781.90/3.00
	1000	11 912.59/9.50	10 972.00/5.00	11 025.52/4.00	11 018.23/3.07
n400mosB(364)/183	200	22 264.16/10.07	–	–	–
	400	13 338.54/10.03	12 937.36/5.07	13 157.84/4.00	–
	600	11 687.35/9.67	11 311.26/5.07	11 690.35/4.07	11 652.05/3.00
	1000	10 996.21/9.77	10 694.08/5.00	10 708.42/4.00	10 772.36/3.00
n500mosA(453)/232	200	33 833.22/12.37	–	–	–
	400	19 272.46/12.57	18 218.14/6.33	17 521.97/4.20	–
	600	14 664.66/12.23	14 321.33/6.40	13 820.93/4.20	13 691.69/3.20
	1000	13 965.66/12.33	12 151.28/6.00	12 023.86/4.07	12 347.79/3.10
n500mosB(454)/228	200	32 295.67/11.60	–	–	–
	400	18 430.17/11.57	18 041.28/6.23	17 646.08/4.13	–
	600	14 450.89/11.23	14 363.94/6.23	14 075.24/4.03	13 752.26/3.13
	1000	13 477.41/11.13	12 455.25/6.07	12 343.47/4.00	12 350.01/3.03

The process of reproduction, i.e., selection–crossover–mutation–local-search, is repeated until the offspring population is filled. The Lamarckism, implemented by local search, enhances the search ability of MA. Based on the Darwinian theory of "Survival of the Fittest", the *survival* operator selects the fittest chromosomes from the offspring population with or without the parental population. The selected chromosomes constitute the next-generation population. For the SPDP, the components of MA need to be modified. Restated, this study presents a novel representation of candidate solutions and related operators to address the constraints on the vehicle load. A local search operator is devised to enhance the visiting order for the SPDP. More details about the proposed MA are described below.

Algorithm 1. Memetic algorithm.

initialize population P;
evaluate P;
while (not terminated)
{
  $P_s = \text{Select}(P)$;
  $P_c = \text{Crossover}(P_s)$;
  $P_m = \text{Mutate}(P_c)$;
  $P' = \text{LocalSearch}(P_m)$;
  evaluate P';
  $P = \text{Survive}(P, P')$;
};

4.1. Representation

This study modifies the order-based representation of chromosomes to indicate both the visiting order and the selection of pickup customers in the SPDP. Specifically, for an SPDP with n nodes (except the depot v_0), we use a permutation of n integers to represent the visiting order of all pickup and delivery nodes. In addition, the selection of a pickup node is designated by the sign of its order in the chromosome, where a positive integer indicates the node is selected and a negative integer indicates it is omitted from the route. Fig. 2 illustrates a chromosome for an SPDP with $n=6$. In the example route, the genes valued -1 and -4 represent that pickup nodes v_1 and v_4 are excluded from the route, respectively. An important feature of this fixed-length representation is its capability to handle the varying number of selected pickup nodes and guarantee that each node occurs at most once in the route.

4.2. Fitness evaluation and constraint handling

The fitness function is vitally important to EAs because it explicitly or implicitly affects the search direction. An effective fitness function must render sufficient information about the search direction and clearly distinguish between good and bad candidate solutions. The fitness function is problem-dependent in essence. In this study, we use the objective function (1) as the fitness function $f(\boldsymbol{p})$ for the MA.

In the SPDP, some chromosomes may be infeasible due to violation of the constraint on vehicle load, i.e., the vehicle load should not exceed its capacity and must be non-negative at each node. Comparison among feasible and infeasible chromosomes becomes an issue at fitness evaluation for the constrained optimization problems. To address this issue, this study adopts the method of Deb (2000). For minimization of route cost in the SPDP, chromosome $\boldsymbol{p}_1$ is considered to be better than chromosome $\boldsymbol{p}_2$ if

1. both $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are feasible and $f(\boldsymbol{p}_1) < f(\boldsymbol{p}_2)$, or
2. $\boldsymbol{p}_1$ is feasible while $\boldsymbol{p}_2$ is infeasible, or

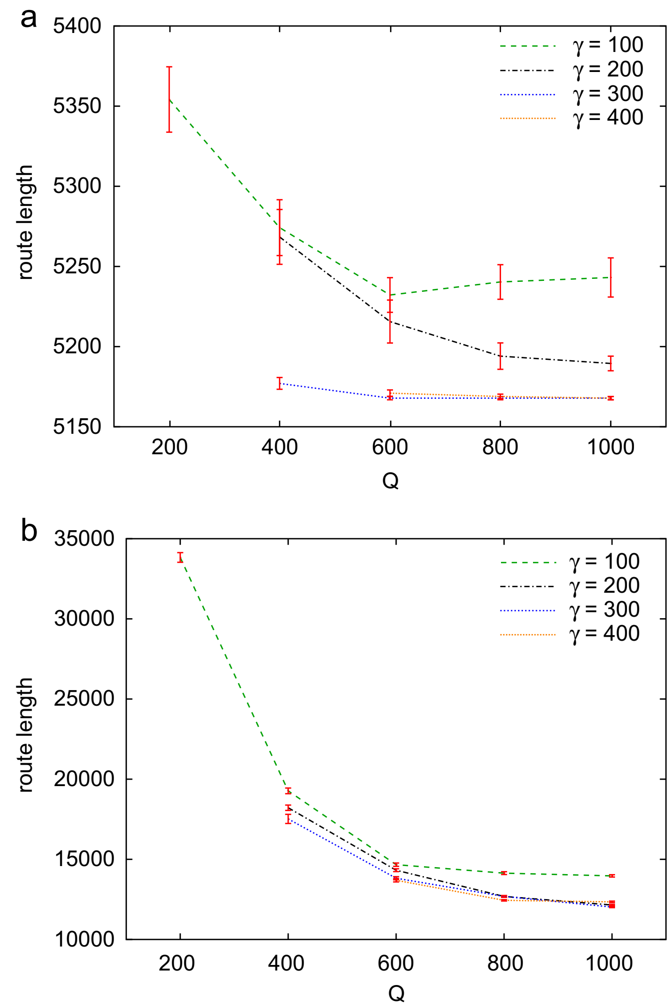

Fig. 6. Variation in route length with respect to Q. (a) n100mosA(91) and (b) n500mosA(453).

3. both $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are infeasible and $\boldsymbol{p}_1$ has less serious constraint violation than $\boldsymbol{p}_2$.

The above criteria prefer low-cost feasible solutions; in addition, they eliminate the need for parameter tuning in the penalty function that is commonly used for constraint handling. Notably, the third criterion requires an evaluation measure for violation of constraint. This study presents a violation measure

$$g(\boldsymbol{p}) = \ell_{\text{exc}} + |\ell_{\text{neg}}| \tag{7}$$

with

$$\ell_{\text{exc}} = \max_{i \in \{1,\ldots,m\}} (\ell_{(i)}, Q) - Q$$

$$\ell_{\text{neg}} = \min_{i \in \{1,\ldots,m\}} (\ell_{(i)}, 0)$$

where ℓ_{exc} represents the amount of load exceeding the vehicle capacity and ℓ_{neg} denotes the shortage of vehicle load. According to (7), the violation measure $g(\boldsymbol{p})$ considers both the exceeding and short amounts of vehicle load along the route. As the basis of parent selection and survivor selection, this fitness evaluation handles the constraint by leading the search toward feasible solutions.

4.3. Genetic operators

The proposed MA is based on the evolutionary scheme of GA. In the reproduction phase, the selection operator chooses

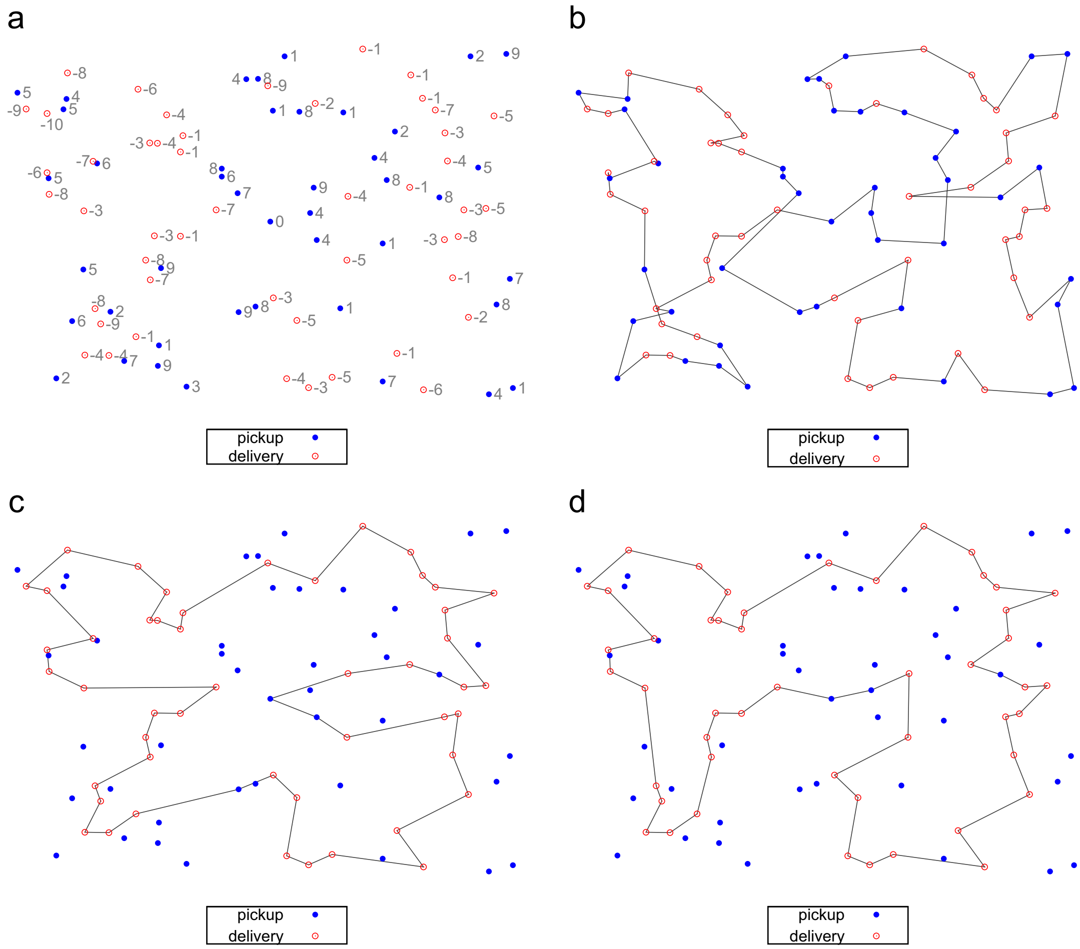

Fig. 7. Routes obtained from MA for different capacities and gains on n100mosA(91). (a) Node distribution. (b) Route for $\gamma=0$ and $Q=200$. (c) Route for $\gamma=100$ and $Q=200$. (d) Route for $\gamma=100$ and $Q=1000$.

chromosomes as parents from the population; then the crossover and mutation operators alter the genetic information in the parents to generate offspring. Afterward, the survival operator selects the chromosomes that can survive into the next generation. These genetic operators, to wit, selection, crossover, mutation, and survival, need to be designed for the MA to address the SPDP.

The selection operator is ordinarily based on fitness and enables fitter chromosomes to have a higher probability to be selected as parents. Several selection operators have been proposed (Bäck et al., 1997; Eiben and Smith, 2003). In this study, we adopt the binary tournament selection in view of its recognized good performance. The binary tournament selection operator chooses the better of two random chromosomes as a parent. Performing this selection twice yields a pair of parents for subsequent reproduction, i.e., crossover and mutation.

Next, the crossover operation exchanges and recombines the genetic information of parents. The crossover operator for order-based representation requires a special design to ensure the legality of an order, *i.e.*, no duplicate numbers. This study uses the well-known order crossover (OX) (Davis, 1985) to manipulate the order-based chromosomes for the SPDP. This crossover randomly chooses two cut points to divide each parent into two segments. An offspring directly inherits the first segment of one parent and fills the remainder genes with the genes of the other parent in order. This crossover can avoid duplicate numbers in a chromosome and thus satisfies the requirement for a permutation. Regarding the selection of pickup nodes, the signs are kept in the crossover operation to reserve information of being selected or omitted. As Fig. 3 illustrates, this crossover operator can deal with both the visiting order and selection of pickup nodes.

The mutation operation slightly changes the composition of offspring to introduce diversity. For the SPDP, the mutation operator needs to be specially designed for two aspects: one for selection of pickup nodes and another one for permutation of visiting order. For the former, the bit-flip mutation for binary GA is applicable. This mutation randomly changes the selection of pickup nodes by flipping the sign of corresponding genes. For the latter, the mutation operators for order-based GA are considered, which afford to slightly change the route without violating the legality of a permutation. This study adopts the inversion operator, which inverts the sub-string between two randomly picked genes (see Fig. 4).

The survival operator genuinely implements the principle of survival of the fittest: Only the fittest individuals are selected as parents for the next generation. The methods of survivor selection can be classified according to the number of parents who compete for survival. For example, the (μ,λ) survivor selection considers only the offspring population, whereas the $(\mu+\lambda)$ survivor selection merges the parental and the offspring populations to compete for survival. This study uses the $(\mu+\lambda)$ survivor selection in the experiments.

4.4. Local search

The MA employs the 2-opt operator to enhance the arrangement of visiting order. The original 2-opt operator inverts segments for a shorter route iteratively. However, for the SPDP, the 2-opt operator needs to be adapted in that the constraint of vehicle load,

Table 4
Average route lengths obtained from TS, GA, and MA for different capacities Q and gains γ. The figure after the slash mark represents the average number of selected pickup nodes over 30 trials. Boldface denotes the shortest route among the three methods.

Instance	γ	Q	TS	GA	MA
n100mosA(91)/42	100	400	**5264.27/2.57**	5442.64/2.33	5274.23/2.17
		600	5278.90/2.70	5445.00/2.53	**5232.19/2.20**
		1000	5272.44/2.43	5394.89/2.63	**5243.14/2.27**
	200	400	**5247.61/2.00**	5430.93/2.00	5268.42/2.00
		600	5257.72/2.03	5433.85/2.03	**5215.64/2.00**
		1000	5224.41/2.00	5394.85/2.00	**5189.47/2.00**
	400	600	5248.93/1.00	5366.06/1.00	**5170.98/1.00**
		800	5214.57/1.03	5442.21/1.00	**5168.91/1.00**
		1000	5228.33/1.00	5459.30/1.00	**5167.87/1.00**
n100mosB(92)/47	100	400	5289.23/3.00	5476.93/3.00	**5272.99/3.00**
		600	**5250.85/3.00**	5460.65/3.00	5261.24/3.00
		1000	**5251.75/3.00**	5432.92/3.00	5263.58/3.00
	200	400	5297.03/2.00	5478.77/2.03	**5231.55/2.00**
		600	5260.83/2.07	5467.76/2.00	**5236.76/2.00**
		1000	5231.18/2.03	5359.74/2.00	**5185.84/2.00**
	400	600	5297.72/1.00	5436.45/1.00	**5172.61/1.00**
		800	5247.98/1.03	5393.40/1.00	**5156.03/1.00**
		1000	5262.79/1.07	5412.67/1.00	**5160.39/1.00**
n200mosA(181)/94	100	400	8232.43/5.17	8558.26/5.07	**7646.87/5.03**
		600	8231.82/5.23	8300.09/5.23	**7560.72/5.07**
		1000	8195.95/5.40	8368.36/5.03	**7549.46/5.00**
	200	400	8293.84/3.10	8780.34/3.07	**7715.88/3.00**
		600	8162.07/3.33	8468.49/3.20	**7515.59/3.00**
		1000	8121.85/3.40	8200.15/3.07	**7457.90/3.00**
	400	600	8320.76/2.00	8770.12/2.00	**7581.82/2.00**
		800	8237.71/2.00	8494.05/2.03	**7471.26/2.03**
		1000	8113.91/2.10	8314.63/2.07	**7418.96/2.00**
n200mosB(184)/88	100	400	8994.44/5.23	9329.64/5.13	**8329.73/5.00**
		600	8937.45/5.23	9159.83/5.10	**8289.96/5.00**
		1000	8927.34/5.23	9116.48/5.13	**8211.91/5.00**
	200	400	8995.63/3.03	9909.09/3.07	**8433.09/3.00**
		600	8885.00/3.33	9302.96/3.20	**8233.01/3.00**
		1000	8838.65/3.47	9164.78/3.10	**8187.13/3.00**
	400	600	9049.66/2.00	9822.91/2.00	**8292.87/2.00**
		800	8960.64/2.10	9383.16/2.07	**8178.86/2.00**
		1000	8890.76/2.27	9146.73/2.10	**8149.09/2.03**
n300mosA(277)/141	100	400	10 312.13/7.30	12 844.86/7.33	**9613.65/7.00**
		600	10 337.44/7.17	12 389.25/7.20	**9321.98/7.03**
		1000	10 286.04/7.20	12 185.67/7.33	**9239.73/7.03**
	200	400	10 344.39/4.03	14 292.87/4.07	**9612.50/4.00**
		600	10 244.07/4.20	12 696.50/4.13	**9289.82/4.00**
		1000	10 122.08/4.23	12 140.47/4.30	**9148.37/4.00**
	400	600	10 254.53/2.10	13 447.19/2.10	**9228.90/2.00**
		800	10 233.89/2.17	12 659.25/2.07	**9156.36/2.00**
		1000	10 147.08/2.30	12 347.45/2.20	**9104.33/2.00**

i.e., $0 \le \ell_{(i)} \le Q$, may be violated by inversion. To address this issue, we look into the variation of vehicle load (Fig. 5) and notice that the peaks and valleys of vehicle loads along a feasible route, e.g., ℓ_0, $\ell_{(1)}$, $\ell_{(3)}$, $\ell_{(6)}$, and $\ell_{(9)}$, must be bounded by [0,Q]. Following this limitation can achieve the feasibility of routes. In addition, permutation of nodes in a segment between peaks or valleys will not affect the feasibility since the subtotal of demands will keep fixed within a segment. Accordingly, the modified 2-opt operator partitions the route into several segments by the positions of peaks and valleys of vehicle loads. Note that peaks and valleys appear at the transition from pickup to delivery nodes and that from delivery to pickup nodes, respectively. For the example route in Fig. 5, the 2-opt operation is performed on the nodes within segments $\{v_{(2)}, \ldots, v_{(3)}\}$, $\{v_{(4)}, \ldots, v_{(6)}\}$, and $\{v_{(7)}, \ldots, v_{(9)}\}$.

5. Experimental results

This study conducts a series of experiments to evaluate the effectiveness of the proposed MA on the SPDP, in comparison with GA and tabu search (TS). The benchmark instances of the SPDP are modified from the PDP instances TS2004t2 and TS2004t3 used in (Hernández-Pérez and Salazar-González, 2004b; Hernández-Pérez et al., 2009). In modifying these instances for the SPDP, we set $d_0 = 0$ for starter v_0 and adjust demand d_n of v_n to satisfy $\sum_{i=0}^{n} d_i = 0$. Additionally, the nodes with zero demand, i.e., $d_i = 0$, are removed from the SPDP instances. Here the SPDP instances are denoted by $X(Y)/Z$, where X is the original instance name of PDP, Y is the number of nodes in the SPDP instance, and optional Z is the number of pickup nodes. For example, n200mosA(181) denotes a 181-node SPDP instance modified from n200mosA. The cost between two nodes is defined as their distance. Hence, the SPDP seeks for the shortest route that can supply all delivery nodes with some selected pickup nodes. The experiments include several sizes of vehicle capacity. To investigate the characteristics of the SPDP, we further introduce parameter γ as a gain in supply for each pickup node, that is, $d'_i = d_i + \gamma$ for all $v_i \in V^+$.

The test algorithms include the proposed MA, GA, and TS. Table 2 summarizes their parameter setting in the experiments.

Table 5
Average route lengths obtained from TS, GA, and MA for different capacities Q and gains γ. The figure after the slash mark represents the average number of selected pickup nodes over 30 trials. Boldface denotes the shortest route among the three methods.

Instance	γ	Q	TS	GA	MA
n300mosB(279)/138	100	400	10 485.94/7.67	13 776.30/7.57	**9807.71/7.17**
		600	10 447.02/7.50	12 617.50/7.47	**9476.11/7.23**
		1000	10 411.95/7.43	12 780.34/7.67	**9412.23/7.20**
	200	400	10 498.41/4.23	14 589.01/4.07	**9816.74/4.03**
		600	10 380.38/4.13	13 142.27/4.17	**9377.53/4.00**
		1000	10 297.00/4.27	12 407.35/4.17	**9270.51/4.00**
	400	600	10 464.34/2.10	13 833.61/2.10	**9376.53/2.03**
		800	10 383.55/2.27	12 893.38/2.07	**9189.02/2.00**
		1000	10 247.41/2.30	12 690.50/2.10	**9189.70/2.00**
n400mosA(358)/172	100	400	14 481.88/10.03	24 680.78/9.93	**13 303.94/9.57**
		600	14 284.20/10.00	22 108.25/10.23	**11 802.30/9.40**
		1000	14 245.97/9.80	20 980.95/10.37	**11 912.59/9.50**
	200	400	14 691.80/5.10	28 103.77/5.23	**12 940.68/5.07**
		600	14 174.06/5.40	23 605.40/5.37	**11 447.40/5.03**
		1000	14 066.04/5.50	21 606.74/5.57	**10 972.00/5.00**
	400	600	14 502.50/3.00	28 117.26/3.00	**11 781.90/3.00**
		800	14 239.25/3.10	25 107.62/3.13	**11 259.50/3.00**
		1000	14 129.12/3.27	23 230.26/3.13	**11 018.23/3.07**
n400mosB(364)/183	100	400	14 598.81/10.07	24 817.27/10.33	**13 338.54/10.03**
		600	14 305.98/10.23	21 890.27/10.50	**11 687.35/9.67**
		1000	14 399.51/10.07	21 517.26/10.50	**10 996.21/9.77**
	200	400	14 808.38/5.13	28 749.01/5.20	**12 937.36/5.07**
		600	14 372.97/5.47	24 867.95/5.43	**11 311.26/5.07**
		1000	13 990.26/5.57	21 118.43/5.63	**10 694.08/5.00**
	400	600	14 586.65/3.00	28 693.80/3.00	**11 652.05/3.00**
		800	14 310.83/3.13	25 007.68/3.13	**11 030.08/3.03**
		1000	14 212.90/3.33	23 720.46/3.30	**10 772.36/3.00**
n500mosA(453)/232	100	400	21 280.82/12.40	38 981.51/12.63	**19 272.46/12.57**
		600	20 123.01/12.37	34 415.62/12.93	**14 664.66/12.23**
		1000	20 149.26/12.53	32 113.63/13.07	**13 965.66/12.33**
	200	400	22 406.07/6.10	48 620.13/6.43	**18 218.14/6.33**
		600	20 627.82/6.47	39 491.62/6.67	**14 321.33/6.40**
		1000	19 545.91/6.93	33 675.57/7.33	**12 151.28/6.00**
	400	600	21 893.33/3.67	46 407.06/3.40	**13 691.69/3.20**
		800	20 666.93/3.70	40 865.75/3.47	**12 444.84/3.00**
		1000	20 072.16/3.93	37 751.00/3.83	**12 347.79/3.10**
n500mosB(454)/228	100	400	21 612.45/11.67	41 197.12/11.83	**18 430.17/11.57**
		600	20 668.00/11.43	36 085.44/12.50	**14 450.89/11.23**
		1000	20 612.56/11.67	33 475.65/12.57	**13 477.41/11.13**
	200	400	22 798.68/6.20	49 710.06/6.27	**18 041.28/6.23**
		600	20 994.43/6.33	40 617.13/6.53	**14 363.94/6.23**
		1000	20 188.55/6.80	35 044.03/7.47	**12 455.25/6.07**
	400	600	22 248.14/3.23	48 411.66/3.17	**13 752.26/3.13**
		800	21 168.54/3.40	41 682.90/3.43	**12 883.73/3.03**
		1000	20 570.37/3.73	38 689.85/3.53	**12 350.01/3.03**

The GA follows the evolutionary framework and operators of MA but does not apply local search. The TS utilizes the mutation operator as its neighborhood function, i.e., flipping a pickup node selection and inverting a random sub-route. The neighborhood size of TS is set to be 500 such that the numbers of fitness evaluations in each generation are equal for GA, MA, and TS. The tabu tenure is empirically set to be 10. Each test includes 30 independent runs concerning the stochastic nature of test algorithms.

5.1. *Influence of Q and* γ

The purpose of the first experiment is to investigate the influence of vehicle capacity Q and gain γ on the route length. Table 3 presents the results for MA on the ten SPDP instances with different Q and γ values. Some experimental settings (e.g., $Q{=}200$ with $\gamma \geq 200$) are omitted in that they cannot yield feasible solutions. The table shows that the route length decreases with the increase in γ, which allows more pickup nodes to be omitted from the route. Fig. 6 plots the route lengths with respect to capacity Q for different γ values. The plots reconfirm that the SPDP results in shorter routes as γ increases. Regarding the influence of capacity Q, the decrease in capacity intensifies the restriction on both the permutation of route and the selection of pickup nodes for a feasible solution. This restriction becomes particularly serious as Q approaches the upper limit of $d_i+\gamma$; for example, the setting of $\gamma=100$ with $Q{=}200$ gives a drastic increase in route length.

Fig. 7 illustrates the routes obtained from the proposed MA on n100mosA(91) instance, of which $\gamma=0$ is a PDP instance and $\gamma=100$ is an SPDP instance. These results exhibit the benefit of the SPDP: The SPDP renders a shorter route than the PDP does, which is of great help for the applications that emphasize supplying all the demands. Notably, the route in Fig. 7(b) with ‘crosses’ seems to be an imperfect route in the sense of the TSP. However, given the constraint on the vehicle load, this cross is of necessity for the optimal route. Figs. 7(c) and 7(d) demonstrate that the increase in capacity Q relaxes the constraint on vehicle

Table 6
Results of one-tailed paired t-test of the route lengths obtained from X and Y algorithms (denoted by X vs. Y) for different capacities Q and gains γ. Positive p-values indicate that X is superior to Y, and vice versa. Boldface denotes that X is significantly better than Y with confidence level $\alpha = 0.05$.

γ	Q	Instance	TS vs. GA	MA vs. GA	MA vs. TS	Instance	TS vs. GA	MA vs. GA	MA vs. TS
100	400	n100mosA(91)	**+1.92E−06**	**+9.51E−06**	−3.21E−01	n100mosB(92)	**+1.24E−04**	**+3.07E−05**	+2.72E−01
	600		**+8.05E−06**	**+6.27E−08**	**+7.36E−03**		**+3.45E−06**	**+1.03E−05**	−3.29E−01
	1000		**+3.50E−04**	**+1.23E−05**	+8.13E−02		**+6.25E−06**	**+2.42E−05**	−3.04E−01
200	400		**+6.86E−07**	**+9.74E−06**	−1.67E−01		**+3.18E−04**	**+4.94E−07**	**+3.28E−02**
	600		**+5.91E−05**	**+1.99E−06**	**+2.50E−02**		**+9.74E−07**	**+8.35E−08**	+1.42E−01
	1000		**+2.66E−05**	**+9.70E−07**	**+1.42E−02**		**+8.54E−04**	**+2.08E−05**	**+5.72E−03**
400	600		**+2.45E−03**	**+2.50E−06**	**+1.20E−04**		**+1.95E−03**	**+9.01E−08**	**+1.01E−05**
	800		**+1.34E−08**	**+3.44E−10**	**+2.02E−07**		**+5.29E−05**	**+6.53E−09**	**+6.97E−07**
	1000		**+9.95E−08**	**+8.81E−10**	**+5.99E−05**		**+7.07E−05**	**+1.57E−09**	**+1.14E−05**
100	400	n200mosA(181)	**+1.14E−04**	**+1.18E−13**	**+9.94E−25**	n200mosB(184)	**+7.56E−04**	**+3.85E−12**	**+1.98E−27**
	600		+1.29E−01	**+3.74E−16**	**+2.25E−25**		**+1.94E−03**	**+1.53E−14**	**+1.15E−25**
	1000		**+1.36E−02**	**+5.17E−13**	**+8.93E−30**		**+5.31E−03**	**+1.25E−14**	**+9.97E−32**
200	400		**+7.12E−05**	**+4.11E−11**	**+2.80E−22**		**+1.17E−08**	**+1.28E−13**	**+2.13E−21**
	600		**+3.06E−04**	**+1.09E−13**	**+6.67E−30**		**+1.04E−04**	**+2.65E−12**	**+1.77E−31**
	1000		+1.17E−01	**+3.74E−14**	**+3.29E−27**		**+4.75E−05**	**+6.16E−15**	**+3.64E−30**
400	600		**+2.40E−04**	**+5.93E−12**	**+1.45E−28**		**+2.84E−07**	**+7.97E−14**	**+7.45E−35**
	800		**+4.69E−03**	**+9.89E−13**	**+6.92E−32**		**+9.67E−06**	**+9.40E−16**	**+2.08E−29**
	1000		**+1.49E−03**	**+4.01E−16**	**+5.31E−33**		**+2.06E−03**	**+1.49E−13**	**+9.10E−40**
100	400	n300mosA(277)	**+4.02E−15**	**+3.31E−18**	**+1.19E−21**	n300mosB(279)	**+2.89E−20**	**+4.13E−24**	**+5.97E−16**
	600		**+1.70E−15**	**+4.82E−20**	**+1.32E−31**		**+5.10E−18**	**+1.03E−22**	**+6.26E−31**
	1000		**+7.04E−15**	**+5.17E−20**	**+4.90E−38**		**+5.81E−18**	**+2.26E−22**	**+2.48E−34**
200	400		**+2.78E−19**	**+1.50E−22**	**+2.31E−15**		**+2.62E−20**	**+2.16E−22**	**+5.25E−19**
	600		**+2.93E−16**	**+4.51E−20**	**+6.54E−37**		**+5.16E−18**	**+1.02E−21**	**+7.50E−36**
	1000		**+1.10E−15**	**+3.55E−20**	**+2.22E−35**		**+3.85E−15**	**+1.26E−19**	**+2.74E−33**
400	600		**+2.45E−17**	**+1.19E−20**	**+1.13E−32**		**+3.86E−20**	**+1.33E−23**	**+4.82E−30**
	800		**+3.87E−15**	**+3.83E−19**	**+2.18E−29**		**+1.60E−17**	**+5.93E−22**	**+1.22E−36**
	1000		**+2.00E−14**	**+9.55E−19**	**+2.16E−31**		**+1.47E−17**	**+9.32E−22**	**+3.79E−35**
100	400	n400mosA(358)	**+2.03E−31**	**+3.01E−40**	**+4.14E−13**	n400mosB(364)	**+1.84E−30**	**+2.67E−35**	**+2.33E−18**
	600		**+8.13E−25**	**+5.50E−29**	**+2.34E−39**		**+3.36E−23**	**+1.69E−27**	**+4.72E−42**
	1000		**+7.66E−23**	**+8.65E−29**	**+9.92E−28**		**+5.98E−25**	**+2.27E−29**	**+2.51E−48**
200	400		**+5.40E−28**	**+8.03E−32**	**+3.00E−20**		**+6.81E−31**	**+2.86E−33**	**+3.79E−31**
	600		**+6.77E−25**	**+6.84E−28**	**+8.12E−49**		**+1.45E−29**	**+1.20E−32**	**+7.06E−54**
	1000		**+6.89E−24**	**+9.43E−28**	**+1.49E−46**		**+1.90E−25**	**+2.77E−29**	**+4.50E−42**
400	600		**+2.79E−27**	**+1.44E−29**	**+1.10E−46**		**+1.29E−31**	**+6.77E−34**	**+3.14E−48**
	800		**+2.88E−28**	**+4.27E−31**	**+1.15E−51**		**+3.69E−30**	**+1.06E−32**	**+1.46E−46**
	1000		**+8.10E−24**	**+2.98E−27**	**+3.87E−51**		**+4.30E−24**	**+1.12E−27**	**+7.09E−51**
100	400	n500mosA(453)	**+2.68E−39**	**+1.04E−51**	**+2.45E−13**	n500mosB(454)	**+7.11E−36**	**+5.23E−45**	**+2.82E−21**
	600		**+4.53E−34**	**+4.75E−41**	**+1.53E−43**		**+2.89E−38**	**+1.94E−40**	**+3.53E−52**
	1000		**+5.54E−31**	**+6.65E−36**	**+5.57E−50**		**+6.59E−30**	**+5.86E−41**	**+1.55E−40**
200	400		**+1.30E−44**	**+1.66E−53**	**+1.03E−26**		**+1.86E−42**	**+2.03E−58**	**+6.76E−27**
	600		**+2.54E−37**	**+3.26E−42**	**+1.23E−50**		**+5.34E−33**	**+4.26E−37**	**+1.96E−58**
	1000		**+7.00E−32**	**+5.01E−36**	**+2.85E−55**		**+4.93E−32**	**+4.41E−37**	**+3.36E−74**
400	600		**+2.97E−37**	**+2.28E−43**	**+1.26E−52**		**+1.49E−46**	**+7.05E−48**	**+1.17E−59**
	800		**+7.93E−42**	**+4.57E−43**	**+9.33E−55**		**+1.35E−37**	**+5.70E−41**	**+4.83E−64**
	1000		**+7.40E−35**	**+3.62E−39**	**+1.05E−66**		**+6.05E−33**	**+7.04E−37**	**+4.64E−61**

load and, therefore, encourages selecting the pickup nodes with large number of commodities to reduce the route length. These outcomes also show that the proposed MA can select required pickup nodes and arrange visiting order to resolve the SPDP effectively.

5.2. Performance comparison

The second experiment examines the performance of the proposed MA in comparison with GA and TS. Tables 4 and 5 compare the experimental results from the three test algorithms on ten SPDP instances with different values of capacity Q and gain γ. The results show that, in general, MA outperforms GA and TS, while TS outperforms GA. In addition, MA obtains the shortest routes on 86 instances and TS does on 4 out of 90 instances. Table 6 further presents the results of one-tailed paired t-test on the route lengths obtained from the test algorithms. With confidence level $\alpha = 0.05$, the t-test results demonstrate that MA achieves significantly shorter routes than GA on all instances and than TS on 83 out of 90 instances; the difference between MA and TS are insignificant on the remaining seven instances. These outcomes validate the effectiveness of MA on the SPDP. The comparative results, moreover, confirm the advantage of the modified 2-opt operator in improving MA upon route length.

Next, we look into the anytime behavior of the three test algorithms. The experiments are conducted on Intel core i7-920 machines. Due to space limitation, this paper presents only the results on n100mosA(91) and n500mosA(453). Fig. 8 depicts the progress of fitness value against the number of generations and running time. The figure shows that MA and TS converge much faster than GA does in terms of generations and running time. This advantage of MA over GA demonstrates that the modified 2-opt operator can substantially increase the efficiency of MA. Further, MA converges faster in terms of generations but slower than TS in terms of running time on n100mosA(91). This reversal of outcome reveals the influence of additional computation cost of local search on the efficiency of MA. However, as the problem size increases (e.g., on n500mosA(453)), MA leads to faster convergence than TS over both generations and running time. In the light of superior solution quality and fast convergence, the

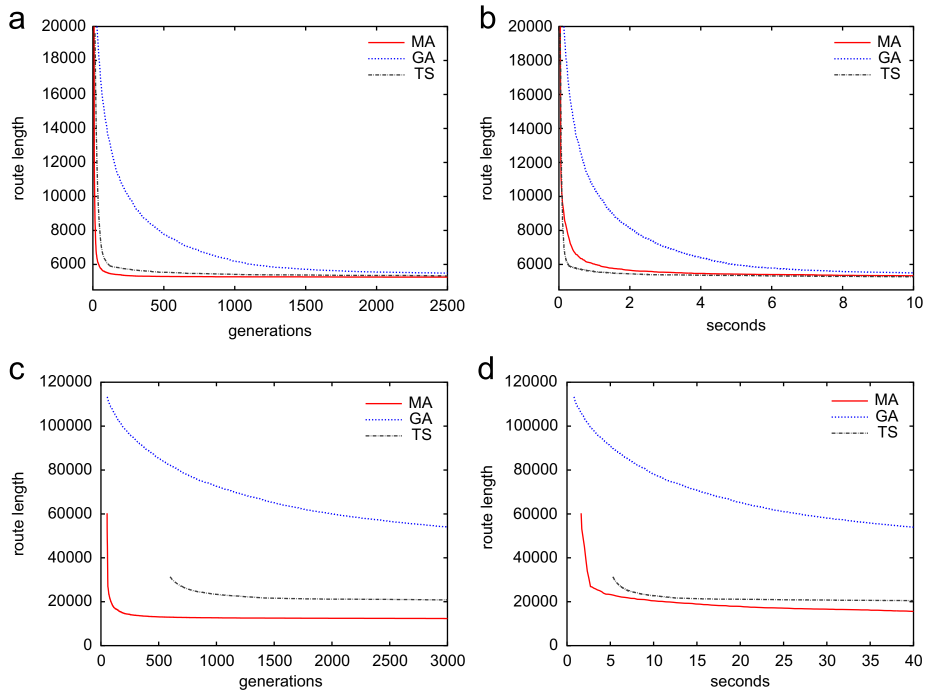

Fig. 8. Anytime behavior over generations (left) and running time (right) for MA, GA, and TS on n100mosA(91) and n500mosA(453) with $Q=1000$. (a) n100mosA(91) with $\gamma=200$. (b) n100mosA(91) with $\gamma=200$. (c) n500mosA(453) with $\gamma=400$. (d) n500mosA(453) with $\gamma=400$.

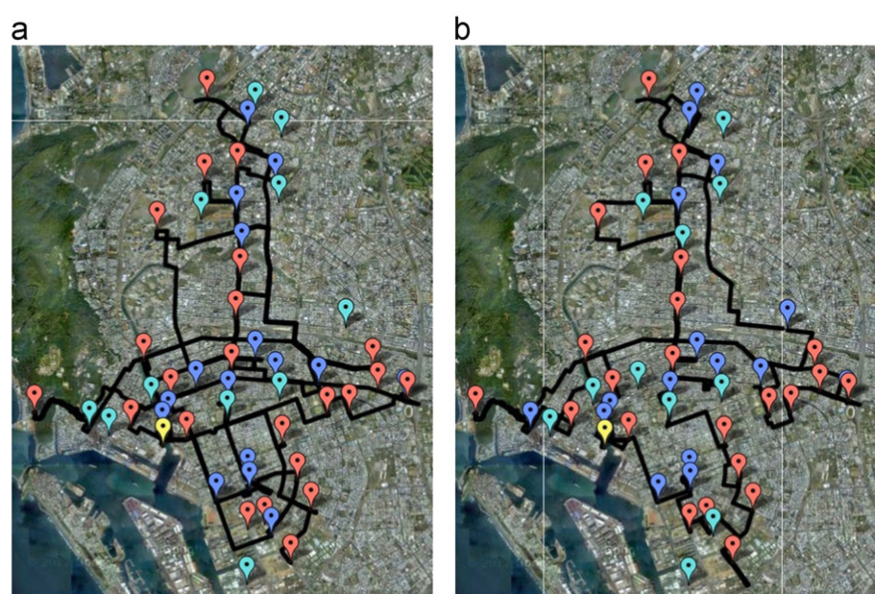

Fig. 9. Routes obtained from MA for bike distribution with $Q=10$ and 90 (yellow: depot, red: delivery node, blue: selected pickup node, cyan: omitted pickup nodes). (a) $Q=10$. (b) $Q=90$. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

merit of 2-opt local search in enhancement still outweighs its increased cost in running time for the SPDP.

5.3. Real-world application

In this study, we further apply the proposed MA to deal with a real-world SPDP in logistics—bikes distribution for the public bike-rental service of Kaohsiung city in Taiwan. The public bike-rental service has 51 rental stations around the Kaohsiung city, among which the depot is the staff-assisted station. The bikes distribution aims to transport bikes to supply the demand for bikes of some rental stations (delivery nodes) from others (pickup nodes), which corresponds exactly with the SPDP. The demand d_i of a station (node) for bikes is determined by the differential between the number of bikes on one day and the average number of bikes over two months at the station.

Fig. 9 plots the routes obtained from MA for transporting bikes using vehicles with capacity $Q=10$ and 90. The routes pass through all delivery nodes and some pickup nodes, while a certain number of pickup nodes are omitted from the route for shorter length. These preferable results confirm the advantage of the SPDP formulation and the effectiveness of the proposed MA in selecting pickup nodes and arranging the visiting order. For low vehicle capacity $Q=10$, a route occasionally needs to detour for a delivery node due to excess of vehicle load. As capacity Q increases, the constraint on the vehicle load is relaxed. As Fig. 9 shows, such flexibility can reduce

Table 7
Average route lengths and average numbers (after slash marks) of selected pickup nodes obtained from MA for the Kaohsiung city public bike distribution.

Q	Route length (km)
10	90.29/17.50
30	52.69/16.10
60	51.00/16.10
90	50.50/16.17

detour and contribute to shorter routes. The results in Table 7 further validate the effect that the average route lengths decrease with the increase in vehicle capacity.

6. Conclusions

This study presents a novel PDP variant—the SPDP—which seeks for a minimum-cost route for a vehicle to gain commodities from *some* pickup nodes and supply the demand of *all* delivery nodes, subject to the constraint on the vehicle load. There exist two major differences between the PDP and the SPDP:

- The requirement of visiting all pickup nodes is relaxed by allowing some pickup nodes omitted in the SPDP.
- The SPDP imposes a constraint that the vehicle load should be greater than zero and lower than the capacity at every node.

This study proves that the SPDP is an NP-hard problem. To tackle the problem, we proposed an MA based on GA and local search. In the MA, the modified order-based representation enables indication of varying number of selected pickup nodes with fixed chromosome length. The fitness function takes the feasibility of solutions into account. A modified 2-opt operator is devised to improve the arrangement of visiting order without destroying the route feasibility.

A series of experiments was conducted to examine the performance of the proposed MA on the SPDP. The empirical analysis shows that the selection feature of the SPDP does lead to shorter routes, which is very useful for the real-world applications that focus on supplying all demands with the commodities gathered from some nodes. Furthermore, the comparative results validate that the proposed MA achieves significantly shorter routes than TS and GA. The results also confirm the advantage of local search in improving the search ability and convergence efficiency of the MA. In addition, this study presents a real-world logistic application of the proposed method. These preferable outcomes show the utility of the SPDP and validate the effectiveness of the proposed MA in solving it.

Future work may further consider different aspects of the SPDP, for example, multi-vehicle or multi-objective SPDP. The uncertainty and dynamics will also be important topics of future work. Moreover, enhancing the search ability of the EA or the local search operator will improve the performance of the MA on the SPDP.

Acknowledgments

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

Ai, T., Kachitvichyanukul, V., 2009. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. Computers & Operations Research 36 (5), 1693–1702.
Anily, S., Bramel, J., 1999. Approximation algorithms for the capacitated traveling salesman problem with pickups and deliveries. Naval Research Logistics 46 (6), 654–670.
Anily, S., Gendreau, M., Laporte, G., 2011. The preemptive swapping problem on a tree. Networks 58 (2), 83–94.
Anily, S., Hassin, R., 1992. The swapping problem. Networks 22 (4), 419–433.
Bäck, T., Fogel, D.B., Michalewicz, Z., 1997. Handbook of Evolutionary Computation. Institute of Physics Publishing and Oxford University Press.
Baldacci, R., Bartolini, E., Mingozzi, A., 2011. An exact algorithm for the pickup and delivery problem with time windows. Operations Research 59 (2), 414–426.
Berbeglia, G., Cordeau, J.F., Gribkovskaia, I., Laporte, G., 2007. Static pickup and delivery problems: a classification scheme and survey. TOP 15, 1–31.
Berbeglia, G., Hahn, G., 2009. Counting feasible solutions of the traveling salesman problem with pickups and deliveries is #p-complete. Discrete Applied Mathematics 157 (11), 2541–2547.
Bordenave, C., Gendreau, M., Laporte, G., 2009. A branch-and-cut algorithm for the nonpreemptive swapping problem. Naval Research Logistics 56 (5), 478–486.
Bordenave, C., Gendreau, M., Laporte, G., 2010. Heuristics for the mixed swapping problem. Computers & Operations Research 37 (1), 108–114.
Çatay, B., 2010. A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery. Expert Systems with Applications 37 (10), 6809–6817.
Chalasani, P., Motwani, R., 1999. Approximating capacitated routing and delivery problems. SIAM Journal on Computing 28 (6), 2133–2149.
Cheung, B.-S., Choy, K., Li, C.-L., Shi, W., Tang, J., 2008. Dynamic routing model and solution methods for fleet management with mobile technologies. International Journal of Production Economics 113 (2), 694–705.
Cordeau, J.-F., Dell'Amico, M., Iori, M., 2010a. Branch-and-cut for the pickup and delivery traveling salesman problem with FIFO loading. Computers & Operations Research 37 (5), 970–980.
Cordeau, J.-F., Iori, M., Laporte, G., Salazar-González, J., 2010b. A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with LIFO loading. Networks 55 (1), 46–59.
Cordeau, J.-F., Laporte, G., Ropke, S., 2008. Recent models and algorithms for one-to-one pickup and delivery problems. In: Golden, B., Raghavan, S., Wasil, E. (Eds.), The Vehicle Routing Problem: Latest Advances and New Challenges, Operations Research/Computer Science Interfaces Series, vol. 43. Springer, pp. 327–357.
Cortés, C., Matamala, M., Contardo, C., 2010. The pickup and delivery problem with transfers: formulation and a branch-and-cut solution method. European Journal of Operational Research 200 (3), 711–724.
Davis, L., 1985. Applying adaptive algorithms to epistatic domains. In: Proceedings of the 9th International Joint Conference on Artificial Intelligence. Morgan Kaufmann, pp. 162–164.
Deb, K., 2000. An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering 186 (2–4), 311–338.
Duhamel, C., Potvin, J., Rousseau, J., 1997. A tabu search heuristic for the vehicle routing problem with backhauls and time windows. Transportation Science 31 (1), 49–59.
Dumitrescu, I., Ropke, S., Cordeau, J.-F., Laporte, G., 2010. The traveling salesman problem with pickup and delivery: polyhedral results and a branch-and-cut algorithm. Mathematical Programming 121, 269–305.
Eiben, A.E., Smith, J.E., 2003. Introduction to Evolutionary Computing. Natural Computing. Springer.
Erdoğan, G., Cordeau, J.-F., Laporte, G., 2009. The pickup and delivery traveling salesman problem with first-in–first-out loading. Computers & Operations Research 36 (6), 1800–1808.
Erdoğan, G., Cordeau, J.-F., Laporte, G., 2010. A branch-and-cut algorithm for solving the non-preemptive capacitated swapping problem. Discrete Applied Mathematics 158 (15), 1599–1614.
Falcon, R., Li, X., Nayak, A., Stojmenovic, I., 2010. 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 (CEC2010), pp. 4326–4333.
Garcia-Najera, A., 2012. The vehicle routing problem with backhauls: a multi-objective evolutionary approach. In: Hao, J.-K., Middendorf, M. (Eds.), Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science, vol. 7245. Springer, pp. 255–266.
Gendreau, M., Hertz, A., Laporte, G., 1996. The traveling salesman problem with backhauls. Computers & Operations Research 23 (5), 501–508.
Gendreau, M., Laporte, G., Vigo, D., 1999. Heuristics for the traveling salesman problem with pickup and delivery. Computers & Operations Research 26 (7), 699–714.
Goksal, F., Karaoglan, I., Altiparmak, F. A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. Computers & Industrial Engineering, http://dx.doi.org/10.1016/j.cie.2012.01.00, in press.
Gonzalez-Torre, P., Adenso-Dıaz, B., Artiba, H., 2004. Environmental and reverse logistics policies in European bottling and packaging firms. International Journal of Production Economics 88 (1), 95–104.

Gribkovskaia, I., Halskau, Ø., Laporte, G., Vlček, M., 2007. General solutions to the single vehicle routing problem with pickups and deliveries. European Journal of Operational Research 180 (2), 568–584.
Gribkovskaia, I., Halskau, Ø., Myklebost, K., 2001. Models for pick-up and deliveries from depots with lasso solutions. In: Stefansson, G., Tilanus, B. (Eds.), Proceedings of the 13th Annual Conference on Logistics Research—NOFOMA 2001 Collaboration in Logistics: Connecting Islands of Technology, pp. 279–293.
Gribkovskaia, I., Laporte, G., 2008. One-to-many-to-one single vehicle pickup and delivery problems. In: Golden, B., Raghavan, S., Wasil, E. (Eds.), The Vehicle Routing Problem: Latest Advances and New Challenges, Operations Research/Computer Science Interfaces Series, vol. 43. Springer, pp. 359–377.
Gribkovskaia, I., Laporte, G., Shyshou, A., 2008. The single vehicle routing problem with deliveries and selective pickups. Computers & Operations Research 35 (9), 2908–2924.
Gutiérrez-Jarpa, G., Desaulniers, G., Laporte, G., Marianov, V., 2010. A branch-and-price algorithm for the vehicle routing problem with deliveries, selective pickups and time windows. European Journal of Operational Research 206 (2), 341–349.
Hart, W.E., Krasnogor, N., Smith, J.E. (Eds.), 2004. Recent Advances in Memetic Algorithms. Springer.
Heilporn, G., Cordeau, J.-F., Laporte, G., 2011. An integer L-shaped algorithm for the dial-a-ride problem with stochastic customer delays. Discrete Applied Mathematics 159 (9), 883–895.
Hernández-Pérez, H., Rodríguez-Martín, I., Salazar-González, J., 2009. A hybrid GRASP/VND heuristic for the one-commodity pickup-and-delivery traveling salesman problem. Computers & Operations Research 36 (5), 1639–1645.
Hernández-Pérez, H., Salazar-González, J., 2003. The one-commodity pickup-and-delivery travelling salesman problem. In: Jünger, M., Reinelt, G., Rinaldi, G. (Eds.), Combinatorial Optimization—Eureka, You Shrink!, Lecture Notes in Computer Science, vol. 2570. Springer, pp. 89–104.
Hernández-Pérez, H., Salazar-González, J., 2004a. A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery. Discrete Applied Mathematics 145 (1), 126–139.
Hernández-Pérez, H., Salazar-González, J., 2004b. Heuristics for the one-commodity pickup-and-delivery traveling salesman problem. Transportation Science 38 (2), 245–255.
Hernández-Pérez, H., Salazar-González, J., 2009. The multi-commodity one-to-one pickup-and-delivery traveling salesman problem. European Journal of Operational Research 196 (3), 987–995.
Hoff, A., Gribkovskaia, I., Laporte, G., Løkketangen, A., 2009. Lasso solution strategies for the vehicle routing problem with pickups and deliveries. European Journal of Operational Research 192 (3), 755–766.
Le, M.N., Ong, Y.S., Jin, Y., Sendhoff, B., 2009. Lamarckian memetic algorithms: local optimum and connectivity structure analysis. Memetic Computing Journal 1 (3), 175–190.
Louveaux, F., Salazar-González, J., 2009. On the one-commodity pickup-and-delivery traveling salesman problem with stochastic demands. Mathematical Programming 119 (1), 169–194.
Meuth, R., Lim, M.H., Ong, Y.S., Wunsch, D.C., 2009. A proposition on memes and meta-memes in computing for higher-order learning. Memetic Computing Journal 1 (2), 85–100.
Mutha, A., Pokharel, S., 2009. Strategic network design for reverse logistics and remanufacturing using new and old product modules. Computers & Industrial Engineering 56 (1), 334–346.
Ong, Y.S., Krasnogor, N., Ishibuchi, H., 2007. Special issue on memetic algorithms. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 37 (1), 2–5.
Ong, Y.S., Lim, M.H., Chen, X.S., 2010. Research frontier: memetic computation—past, present & future. IEEE Computational Intelligence Magazine 5 (2), 24–36.
Ong, Y.S., Lim, M.H., Zhu, N., Wong, K.W., 2006. Classification of adaptive memetic algorithms: a comparative study. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 36 (1), 141–152.
Pang, K., Xu, Z., Li, C., 2011. Ship routing problem with berthing time clash avoidance constraints. International Journal of Production Economics 131 (2), 752–762.
Parragh, S., Doerner, K., Hartl, R., 2008a. A survey on pickup and delivery problems—part i: transportation between customers and depot. Journal für Betriebswirtschaft 58, 21–51.
Parragh, S., Doerner, K., Hartl, R., 2008b. A survey on pickup and delivery problems—part ii: transportation between pickup and delivery locations. Journal für Betriebswirtschaft 58, 81–117.
Psaraftis, H., 2011. A multi-commodity, capacitated pickup and delivery problem: the single and two-vehicle cases. European Journal of Operational Research 215, 572–580.
Renaud, J., Boctor, F., Laporte, G., 2002. Perturbation heuristics for the pickup and delivery traveling salesman problem. Computers & Operations Research 29 (9), 1129–1141.
Renaud, J., Boctor, F., Ouenniche, J., 2000. A heuristic for the pickup and delivery traveling salesman problem. Computers & Operations Research 27 (9), 905–916.
Ropke, S., Cordeau, J.-F., 2009. Branch and cut and price for the pickup and delivery problem with time windows. Transportation Science 43 (3), 267–286.
Shang, J., Cuff, C., 1996. Multicriteria pickup and delivery problem with transfer opportunity. Computers & Industrial Engineering 30 (4), 631–645.
Sinha, A., Chen, Y.-P., Goldberg, D.E., 2004. Designing efficient genetic and evolutionary algorithm hybrids. In: Hart, W.E., Krasnogor, N., Smith, J. (Eds.), Recent Advances in Memetic Algorithms, Studies in Fuzziness and Soft Computing, vol. 166. Springer, pp. 259–288.
Subramanian, A., Drummond, L., Bentes, C., Ochi, L., Farias, R., 2010. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. Computers & Operations Research 37 (11), 1899–1911.
Subramanian, A., Uchoa, E., Pessoa, A., Ochi, L., 2011. Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery. Operations Research Letters 39, 338–341.
Toth, P., Vigo, D., 1997. An exact algorithm for the vehicle routing problem with backhauls. Transportation Science 31 (4), 372–385.
Tu, D., Guo, S., Qin, H., Oon, W., Lim, A., 2010. The tree representation of feasible solutions for the TSP with pickup and delivery and LIFO loading. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI2010), pp. 191–196.
Wang, H., Chen, Y. A coevolutionary algorithm for the flexible delivery and pickup problem with time windows. International Journal of Production Economics, http://dx.doi.org/10.1016/j.ijpe.2012.04.011, in press.
Zhang, R., Yun, W., Moon, I., 2009. A reactive tabu search algorithm for the multi-depot container truck transportation problem. Transportation Research Part E: Logistics and Transportation Review 45 (6), 904–914.
Zhang, R., Yun, W., Moon, I., 2011. Modeling and optimization of a container drayage problem with resource constraints. International Journal of Production Economics 133 (1), 351–359.
Zhao, F., Li, S., Sun, J., Mei, D., 2009. Genetic algorithm for the one-commodity pickup-and-delivery traveling salesman problem. Computers & Industrial Engineering 56 (4), 1642–1648.