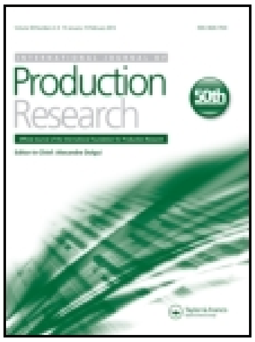

International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: www.tandfonline.com/journals/tprs20

Solving the location-routing problem with simultaneous pickup and delivery by simulated annealing

Vincent F. Yu & Shin-Yu Lin

To cite this article: Vincent F. Yu & Shin-Yu Lin (2016) Solving the location-routing problem with simultaneous pickup and delivery by simulated annealing, International Journal of Production Research, 54:2, 526-549, DOI: 10.1080/00207543.2015.1085655

To link to this article: https://doi.org/10.1080/00207543.2015.1085655

Published online: 21 Sep 2015.

International Journal of Production Research, 2016
Vol. 54, No. 2, 526–549, http://dx.doi.org/10.1080/00207543.2015.1085655

Solving the location-routing problem with simultaneous pickup and delivery by simulated annealing

Vincent F. Yu* and Shin-Yu Lin

Department of Industrial Management, National Taiwan University of Science and Technology, Taipei, Taiwan

(*Received 26 September 2014; accepted 16 August 2015*)

The study concerns the location-routing problem with simultaneous pickup and delivery (LRPSPD) in which the pickup and delivery take place at the same time for each customer. The goal is to determine the facility locations and vehicle routes in order to minimise the total system cost as a sum of facility opening cost, vehicle fixed cost and vehicle travel cost. A simulated annealing (SA) heuristic is proposed for the problem and extensive computational experiments are conducted. The results show that the proposed SA effectively solves LRPSPD and outperforms existing exact approaches in terms of solution quality.

Keywords: location-routing problem; reverse logistics; simulated annealing; simultaneous pickup and delivery

1. Introduction

In the last few decades, almost every successful business has devoted a great deal of attention to supply chain management to optimise and manage its logistics system. Two of the important and cost-effective operational issues are the determination of depots' locations and the optimisation of vehicles routes. For this, Jacobsen and Madsen (1980) and Madsen (1983) first introduced the location-routing problem (LRP) for a newspaper delivery system. The LRP is a combination of the vehicle routing problem (VRP) and the facility location problem (FLP) which simultaneously determines the locations of depots and distribution routes of vehicles to satisfy every customer's demand while minimising transportation costs, depot opening costs and vehicle fixed costs.

The LRP has been successfully applied to many practical problems in a wide variety of fields, such as drink distribution (Watson-Gandy and Dohrn 1973), bill delivery (Lin, Chow, and Chen 2002), military equipment allocation (Murty and Djang 1999) and retailing (Aksen and Altinkemer 2008), to name a few. And the LRP under certain capacity constraints of both depots and vehicles becomes the capacitated location-routing problem (CLRP). However, both LRP and CLRP fail to consider the reverse flow of the supply chain, i.e. the reverse logistics, which is regarded as a second chance to gain profit (Andel 1997). The location-routing problem with simultaneous pickup and delivery (LRPSPD), a variant of the CLRP, is therefore introduced by Karaoglan et al. (2011). Similarly, to the vehicle routing problem with simultaneous pickup and delivery (VRPSPD), in the LRPSPD, vehicles not only deliver goods to customers, but also pick up goods from customers.

With the popularity of reverse logistics, the VRPSPD has been widely used in practice, such as book distribution (Min 1989), laundry service in hotels (Çatay 2010) and the soft drink industry (Dethloff 2001). For the same reason, LRPSPD also has many applications, such as grocery store chain (Karaoglan et al. 2012) and beverage industry (Karaoglan et al. 2011). Recently, after-sales services attract much attention from customers, especially in the consumer electronics industry. Thus, when building a logistics network to distribute new products, manufacturers may need to consider collecting broken products back to depots for repair and returning the fixed products to customers in order to increase customer satisfaction and loyalty.

LRPSPD can be defined as follows. Given a set of customers with known locations and pickup and delivery demands, and potential locations of depots each with a fixed capacity, one can determine the locations of open depots and vehicle routes from depots to customers to minimise the sum of transportation costs, depots' opening costs and vehicles' fixed costs, while satisfying customers' pickup and delivery demands. Each customer is assigned to a depot and served by a vehicle exactly once. All vehicles should start from and end at the same depot. Since CLRP can be considered as a special case of LRPSPD which is an NP-hard problem (Laporte, Nobert, and Arpin 1986), the LRPSPD

*Corresponding author. Email: vincent@mail.ntust.edu.tw

also belongs to the class of NP-hard problems. Therefore, it is unlikely to obtain the optimal solution to large-scale LRPSPD instances within a reasonable amount of time, and thus meta-heuristics are effective and viable approaches for solving these problems. This study proposes a simulated annealing (SA) heuristic with a special solution representation scheme for solving the LRPSPD and test the proposed SA algorithm on four sets of benchmark LRPSPD instances generated by Karaoglan et al. (2011). Then, we generate four new sets of large-scale LRPSPD benchmark problems. All these benchmark problems are solved with the LP/MIP solver CPLEX 12.1 to obtain upper and lower bounds as the basis for further comparison with the existing approaches and testing the performance of our algorithm. Computation results show that the proposed SA outperforms existing approaches and is capable of solving large-scale problems in many real-world applications.

The remainder of this paper is organised as follows. Section 2 reviews relevant literature and provides the mathematical formulation of LRPSPD. Section 3 introduces the proposed SA heuristic for solving the LRPSPD. Section 4 reports the results of the computational study. The results obtained by the proposed heuristic are compared to those obtained by CPLEX and previous researches. Finally, Section 5 presents conclusions and suggestions for future research.

2. Literature review and mathematical formulation

2.1 *Literature review*

As LRPSPD possesses the properties of both VRPSPD and CLRP (Karaoglan et al. 2011), we first discuss VRPSPD, where the customers with both pickup and delivery demands are served by a single depot. Then, we discuss CLRP, where customers have only one type of demand and may be served by any one of the potential depots. Next, we discuss the LRPSPD in which customers have both pickup and delivery demands and may be served by one of the potential depots. Lastly, we examine the SA heuristic and its applications.

Min (1989) introduced the VRPSPD to model a book distribution network among libraries. He divided the problem into three stages and solved a small-scale book distribution problem. First, customers are grouped into several clusters with their pickup or delivery size under vehicle capacity limits. Second, a vehicle is assigned to each cluster. Lastly, the order of each customer in every cluster is determined. Dethloff (2001) proposed an insertion-based heuristic for solving the VRPSPD with the approach of clustering first and routing second. The insertions are done based on several factors, such as travel distance, residual capacity, etc. Wassan, Wassan, and Nagy (2007) solved the vehicle routing problem with backhauling by first applying the Clarke and Wright's saving algorithm (1964) to construct an initial solution, then using a reactive tabu search (TS) algorithm, originally developed by Osman and Wassan (2002), to improve the solution. The improving procedure of the reactive TS indicates that shifts and swaps are useful in finding good neighbourhood solutions.

As time went on, many meta-heuristics for the VRPSPD were proposed in the literature. Gajpal and Abad (2009) proposed the ant colony system algorithm and Çatay (2010) conducted a saving-based ant algorithm using a saving function that considers customer distances. Subramanian et al. (2011) proposed a branch-and-cut (BC) algorithm, which is implemented using the CPLEX library with lazy cuts to solve large-scale problems. Goksal, Karaoglan, and Altiparmak (2013) offered a hybrid discrete particle swarm optimisation with variable neighbourhood descent.

The CLRP consists of minimising total distribution network costs – that is, to determine which potential depots should be opened, how many vehicles should be operated and how should the operated vehicles serve all customers under routing and capacity constraints. Tuzun and Burke (1999) proposed a two-phase TS approach for solving the LRP. First, they randomly chose depots to input the route phase. Then, in the second phase, they perform a TS to determine a good configuration of depots. Prins, Prodhon, and Wolfler Calvo (2006) introduced a greedy randomised adaptive search procedure (GRASP) which combines the concept of Clarke and Wright's savings heuristic with the path relinking method and learning processes on the depot choosing strategy. Duhamel et al. (2010) used a GRASP to search two spaces: a tour without any trip delimiter and the CLRP. Barreto (2004) solved the CLRP by a three-phase heuristic with clustering techniques that first clusters all customers under vehicle capacity constraints and then solves every cluster as a travelling salesman problem (TSP) to help determine which depots should be opened, as done in the FLP. Yu et al. (2010) proposed a special solution representation with SA for solving the CLRP and obtained good results. Nguyen, Prins, and Prodhon (2012) used an iterated local search (ILS) with multiple ways of generating an initial solution. Derbel et al. (2012) presented a genetic algorithm with some specific ILS mechanisms. Ting and Chen (2013) proposed a multiple ant colony optimisation algorithm (MACO) and solved CLRP by optimising three different subproblems, location selection, customer assignment and VRP. According to the taxonomical analysis made by Lopes et al. (2013), the MACO has great performance on solving the CLRP.

Karaoglan et al. (2011) introduced the LRPSPD and proposed an exact method solution which is a hybrid of BC and SA (BC-SA) for solving the problem. They also generated LRPSPD problem sets from two famous sets of CLRP benchmark problems to test the performance of their algorithm. Karaoglan et al. (2012) built two types of LRPSPD formulations, node based and flow based, which are detailed in Section 2.

SA was introduced by Metropolis et al. (1953). Then, it was later populated by Kirkpatrick, Gelatt, and Vecchi (1983). It is an extension of the Markov chain Monte Carlo algorithm and is motivated by the physical analogy between annealing solid metals and optimising a problem as a meta-heuristic. Recently, SA has become a popular algorithm to solve NP-hard problems due to its capability of escaping from the local optima. SA has been applied to many types of routing and scheduling problems, such as TSP (Jeong and Kim 1991), CLRP (Yu et al. 2010), allocation problems (Sofianopoulou 1992), flow shops (Osman and Potts 2003) and job rotation scheduling (Seçkiner and Kurt 2007), to name a few. Therefore, we propose an SA to solve the LRPSPD with the hope of deriving competitive results.

2.2 *Mathematical formulation*

Karaoglan et al. (2011, 2012) defined two mathematical models for the LRPSPD, node based and flow based. This study adopts the flow-based formulation because computational results in the previous studies have shown that flow-based formulation has the potential to obtain better solutions with shorter computational time. In the LRPSPD, each vehicle can be operated one time and only once, and it must return to the depot from which it started. Total pickup or delivery demands of the customers assigned to each depot cannot exceed the depot capacity, and each customer can only be served once. The total vehicle load capacity constraint should not be violated.

Let $G = (N, A)$ be a directed network where $N = N_0 \cup N_c$, N_0 represents the nodes of potential depots, N_c denotes the nodes of customers and A is a set of arcs $A = \{(i, j): i, j \in N\}$. Let c_{ij} represent the distance of each arc (i, j) and let the triangular inequality $c_{id} + c_{dj} \geq c_{ij}$ be satisfied. Each potential depot $k \in N_0$ has capacity CD_k and fixed cost FD_k. Each vehicle has capacity CV and fixed cost FV. For each customer $i \in N_c$, let p_i be its pickup demand and d_i be its delivery demand, with $0 \leq d_i, p_i \leq CV$. Let U_{ij} be the remaining delivery demands after leaving node i if a vehicle travels from node i to node j, for every $i, j \in N$. Otherwise U_{ij} is 0. Similarly, let V_{ij} denote the cumulated pickup up demands up to node i if a vehicle travels from node i to node j, for every $i, j \in N$. Otherwise V_{ij} is 0. The objective of the LRPSPD is to minimise the total distribution cost, including depot opening costs and fixed costs and variable costs of vehicles, while satisfying all customer demands.

Define binary variable x_{ij}, y_k, z_{ik} as follows. $x_{ij} = 1$ *iff* a vehicle travels from node i to node j, $\forall i, j \in N$. $y_k = 1$ *iff* a depot k is opened. $z_{ik} = 1$ *iff* customer i is assigned to depot k. Karaoglan et al. (2011) gave the following mathematical model for LRPSPD.

$$\min z = \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} + \sum_{k \in N_0} FD_k y_k + \sum_{k \in N_0} \sum_{i \in N_c} FV x_{ki} \tag{1}$$

subject to

$$\sum_{j \in N} x_{ij} = 1, \quad \forall i \in N_c \tag{2}$$

$$\sum_{j \in N} x_{ji} = \sum_{j \in N} x_{ij}, \quad \forall i \in N \tag{3}$$

$$\sum_{j \in N} U_{ji} - \sum_{j \in N} U_{ij} = d_i, \quad \forall i \in N_c \tag{4}$$

$$\sum_{j \in N} V_{ij} - \sum_{j \in N} V_{ji} = p_i, \quad \forall i \in N_c \tag{5}$$

$$U_{ij} + V_{ij} \leq CV x_{ij}, \quad \forall i, j \in N, i \notin j \tag{6}$$

$$\sum_{j \in N_c} U_{kj} = \sum_{j \in N_c} z_{jk} d_j, \quad \forall k \in N_0 \tag{7}$$

$$\sum_{j \in N_c} U_{jk} = 0, \quad \forall k \in N_0 \tag{8}$$

$$\sum_{j \in N_c} V_{jk} = \sum_{j \in N_c} z_{jk} p_j, \quad \forall k \in N_0 \tag{9}$$

$$\sum_{j \in N_c} V_{kj} = 0, \quad \forall k \in N_0 \tag{10}$$

$$U_{ij} \leq (CV - d_i)x_{ij}, \quad \forall i \in N_c, \quad \forall j \in N \tag{11}$$

$$V_{ij} \leq (CV - p_i)x_{ij}, \quad \forall i \in N, \quad \forall j \in N_c \tag{12}$$

$$U_{ij} \geq d_j x_{ij}, \quad \forall i \in N, \quad \forall j \in N_c \tag{13}$$

$$V_{ij} \geq p_i x_{ij}, \quad \forall i \in N_c, \quad \forall j \in N \tag{14}$$

$$\sum_{k \in N_0} z_{ik} = 1, \quad \forall i \in N_c \tag{15}$$

$$\sum_{i \in N_c} d_i z_{ik} \leq CD_k y_k, \quad \forall k \in N_0 \tag{16}$$

$$\sum_{i \in N_c} p_i z_{ik} \leq CD_k y_k, \quad \forall k \in N_0 \tag{17}$$

$$x_{ik} \leq z_{ik}, \quad \forall i \in N_c, \quad \forall k \in N_0 \tag{18}$$

$$x_{ki} \leq z_{ik}, \quad \forall i \in N_c, \quad \forall k \in N_0 \tag{19}$$

$$x_{ij} + z_{ik} + \sum_{m \in N_0, m \neq k} z_{jm} \leq 2, \quad \forall i,j \in N_c, i \neq j, \quad \forall k \in N_0 \tag{20}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i,j \in N \tag{21}$$

$$z_{ik} \in \{0, 1\}, \quad \forall i \in N_c, \quad \forall k \in N_0 \tag{22}$$

$$y_k \in \{0, 1\}, \quad \forall k \in N_0 \tag{23}$$

$$U_{ij}, V_{ij} \geq 0, \quad \forall i,j \in N \tag{24}$$

The objective function (1) is the total cost including transportation costs, depot opening costs and vehicle fixed costs. Constraint (2) guarantees that each customer is served once and only once. Constraint (3) ensures that the number of arcs entering and leaving each node is the same. Constraints (4) and (5) eliminate any subtour and ensure every customer's pickup and delivery demands are satisfied. Constraint (6) limits the total load of any arc to not exceed the vehicle capacity. Constraint (7) ensures that the total delivery demands of all customers assigned to the corresponding depot are equal to the same depot's total delivery load. Constraint (8) guarantees that all delivery demands of each depot should be distributed to its assigned customers; therefore, the total delivery demands of each depot must be zero, while all vehicles return to their depot. Constraint (9) similarly ensures that the total pickup demands of all customers assigned to the corresponding depot are equal to the same depot total pickup load. Constraint (10) guarantees that all pickup demands of each depot should be distributed to its assigned customers; therefore, the total pickup demands of each depot must be zero while all vehicles are set up from their depot. Constraints (11)–(14) are bounding constraints for additional variables. Constraint (15) guarantees that each customer is served once by one depot only. Constraint (16) ensures that the total delivery demands of all customers assigned to each depot must not exceed the depot's capacity limit, while constraint (17) guarantees that the total pickup demands of all customers assigned to each depot must not exceed the depot's capacity limit. Constraints (18) and (19) ensure that each node connected to a depot must be first assigned to the depot. Constraint (20) makes sure that two consecutive customers on a route are assigned to the same depot. Constraints (18)–(20) together forbid infeasible routes that do not start and end at the same depot. Constraints (21)–(24) define some integrality constraints for the decision variables.

3. SA heuristic of the LRPSPD

An analogy for using SA to solve the LRPSPD is annealing solid metals, whereby the procedure is separated into heating the metallic parts to the set-up temperature and cooling them off at a specific slow rate. The process of heating allows atoms to become brisk and bounce around their original position, while slowly cooling gives atoms more chances to find configurations with lower energy. Therefore, SA could avoid being trapped into the local optima.

SA starts from an initial solution and keeps generating neighbourhood solutions of the current solution. If a new neighbourhood solution is better than the current solution, then the new neighbourhood solution is accepted and replaces the current solution. The algorithm may also accept a non-improving solution with certain probability related to the input parameters, which we introduce in the following subsections.

3.1 *Solution representation*

A solution can be represented by a string of numbers consisting of N_0 potential depots denoted by the set $\{1, 2, \ldots, N_0\}$, N_c customers denoted by the set $\{N_0 + 1, \ldots, N_0 + N_c\}$ and N_{dummy} dummy zeros. The dummy zeros are used to terminate routes. The value of N_{dummy} is calculated as $\left\lceil \max\left(\sum_{i \in N_c} \frac{d_i}{CV}, \sum_{i \in N_c} \frac{p_i}{CV}\right) \right\rceil$, where d_i and p_i are the respective delivery and pickup demands of customer i, with $\lceil x \rceil$ representing the smallest integer that is larger than or equal to x. Note that the first number in the string must be a depot.

Using this special solution representation, the first depot serves the customers between itself and the next depot in the string, according to the order of numbers in the sequence. Customers are added to the current route one by one, provided that the capacity of the vehicle is not violated. If the next number in the string is a dummy zero, then the current route is terminated and a new route is generated and assigned to the current depot. Therefore, using dummy zeros in the solution representation, a route may be randomly terminated, even though the vehicle has not reached its capacity. As a result, the search space is larger and better solutions may be found. Note that the depot's capacity constraint may be violated when assigning customers to vehicles. We take a small-scale LRPSPD instance (Srivastava86-8x2_W) (Karaoglan et al. 2011) with two potential depots (N_0) and eight customers (N_c) to illustrate how a solution string is transformed into a distribution network. Table 1 gives the coordinates (X, Y) of depots and customers, capacity of depots (CD), operating cost of depots (FD), capacity of vehicles (CV), operating cost of vehicles (FV), pickup demand of customers (p) and

Table 1. Data of the LRPSPD instance, Srivastava86-8x2_W.

Depot ID	Depot coordinate		Depot capacity (*CD*)	Depot fixed cost (*FD*)	Vehicle capacity (*CV*)	Vehicle fixed cost (*FV*)
	X	*Y*				
1	117	174	1000	36	200	20
2	79	29	1000	33	200	20

Customer ID	Customer coordinate		Delivery (*d*)	Pickup (*p*)
	X	*Y*		
3	92	162	112	23
4	20	86	112	200
5	83	43	62	13
6	69	212	145	200
7	42	50	135	27
8	94	118	128	200
9	74	64	71	15
10	69	194	54	98

2	7	9	5	0	0	4	**1**	0	8	0	6	3	10	0

Figure 1. A solution representation of an LRPSPD instance.

delivery demand of customers (d). Figure 1 presents a solution string of this instance with five dummy zeros. According the solution string, depot 2 serves customers 7, 9, 5 and 4, while depot 1 serves customers 8, 6, 3 and 10. If adding the next customer violates the vehicle's capacity constraint or the next number is a dummy zero, the current route is terminated. Figure 2 depicts the distribution network corresponding to the solution string in Figure 1.

3.2 *Initial solution*

We attempt to find a good initial solution within a reasonable amount of time in the first stage. The initial solution is generated using a modified greedy heuristic. The five steps of the heuristic are as follows.

Step 1. Let $cc(i)$ represent the number of all unassigned customers whose closest depot is depot i. We choose the depot with the largest cc value. If there is a tie, then the depot with the largest capacity is selected.
Step 2. Let T be the set of unassigned customers. Assign customers in T to the depot selected in Step 1 in a non-decreasing order of the distance between the customer and the depot. Stop when T is empty or assigning the next customer to the chosen depot violates the depot's capacity constraint. Remove assigned customers from T.
Step 3. Construct a TSP route for the customers assigned to the chosen depot. The route starts from and ends at the chosen depot. We use an efficient algorithm proposed by Lin and Kernighan (1973) for generating the TSP route.
Step 4. Split the TSP route generated in Step 3 into several routes based on vehicle capacity.
Step 5. If T is not empty, return to Step 1; otherwise, terminate and encode the current solution using the special solution representation scheme described in Section 3.1.

In Step 4, the TSP route generated in Step 3 is divided into several vehicle routes based on vehicle capacity. Starting from the first customer in the TSP route, customers are assigned to a vehicle one at a time in their order of appearance in the TSP route, until adding the next customer to the current route violates the vehicle's capacity constraint. In this case, a new route is started with this customer and the process continues until all the customers are assigned to a vehicle.

The initial solution obtained by the modified greedy algorithm is encoded into the solution string according to their order of appearance in the TSP solution. Then, all closed depots and the N_{Dummy} dummy zeros are appended to finalise the solution string.

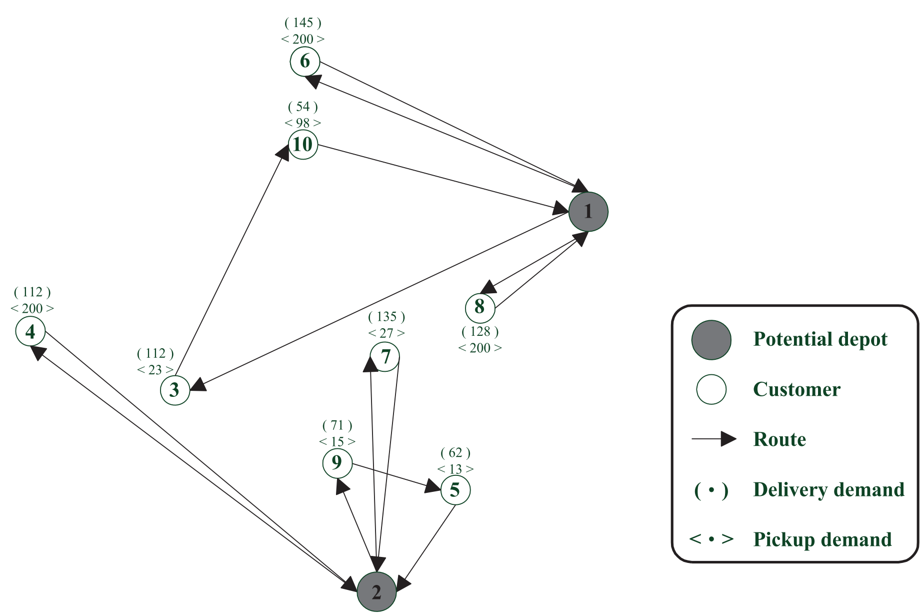

Figure 2. A visual illustration of the LRPSPD distribution network.

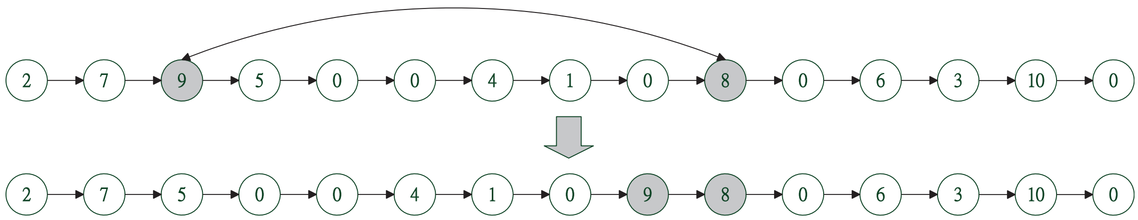

Figure 3. An example of the insertion move.

3.3 *Neighbourhood search mechanisms*

The proposed SA heuristic employees three types of local neighbourhood search mechanisms: insertion move, swap move and reverse move, defined as follows. Each move is implemented with the same probability of 1/3. Note that the first number of the solution string must remain a depot after every move.

3.3.1 *Insertion*

The insertion is executed by randomly selecting the ith and the jth positions of the solution string. Then, the number in the ith position is inserted into the position immediately before the jth position.

Four cases may occur after the insertion move. If position i is a customer and position j is a depot, then the customer in position i is served by the depot before j after insertion. If i is a depot while j is a customer, then the depot in position i serves the customers in and after position j. Customers originally assigned to the depot in position i are assigned to the depot before position i. If both i and j are customers, then the depot that originally serves the customer in position j serves the customer in position i first and then serves the customer in position j. If both i and j are depots, then the depot in position i is closed, and all of its customers are reassigned to the depot before position i. Figure 3 shows the insertion move. The chosen position i and position j are customer 9 and customer 8, respectively.

3.3.2 *Swap*

The swap move randomly chooses the ith and jth positions of the solution string. Then, the two numbers in these two positions are exchanged. Four cases may occur after the swap move. If position i is a customer and position j is a depot, then the depot in position j serves customers after customer i, while customer in position i is reassigned to the depot originally in front of depot j. The case in which position i is a depot and position j is a customer is similar. If both i and j are customers, then customer i is reassigned to the depot originally serving customer j and vice versa. If both i and j are depots, then the customers assigned to depot i are served by depot j and vice versa. Figure 4 illustrates the swap move. The chosen positions i and j are customer 9 and a dummy 0, respectively.

3.3.3 *Reverse*

The reverse move randomly selects two positions of the solution string and then reverses the substring between the two positions. Figure 5 depicts the reverse move. The chosen position i and position j are a dummy 0 and customer 5, respectively.

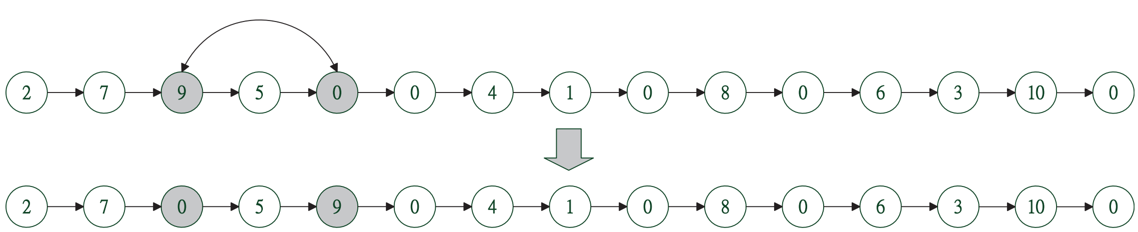

Figure 4. An example of the swap move.

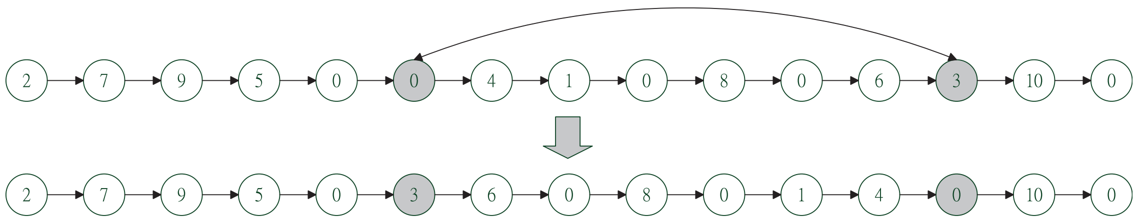

Figure 5. An example of the reverse move.

3.4 *The SA procedure*

The proposed SA heuristic uses six parameters in solving the LRPSPD: I_{iter}, N_{ni}, T_0, T_f, α and K. I_{iter} represents the total iterations the SA should repeat at a particular temperature. N_{ni} is the maximum allowable number of consecutive temperature reductions without improvement in solution value. T_0 denotes the initial temperature, while T_f is the final temperature which is one of the termination conditions. α represents the coefficient of cooling procedure and K is the Boltzmann's constant for calculating the probability of accepting a worse solution.

Figure 6 presents the pseudo-code of the proposed SA for solving LRPSPD. At the beginning of the SA procedure, the current temperature (T) is set to be the initial temperature (T_0). The procedure starts with an initial solution including N_{dummy} zeros as the current solution (X) as mentioned in Section 3.2. X_{best} and F_{best}, initially set to be X and a sufficiently large number respectively, record the current best solution and its solution value.

Each iteration at a particular temperature generates a neighbourhood solution (Y) from the current solution by the different neighbourhood search mechanisms described in Section 3.3. Let Δ be the difference between the objective function value of Y, denoted by obj(Y), and that of X, that is $\Delta = \text{obj}(Y) - \text{obj}(X)$. If $\Delta < 0$, the neighbourhood solution is better than the current solution, then Y replaces X as the new current solution; otherwise, we calculate the probability, $\exp(-\Delta/KT)$, of accepting the worse solution and then generate a random number $r \sim U(0, 1)$. If $r < \exp(-\Delta/KT)$, then we replace the current solution X with the neighbourhood solution Y. After generating neighbourhood solutions I_{iter} times at a specific temperature, we reduce the current temperature by setting $T = \alpha T$, where $0 < \alpha < 1$. The SA algorithm terminates on two conditions: the current temperature is less than or equal to the final temperature T_f or the current best solution has not been improved for N_{ni} consecutive temperature reductions.

4. Computational study

The proposed SA heuristic is implemented in Microsoft Visual C++ 2008 and run on a PC with Intel Core 2 Quad Q8200 CPU at 2.34 GHz and 1.93 GB memory under the Windows XP system.

4.1 *Test instances*

The numerical experiment is performed using eight sets of LRPSPD instances – KBAM, KBSN, KPAM, KPSN, YLBAM, YLBSN, YLPAM and YLPSN. The first four sets are benchmark instances introduced by Karaoglan et al. (2011). The number of customers in each of these instances is less than or equal to 100, so we classify these instances as small- and medium-scale problem sets. The last four sets are generated by this study and regarded as large-scale instances since they have more than 100 customers. We convert them from two sets of well-known CLRP benchmark instances created by Prins, Prodhon, and Wolfler Calvo (2004) and Barreto (2004). Two separation approaches proposed by Angelelli and Mansini (2002) and Salhi and Nagy's (1999) are applied to these instances to generate pickup demands. In all data-sets, coordinates of customer locations and depot locations are given, and the distance between locations is the Euclidean distance rounded to the fourth digit. Details of the eight sets of LRPSPD benchmark instances are given in Table 2.

Altogether, we use 200 LRPSPD test instances for the computational study. Both KBAM and KBSN have 30 instances, both KPAM and KPSN have 44 instances, both YLBAM and YLBSN have 10 instances and both YLPAM and YLPSN have 16 instances. Karaoglan et al. (2011) have tested their BC-SA algorithm on the four small- and medium-scale data-sets, KBAM, KBSN, KPAM and KPSN. Their experiments are performed on a computer equipped with Intel Xeon 3.16 GHz and 1 GB RAM. A time limit of 4 h is imposed on each instance.

```
SA for LRPSPD (I_iter , N_ni , T_0 , T_f ,α, K)
begin
    N_dummy = max(ceil (Σd_i / CV), ceil (Σp_i / CV));
    Generate the initial solution X by the TSP heuristic;
    T = T_0; N_1 = 0; X_best = X; F_best = obj(X);
    while ( T > T_f ) {
        I = 0; N_1 = N_1 + 1;
        while ( I < I_iter ) {
            Generate a random number q ~ U(0, 1);
            if (q < 1/3){ Generate a new solution Y from X by insertion move operation;}else if (q
    < 2/3) { Generate a new solution Y from X by swap move operation;}
            else { Generate a new solution Y from X by reverse move operation;}
            if ( obj(Y) < obj(X) ) { X = Y; }
            else {
              Generate a random number r ~ U(0, 1);
              Δ = obj(Y) – obj(X);
              if (r < exp(-Δ/KT )) { X = Y; }
            }
            if ( obj(X) < F_best and X is feasible ) { X_best = X; F_best = obj(X); N_1 = 0; }
            I = I + 1;
        }
        T =αT;
        if ( N_1 = N_ni) { Terminate; }
    }
end;
```

Figure 6. Pseudo-code of the proposed SA for solving LRPSPD.

Table 2. Characteristics of the eight LRPSPD data-sets.

	Source	Separation approach	Size	Type	Number of depots	Number of customers
KBAM	Karaoglan et al. (2011)	Angelelli and Mansini (2002)	30	W, Z	2–10	8–100
KBSN	Karaoglan et al. (2011)	Salhi and Nagy's (1999)	30	X, Y	2–10	8–100
KPAM	Karaoglan et al. (2011)	Angelelli and Mansini's (2004)	44	W, Z	5 or 10	20, 50, or 100
KPSN	Karaoglan et al. (2011)	Salhi and Nagy's (1999)	44	X, Y	5 or 10	20, 50, or 100
YLBAM	This study	Angelelli and Mansini's (2004)	10	W, Z	4–18	117–318
YLBSN	This study	Salhi and Nagy's (1999)	10	X, Y	4–18	117–318
YLPAM	This study	Angelelli and Mansini's (2004)	16	W, Z	5 or 10	50 or 200
YLPSN	This study	Salhi and Nagy's (1999)	16	X, Y	5 or 10	50 or 200

4.2 *Separation approaches*

Salhi and Nagy's (1999) and Angelelli and Mansini's (2002) separation approaches are used in generating all eight LRPSPD data-sets. The following presents details on the two approaches.

4.2.1 *Salhi and Nagy's separation approach*

Salhi and Nagy (1999) introduced the VRPSPD problem data-set derived from 14 VRP benchmark problems of Christofides, Mingozzi, and Toth (1979). They generated the pickup demands as follows. First, they calculate the minimum ratio, r_i, of each customer i from its coordinates (x_i and y_i) as $r_i = \min(x_i/y_i, y_i/x_i)$. Using the original delivery demand q_i, the delivery demand is calculated as $d_i = r_i q_i$. Then, the pickup demand is calculated as $p_i = q_i - d_i$. Karaoglan et al. (2011) call this type of problems as the type X problems. Another type of problems, named Y type, is generated by exchanging the pickup and delivery demands of type X problems. Two of our new LRPSPD data-sets, YLBSN and YLPSN, are generated in a similar manner.

4.2.2 *Angelelli and Mansini's separation approach*

Angelelli and Mansini (2002) proposed an approach to determine pickup and delivery demands in the VRP with time windows and simultaneous pickup and delivery. The delivery demand of customer i is equal to the original demand, that is $d_i = q_i$. The pickup demand of the corresponding customer is determined by $p_i = \lfloor (1-\beta)q_i \rfloor$ if i is an even number, whereas $p_i = \lfloor (1+\beta)q_i \rfloor$ if i is an odd number, where β is a constant and $\lfloor x \rfloor$ denotes the greatest integer that is less than or equal to x. Karaoglan et al. (2011) set β to be 0.2 for type W instances and 0.8 for type Z instances. Two of our new LRPSPD data-sets, YLBAM and YLPAM, are generated similarly.

Note that if i is an odd number, the pickup demand generated by the formula, $p_i = \lfloor (1+\beta)q_i \rfloor$, may exceed the vehicle's capacity. Therefore, in generating our new LRPSPD instances, we use $p_i = \min\{\lfloor (1+\beta)q_i \rfloor, CV\}$ to generate pickup demands when i is an odd number, where CV is the vehicle's capacity.

4.3 *Parameter setting*

In addition to the components of the algorithm, the parameters used by the algorithm also play an important role in the quality of solutions. We conduct an initial experiment with various parameter settings as follows.

- I_{iter} = 1000L, 2000L, 3000L, 4000L, 5000L, 6000L, where $L = N_0 + N_c + N_{dummy}$ is the coding length of the solution string;
- N_{ni} = 100, 200;
- T_0 = 20, 30, 40;
- T_f = 0.1, 0.01, 0.001;
- K = 1, 1/2, 1/3, … , 1/14, 1/15;
- α = 0.96, 0.97, 0.98, 0.99.

Each parameter combination is tested five times. According to the results, we find that when I_{iter} = 5000L, N_{ni} = 100, T_0 = 30, T_f = 0.1, K = 1/7, and α = 0.99, the proposed SA obtains the best results. Therefore, this parameter setting is used in the final analysis to solve all eight sets of LRPSPD instances, including small-, medium- and large-scale problems.

4.4 *Computational results*

4.4.1 *Sensitivity analysis*

For analysing the performance of the proposed SA and neighbourhood search methods, we conduct an experiment on four LRPSPD instances with various numbers of customers as presented in Table 3. The analysis aims to show how the solution converges in four different scenarios, randomly applying all three neighbourhood search methods or applying one of them alone. It can be seen from Figure 7 that using all 3 neighbourhood search methods randomly results in faster convergence and better solution.

Table 3. Performance analysis of different neighbourhood search methods.

	ALL_random		SWAP		INSERTION		REVERSE	
Instance	Sol value	Iteration	Sol value	Iteration	Sol value	Iteration	Sol value	Iteration
Prod_20_5_W_Ncoord20-5-1.txt	26468.86	103	26468.86	106	26468.86	136	26468.86	120
Prod_50_5_X_Ncoord50-5-2BIS.txt	20414.74	390	20416.38	340	20419.94	283	20415.46	328
Prod_100_10_Y_Ncoord100-10-1.txt	109704.30	381	109719.98	404	109832.13	364	109973.56	264
Prod_200_10_Z_Ncoord200-10-1.txt	286726.37	320	286748.91	409	287142.67	237	287924.95	237

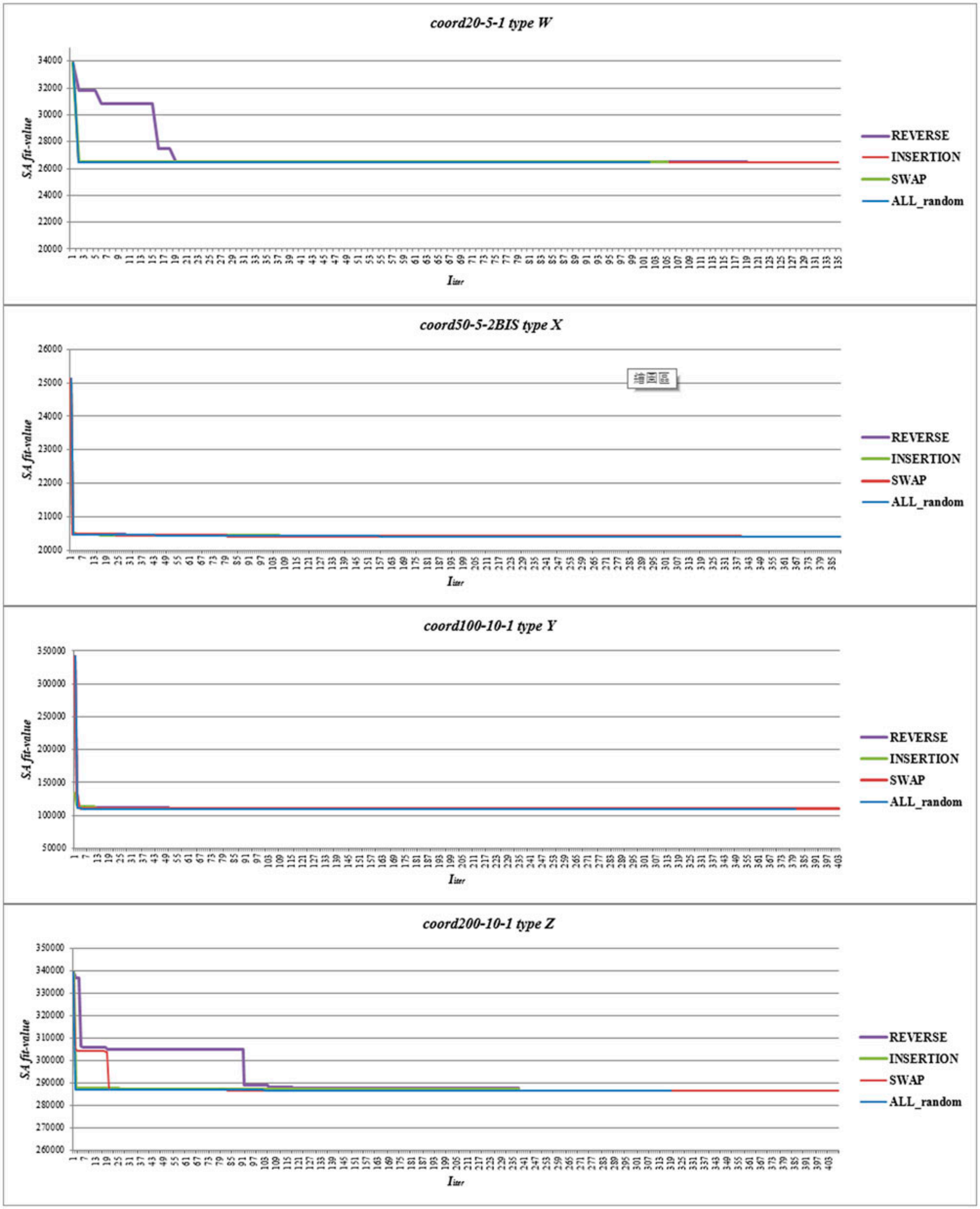

Figure 7. Comparison on the convergence of different neighbourhood search methods.

4.4.2 *Performance comparison*

We solve all LRPSPD instances in the eight data-sets by CPLEX 12.1 using the flow-based model to obtain upper bounds and lower bounds on solution value as the basis for the comparison between different approaches. Note that the lower bound is the solution to the LP relaxation and the upper bound is the best integer solution obtained by CPLEX.

4.4.2.1 *KBAM and KBSN.* Tables 4 and 5, respectively, present the computational results for the KBAM and KBSN data-sets. We compare the proposed SA to the exact approach, hybrid BC-SA, proposed by Karaoglan et al. (2011).

CPLEX obtains optimal solutions to 14 out of the 30 instances in KBAM data-set and to 18 instances among the 30 instances in KBSN. Both the hybrid BC-SA and the proposed SA can obtain these optimal solutions as well. The difference in solution quality between these two methods becomes apparent as the number of customers increases. Compared to hybrid BC-SA, using the best parameters in the final analysis, the proposed SA produces better solutions for all instances in both data-sets. In fact, 14 solutions to those instances are the new best solutions. Moreover, during the parameter analysis, our SA heuristic obtains the best solutions for all 60 instances, with 22 solutions of them being the new best solutions. The average improvement on solution quality is 1.04 and 1.00% and the average computational time is 155.02 and 134.52 s for the KBAM and KBSN data-sets, respectively.

4.4.2.2 *YLBAM and YLBSN.* We solve the two new large-scale data-sets, YLBAM and YLBSN, by CPLEX 12.1 with a time limit of 14,400 s to find an upper bound and a lower bound for each instance in both data-sets. These bounds will be used to evaluate the performance of the proposed SA heuristic. Tables 6 and 7 display the results for the YLBAM and the YLBSN data-sets, respectively. As can be seen in both tables, CPLEX fails to obtain a solution for instances with 318 customers. For the other instances, the proposed SA heuristic performs well since its solutions are well below the upper bounds. On average, solution values of SA solutions are 80.12–89.56% lower than those of CPLEX solutions. The average computational time for these large-scale problems ranges from 447.97 s for an instance with 117 customers and 14 potential depots to 16,858.20 s for an instance with 318 customers and 4 potential depots.

4.4.2.3 *KPAM and KPSN.* Tables 8 and 9, respectively, show the computational results for the KPAM and KPSN data-sets. The average gaps between the lower bounds obtained by CPLEX and the solution values obtained by the proposed SA are merely 1.04 and 0.17% for KPAM and KPSN data-sets, respectively. Meanwhile, comparing with BC-SA, using the best parameters in the final analysis, the proposed SA produces better solutions for all problems in both data-sets, with 25 solutions being the new best solutions. During the parameter analysis, our SA heuristic obtained the best solutions for all 88 instances, 66 of them are new.

The average time for solving LRPSPD by the proposed SA using the best parameter setting is about 530.82 and 401.37 s for KPAM and KPSN data-sets, respectively.

4.4.2.4 *YLPAM and YLPSN.* The next two large-scale data-sets, YLPAM and YLPSN, are also solved by executing CPLEX 12.1 for 14,400 s to find an upper bound and a lower bound for each instance. CPLEX successfully finds solution boundaries for instances with 50 customers in these data-sets. The performance of our SA heuristic is considered good since the average gap between the lower bounds and the solution values obtained by our SA is at most 1.99%. On the other hand, the proposed SA not only yields an average improvement of 5.12% in solution quality but also finds feasible solutions for those large-scale instances with 200 customers which CPLEX fails to find any feasible solution within 14,400 s. We display the computational results for the YLPAM and YLPSN data-sets in Tables 10 and 11. The average computational time ranges from 183.67 s for an instance with 50 customers and 5 potential depots to 3174.47 s for an instance with 200 customers and 10 potential depots.

In summary, we have solved eight benchmark data-sets, with a total of 200 LRPSPD instances. We compare our results with Karaoglan et al. (2011). The proposed SA obtains best solutions for all instances and 88 new best solutions for four data-sets, KBAM, KBSN, KPAM and KPSN.

Table 4. Computational results for the KBAM data-set.

		CPLEX					BC-SA				SA					SA*			
Instance	Type	LB[a]	Int Sol[b]	Nodes	Cuts	Soltime	BC-SA	Gap1[c] (%)	Gap2[d] (%)	Soltime	SA	Gap1 (%)	Gap2 (%)	Gap3[e] (%)	Soltime	SA*[f]	Gap1 (%)	Gap2 (%)	Gap3 (%)
Srivastava86-8x2	W	***873.58***	***873.58***	0	0	0.03	**873.58**	0.00	0.00	0.00	**873.58**	0.00	0.00	0.00	0.42	**873.58**	0.00	0.00	0.00
Srivastava86-8x2	Z	***806.06***	***806.06***	0	0	0.02	**806.06**	0.00	0.00	0.00	**806.06**	0.00	0.00	0.00	0.34	**806.06**	0.00	0.00	0.00
Perl83-12x2	W	***243.98***	***243.98***	0	32	0.67	**243.98**	0.00	0.00	0.57	**243.98**	0.00	0.00	0.00	0.37	**243.98**	0.00	0.00	0.00
Perl83-12x2	Z	***243.98***	***243.98***	0	34	0.81	**243.98**	0.00	0.00	0.65	**243.98**	0.00	0.00	0.00	0.35	**243.98**	0.00	0.00	0.00
Gaskell67-21x5	W	***528.42***	***528.42***	121,179	329	6808.20	**528.42**	0.00	0.00	290.18	**528.42**	0.00	0.00	0.00	13.22	**528.42**	0.00	0.00	0.00
Gaskell67-21x5	Z	***513.30***	***513.30***	58,549	229	2420.49	**513.30**	0.00	0.00	89.43	**513.30**	0.00	0.00	0.00	1.47	**513.30**	0.00	0.00	0.00
Gaskell67-22x5	W	***653.80***	***653.80***	499	96	5.41	**653.80**	0.00	0.00	3.76	**653.80**	0.00	0.00	0.00	1.23	**653.80**	0.00	0.00	0.00
Gaskell67-22x5	Z	***653.80***	***653.80***	402	103	4.42	**653.80**	0.00	0.00	2.66	**653.80**	0.00	0.00	0.00	1.94	**653.80**	0.00	0.00	0.00
Min92-27x5	W	***3142.02***	***3142.02***	6457	238	826.49	**3142.02**	0.00	0.00	19.66	**3142.02**	0.00	0.00	0.00	10.65	**3142.02**	0.00	0.00	0.00
Min92-27x5	Z	***3142.02***	***3142.02***	50,103	261	3368.70	**3142.02**	0.00	0.00	18.12	**3142.02**	0.00	0.00	0.00	0.88	**3142.02**	0.00	0.00	0.00
Gaskell67-29x5	W	***592.10***	***592.10***	19,927	142	898.16	**592.10**	0.00	0.00	6.76	**592.10**	0.00	0.00	0.00	3.34	**592.10**	0.00	0.00	0.00
Gaskell67-29x5	Z	558.58	**592.10**	128,335	354	14,400.00	**592.10**	6.00	0.00	100.91	**592.10**	6.00	0.00	0.00	4.08	**592.10**	6.00	0.00	0.00
Gaskell67-32x5_1	W	659.44	707.99	30,021	136	14,400.00	**696.38**	5.60	−1.64	5694.09	**696.38**	5.60	−1.64	0.00	28.24	**696.38**	5.60	−1.64	0.00
Gaskell67-32x5_1	Z	622.93	**643.37**	158,074	207	14,400.00	**643.37**	3.28	0.00	259.67	**643.37**	3.28	0.00	0.00	46.93	**643.37**	3.28	0.00	0.00
Gaskell67-32x5_2	W	***595.27***	***595.27***	9221	78	518.34	**595.27**	0.00	0.00	15.68	**595.27**	0.00	0.00	0.00	8.75	**595.27**	0.00	0.00	0.00
Gaskell67-32x5_2	Z	***564.33***	***564.33***	31,023	178	2723.38	**564.33**	0.00	0.00	30.90	**564.33**	0.00	0.00	0.00	6.69	**564.33**	0.00	0.00	0.00
Gaskell67-36x5	W	534.42	585.48	20,776	74	14,400.00	**540.37**	1.11	−7.70	27.20	**540.37**	1.11	−7.70	0.00	14.05	**540.37**	1.11	−7.70	0.00
Gaskell67-36x5	Z	***540.37***	***540.37***	46,329	204	6699.59	**540.37**	0.00	0.00	46.46	**540.37**	0.00	0.00	0.00	6.40	**540.37**	0.00	0.00	0.00
Ch69-50x5	W	663.70	749.69	19,056	167	14,400.00	**708.37**	6.73	−5.51	14,400.00	**708.37**	6.73	−5.51	0.00	83.89	**708.37**	6.73	−5.51	0.00
Ch69-50x5	Z	671.19	709.25	20,196	196	14,400.00	701.91	4.58	−1.03	14,400.00	700.87	4.42	−1.18	−0.15	118.82	**700.72**	4.40	−1.20	−0.17
Perl83-55x15	W	1046.68	1498.13	845	90	14,400.00	1330.85	27.15	−11.17	14,400.00	**1327.06**	26.79	−11.42	−0.28	239.46	**1327.06**	26.79	−11.42	−0.28
Perl83-55x15	Z	1039.82	1074.82	832	136	14,400.00	1338.30	28.70	24.51	14,400.00	**1322.62**	27.20	23.06	−1.17	213.69	**1322.60**	27.20	23.05	−1.17
Ch69-75x10	W	967.81	1449.56	461	172	14,400.00	1177.65	21.68	−18.76	14,400.00	**1132.80**	17.05	−21.85	−3.81	350.96	**1132.80**	17.05	−21.85	−3.81
Ch69-75x10	Z	951.86	1424.26	251	207	14,400.00	1108.82	16.49	−22.15	14,400.00	1079.32	13.39	−24.22	−2.66	423.02	**1076.81**	13.13	−24.40	−2.89
Perl83-85x7	W	1463.58	2142.07	1584	291	14,400.00	1901.09	29.89	−11.25	14,400.00	1857.56	26.92	−13.28	−2.29	603.11	**1855.55**	26.78	−13.38	−2.40
Perl83-85x7	Z	1438.26	2158.36	800	293	14,400.00	1893.28	31.64	−12.28	14,400.00	1855.97	29.04	−14.01	−1.97	481.77	**1855.55**	29.01	−14.03	−1.99
Daskin95-88x8	W	0.00	6311.85	0	0	14,400.00	533.37	N/A	−91.55	14,400.00	**497.60**	N/A	−92.12	−6.71	504.47	**497.60**	N/A	−92.12	−6.71
Daskin95-88x8	Z	0.00	6201.10	0	0	14,400.00	487.81	N/A	−92.13	14,400.00	**479.99**	N/A	−92.26	−1.60	529.59	**479.99**	N/A	−92.26	−1.60
Ch69-100x10	W	913.19	1351.60	16	231	14,400.00	1079.41	18.20	−20.14	14,400.00	1018.10	11.49	−24.67	−5.68	423.01	**1011.53**	10.77	−25.16	−6.29
Ch69-100x10	Z	914.08	1640.83	29	207	14,400.00	1038.26	13.59	−36.72	14,400.00	**998.48**	9.23	−39.15	−3.83	529.59	**998.48**	9.23	−39.15	−3.83
Avg.		851.29	1411.12	24,166	156	8489.16	962.15	7.67	−10.25	5980.22	951.47	6.72	−10.87	−1.01	155.02	951.08	6.68	−10.89	−1.04

Notes: N/A: the lower bound of the problem cannot be found by CPLEX.
Italic numbers present the solution values which have been approved to be the optimum solution to the problem by CPLEX.
Bold numbers are the best-known solution values obtained by the corresponding approach.
[a]LB: the lower bound obtained from CPLEX under the LP relaxation.
[b]Int Sol: the best solution which CPLEX can solve as an integer programming problem.
[c]Gap_1: the percentage gap can be calculated as (the solution values – LB)/LB.
[d]Gap_2: the percentage gap can be calculated as (the solution values – Int Sol)/Int Sol.
[e]Gap_3: the percentage gap can be calculated as (SA – BC-SA)/BC-SA.
[f]SA*: best solution gained during all parameter analysis.

Table 5. Computational results for the KBSN data-set.

Instance	Type	CPLEX LB[a]	Int Sol[b]	Nodes	Cuts	Soltime	BC-SA BC-SA	Gap$_1$[c] (%)	Gap$_2$[d] (%)	Soltime	SA SA	Gap$_1$ (%)	Gap$_2$ (%)	Gap$_3$[e] (%)	Soltime	SA* SA*[f]	Gap$_1$ (%)	Gap$_2$ (%)	Gap$_3$ (%)
Srivastava86-8x2	X	***625.43***	***625.43***	0	37	0.53	**625.43**	0.00	0.00	0.27	**625.43**	0.00	0.00	0.00	0.30	**625.43**	0.00	0.00	0.00
Srivastava86-8x2	Y	***625.43***	***625.43***	0	26	0.67	**625.43**	0.00	0.00	0.40	**625.43**	0.00	0.00	0.00	0.28	**625.43**	0.00	0.00	0.00
Perl83-12x2	X	***242.41***	***242.41***	65	48	0.56	**242.41**	0.00	0.00	13.34	**242.41**	0.00	0.00	0.00	0.37	**242.41**	0.00	0.00	0.00
Perl83-12x2	Y	***242.41***	***242.41***	0	31	0.44	**242.41**	0.00	0.00	8.98	**242.41**	0.00	0.00	0.00	0.34	**242.41**	0.00	0.00	0.00
Gaskell67-21x5	X	***454.48***	***454.48***	5709	120	150.97	**454.48**	0.00	0.00	99.65	**454.48**	0.00	0.00	0.00	0.85	**454.48**	0.00	0.00	0.00
Gaskell67-21x5	Y	***454.48***	***454.48***	11,377	125	201.19	**454.48**	0.00	0.00	41.30	**454.48**	0.00	0.00	0.00	1.02	**454.48**	0.00	0.00	0.00
Gaskell67-22x5	X	***629.51***	***629.51***	52,608	164	593.67	**629.51**	0.00	0.00	91.87	**629.51**	0.00	0.00	0.00	1.50	**629.51**	0.00	0.00	0.00
Gaskell67-22x5	Y	***629.51***	***629.51***	54,620	142	715.34	**629.51**	0.00	0.00	130.73	**629.51**	0.00	0.00	0.00	1.50	**629.51**	0.00	0.00	0.00
Min92-27x5	X	***2998.80***	***2998.80***	383,431	403	12,246.03	**2998.80**	0.00	0.00	5.97	**2998.80**	0.00	0.00	0.00	1.13	**2998.80**	0.00	0.00	0.00
Min92-27x5	Y	***2998.80***	***2998.80***	479,065	400	12,739.70	**2998.80**	0.00	0.00	5.40	**2998.80**	0.00	0.00	0.00	1.26	**2998.80**	0.00	0.00	0.00
Gaskell67-29x5	X	***490.34***	***490.34***	16,075	162	394.08	**490.34**	0.00	0.00	18.77	**490.34**	0.00	0.00	0.00	2.54	**490.34**	0.00	0.00	0.00
Gaskell67-29x5	Y	***490.34***	***490.34***	6183	120	270.11	**490.34**	0.00	0.00	27.31	**490.34**	0.00	0.00	0.00	6.52	**490.34**	0.00	0.00	0.00
Gaskell67-32x5_1	X	***563.48***	***563.48***	40,198	176	2206.03	**563.47**	0.00	0.00	13.72	**563.48**	0.00	0.00	0.00	12.91	**563.48**	0.00	0.00	0.00
Gaskell67-32x5_1	Y	***563.48***	***563.48***	38,957	168	1982.05	**563.47**	0.00	0.00	7.15	**563.48**	0.00	0.00	0.00	10.94	**563.48**	0.00	0.00	0.00
Gaskell67-32x5_2	X	***507.03***	***507.03***	1315	108	158.36	**507.03**	0.00	0.00	18.36	**507.03**	0.00	0.00	0.00	14.43	**507.03**	0.00	0.00	0.00
Gaskell67-32x5_2	Y	***507.03***	***507.03***	3753	110	234.69	**507.03**	0.00	0.00	18.13	**507.03**	0.00	0.00	0.00	6.66	**507.03**	0.00	0.00	0.00
Gaskell67-36x5	X	***494.86***	***494.86***	3838	44	339.24	**494.86**	0.00	0.00	14,400.00	**494.86**	0.00	0.00	0.00	13.01	**494.86**	0.00	0.00	0.00
Gaskell67-36x5	Y	***494.86***	***494.86***	4667	89	533.38	**494.86**	0.00	0.00	14,400.00	**494.86**	0.00	0.00	0.00	13.34	**494.86**	0.00	0.00	0.00
Ch69-50x5	X	558.06	623.19	56,656	195	14,400.00	**576.69**	3.34	−7.46	14,400.00	578.97	3.61	−7.10	0.40	113.44	**576.69**	3.34	−7.46	0.00
Ch69-50x5	Y	558.24	580.05	48,776	176	14,400.00	578.97	3.71	−0.19	14,400.00	578.97	3.58	−0.19	0.00	166.79	**576.69**	3.30	−0.58	−0.39
Perl83-55x15	X	945.71	1073.09	856	168	14,400.00	985.56	4.21	−8.16	14,400.00	**982.55**	3.75	−8.44	−0.31	194.26	**982.55**	3.90	−8.44	−0.31
Perl83-55x15	Y	945.84	1066.82	873	98	14,400.00	988.06	4.46	−7.38	14,400.00	**982.55**	3.74	−7.90	−0.56	185.39	**982.55**	3.88	−7.90	−0.56
Ch69-75x10	X	782.58	1073.21	678	160	14,400.00	888.16	13.49	−17.24	14,400.00	**858.27**	8.82	−20.03	−3.37	216.51	**858.27**	9.67	−20.03	−3.37
Ch69-75x10	Y	782.29	1019.18	610	148	14,400.00	884.00	13.00	−13.26	14,400.00	861.08	9.15	−15.51	−2.59	271.89	**858.27**	9.71	−15.79	−2.91
Perl83-85x7	X	1295.38	1484.66	820	182	14,400.00	1381.57	6.65	−6.94	14,400.00	1346.42	3.79	−9.31	−2.54	420.22	**1346.30**	3.93	−9.32	−2.55
Perl83-85x7	Y	1299.50	1532.94	800	180	14,400.00	1368.64	5.32	−10.72	14,400.00	**1346.30**	3.48	−12.18	−1.63	337.39	**1346.30**	3.60	−12.18	−1.63
Daskin95-88x8	X	0.00	6470.99	0	0	14,400.00	399.17	N/A	−93.83	14,400.00	**375.69**	N/A	−94.19	−5.88	404.75	**375.69**	N/A	−94.19	−5.88
Daskin95-88x8	Y	339.03	6470.99	0	181	14,400.00	401.98	18.57	−93.79	14,400.00	**375.69**	9.76	−94.19	−6.54	430.26	**375.69**	10.81	−94.19	−6.54
Ch69-100x10	X	805.47	1091.12	154	190	14,400.00	880.18	9.28	−19.33	14,400.00	**854.41**	5.73	−21.69	−2.93	645.38	**854.41**	6.08	−21.69	−2.93
Ch69-100x10	Y	805.32	1149.91	122	165	14,400.00	880.89	9.38	−23.39	14,400.00	**854.41**	5.75	−25.70	−3.01	560.44	**854.41**	6.10	−25.70	−3.01
Avg.		771.00	1254.96	40,407	144	6852.23	807.55	3.15	−10.06	6736.71	800.27	2.11	−10.55	−0.97	134.52	800.02	2.22	−10.58	−1.00

Notes: N/A: the lower bound of the problem cannot be found by CPLEX.
Italic numbers present the solution values which have been approved to be the optimum solution to the problem by CPLEX.
Bold numbers are the best-known solution values obtained by the corresponding approach.
[a]LB: the lower bound obtained from CPLEX under the LP relaxation.
[b]Int Sol: the best solution which CPLEX can solve as an integer programming problem.
[c]Gap$_1$: the percentage gap can be calculated as (the solution values – LB)/LB.
[d]Gap$_2$: the percentage gap can be calculated as (the solution values – Int Sol)/Int Sol.
[e]Gap$_3$: the percentage gap can be calculated as (SA – BC-SA)/BC-SA.
[f]SA*: best solution gained during all parameter analysis.

Table 6. Computational results for the YLBAM data-set.

		CPLEX					SA				SA*		
Instance	Type	LB[a]	Int Sol[b]	Nodes	Cuts	Soltime	SA	Gap_1[c] (%)	Gap_2[d] (%)	Soltime	SA*[e]	Gap_1 (%)	Gap_2 (%)
Or76-117x14	W	10,822.84	119,515.29	0	0	14,400.00	12,498.00	15.48	−89.54	1076.56	**12,360.70**	12.44	−89.66
Or76-117x14	Z	10,798.72	119,084.92	0	0	14,400.00	12,563.60	16.34	−89.45	661.36	**12,438.30**	13.18	−89.56
Min92-134x8	W	5096.35	24,367.32	5	354	14,400.00	5973.36	17.21	−75.49	952.28	**5953.68**	14.40	−75.57
Min92-134x8	Z	5059.29	13,725.17	7	293	14,400.00	5886.07	16.34	−57.11	1224.05	**5762.55**	12.20	−58.01
Daskin95-150x10	W	37,419.17	419,188.28	0	0	14,400.00	46,088.40	23.17	−89.01	1471.59	**45,152.90**	17.13	−89.23
Daskin95-150x10	Z	0.00	404,854.70	0	0	14,400.00	45,473.70	N/A	N/A	1170.21	**44,418.40**	N/A	−89.03
Perl83-318x4	W	N/A	N/A	N/A	N/A	14,400.00	578,064.00	N/A	N/A	5098.35	**573,335.00**	N/A	N/A
Perl83-318x4	Z	N/A	N/A	N/A	N/A	14,400.00	569,787.00	N/A	N/A	4898.41	**566,157.00**	N/A	N/A
Perl83-318x4_2	W	N/A	N/A	N/A	N/A	14,400.00	726,192.00	N/A	N/A	16,858.20	**724,561.00**	N/A	N/A
Perl83-318x4_2	Z	N/A	N/A	N/A	N/A	14,400.00	693,592.00	N/A	N/A	11,732.70	**690,938.00**	N/A	N/A
Avg.		11,532.73	183,455.95	2	108	14,400.00	269,611.81	17.71	−80.12	4514.37	268,107.75	13.87	−81.84

Notes: N/A: the problem cannot be solved by CPLEX.
Bold numbers are the best known solution values obtained by the corresponding approach.
[a]LB: the lower bound obtained from CPLEX under the LP relaxation.
[b]Int Sol: the best solution which CPLEX can solve as an integer programming problem.
[c]Gap_1: the percentage gap can be calculated as (the solution values – LB)/LB.
[d]Gap_2: the percentage gap can be calculated as (the solution values – Int Sol)/Int Sol.
[e]SA*: best solution gained during all parameter analysis.

Table 7. Computational results for the YLBSN data-set.

		CPLEX					SA				SA*		
Instance	Type	LB[a]	Int Sol[b]	Nodes	Cuts	Soltime	SA	Gap_1[c] (%)	Gap_2[d] (%)	Soltime	SA*[e]	Gap_1 (%)	Gap_2 (%)
Or76-117x14	X	10,254.46	118,954.05	0	0	14,400.00	**11,716.20**	14.25	−90.15	447.97	**11,716.20**	12.48	−90.15
Or76-117x14	Y	10,254.46	116,127.12	0	0	14,400.00	11,935.40	16.39	−89.72	685.65	**11,825.60**	13.29	−89.82
Min92-134x8	X	4488.53	47,264.69	6	324	14,400.00	**5292.21**	17.91	−88.80	952.28	**5292.21**	15.19	−88.80
Min92-134x8	Y	4484.14	52,768.62	4	385	14,400.00	5303.18	18.27	−89.95	1009.97	**5292.21**	15.27	−89.97
Daskin95-150x10	X	31,398.65	406,491.46	0	0	14,400.00	37,848.50	20.54	−90.69	596.68	**37,418.40**	16.09	−90.79
Daskin95-150x10	Y	31,399.02	406,491.46	0	0	14,400.00	37,890.80	N/A	N/A	1170.21	**37,351.90**	N/A	−90.81
Perl83-318x4	X	N/A	N/A	N/A	N/A	14,400.00	562,748.00	N/A	N/A	4986.11	**560,146.00**	N/A	N/A
Perl83-318x4	Y	N/A	N/A	N/A	N/A	14,400.00	558,496.00	N/A	N/A	4219.51	**553,259.00**	N/A	N/A
Perl83-318x4_2	X	N/A	N/A	N/A	N/A	14,400.00	621,183.00	N/A	N/A	14,826.24	**602,211.00**	N/A	N/A
Perl83-318x4_2	Y	N/A	N/A	N/A	N/A	14,400.00	**599,555.00**	N/A	N/A	12,000.85	**599,555.00**	N/A	N/A
Avg.		15,379.88	191,349.57	2	118	14,400.00	245,196.83	17.47	−89.86	4089.55	242,406.75	14.46	−90.06

Notes: N/A: the problem cannot be solved by CPLEX.
Bold numbers are the best known solution values obtained by the corresponding approach.
[a]LB: the lower bound obtained from CPLEX under the LP relaxation.
[b]Int Sol: the best solution which CPLEX can solve as an integer programming problem.
[c]Gap_1: the percentage gap can be calculated as (the solution values – LB)/LB.
[d]Gap_2: the percentage gap can be calculated as (the solution values – Int Sol)/Int Sol.
[e]SA*: best solution gained during all parameter analysis.

Table 8. Computational results for the KPAM data-set.

		CPLEX					BC-SA				SA					SA*			
Instance	Type	LB[a]	Int Sol[b]	Nodes	Cuts	Soltime	BC-SA	Gap_1[c] (%)	Gap_2[d] (%)	Soltime	SA	Gap_1 (%)	Gap_2 (%)	Gap_3[e] (%)	Soltime	SA*[f]	Gap_1 (%)	Gap_2 (%)	Gap_3 (%)
coord20-5-1	W	***26,468.90***	***26,468.90***	142,268	169	3260.97	**26,468.90**	0.00	0.00	7337.09	**26,468.90**	0.00	0.00	0.00	1.99	**26,468.90**	0.00	0.00	0.00
coord20-5-1	Z	26,449.76	26,485.65	97,460	111	14,400.00	**26,461.30**	0.04	−0.09	75.24	**26,461.30**	0.04	−0.09	0.00	1.31	**26,461.30**	0.04	−0.09	0.00
coord20-5-1b	W	***18,718.80***	***18,718.80***	42,805	121	505.51	**18,718.80**	0.00	0.00	1.32	**18,718.80**	0.00	0.00	0.00	1.15	**18,718.80**	0.00	0.00	0.00
coord20-5-1b	Z	***18,703.00***	***18,703.00***	4080	121	48.47	**18,703.00**	0.00	0.00	1.36	**18,703.00**	0.00	0.00	0.00	0.67	**18,703.00**	0.00	0.00	0.00
coord20-5-2	W	***27,988.70***	***27,988.70***	7245	169	227.44	**27,988.70**	0.00	0.00	37.79	**27,988.70**	0.00	0.00	0.00	1.43	**27,988.70**	0.00	0.00	0.00
coord20-5-2	Z	***27,980.60***	***27,980.60***	20,785	121	403.17	**27,980.60**	0.00	0.00	73.81	**27,980.60**	0.00	0.00	0.00	1.17	**27,980.60**	0.00	0.00	0.00
coord20-5-2b	W	***17,125.50***	***17,125.50***	18,947	71	209.08	**17,125.50**	0.00	0.00	7.09	**17,125.50**	0.00	0.00	0.00	1.44	**17,125.50**	0.00	0.00	0.00
coord20-5-2b	Z	***17,120.50***	***17,120.50***	10,669	44	85.75	**17,120.50**	0.00	0.00	4.40	**17,120.50**	0.00	0.00	0.00	0.88	**17,120.50**	0.00	0.00	0.00
coord50-5-1	W	32,711.88	34,028.50	18,049	227	14,400.00	33,786.20	3.28	−0.71	14,400.00	32,855.30	0.44	−3.45	−2.76	155.66	**32,822.30**	0.34	−3.54	−2.85
coord50-5-1	Z	32,685.69	35,996.86	7572	225	14,400.00	32,804.50	0.36	−8.87	14,400.00	**32,768.90**	0.25	−8.97	−0.11	331.37	**32,768.40**	0.25	−8.97	−0.11
coord50-5-1b	W	26,478.17	26,550.90	61,522	213	14,400.00	26,541.40	0.24	−0.04	14,400.00	**26,528.90**	0.19	−0.08	−0.05	272.28	**26,528.90**	0.19	−0.08	−0.05
coord50-5-1b	Z	26,477.99	26,566.01	13,875	180	14,400.00	26,530.80	0.20	−0.13	14,400.00	26,521.80	0.17	−0.17	−0.03	279.46	**26,520.10**	0.16	−0.17	−0.04
coord50–5-2	W	41,770.08	44,074.11	8265	211	14,400.00	42,860.40	2.61	−2.75	14,400.00	**42,839.10**	2.56	−2.80	−0.05	186.20	**42,839.10**	2.56	−2.80	−0.05
coord50-5-2	Z	41,753.44	41,908.26	40,499	330	14,400.00	41,825.80	0.17	−0.20	14,400.00	**41,807.60**	0.13	−0.24	−0.04	212.79	**41,807.60**	0.13	−0.24	−0.04
coord50-5-2b	W	35,608.37	35,662.71	37,439	281	14,400.00	35,661.90	0.15	0.00	14,400.00	**35,657.00**	0.14	−0.02	−0.01	142.54	**35,657.00**	0.14	−0.02	−0.01
coord50-5-2b	Z	35,601.08	35,657.16	46,323	344	14,400.00	**35,648.40**	0.13	−0.02	14,400.00	**35,648.40**	0.13	−0.02	0.00	133.07	**35,648.40**	0.13	−0.02	0.00
coord50-5-3	W	22,429.06	25,657.09	7951	263	14,400.00	23,533.10	4.92	−8.28	14,400.00	**23,501.10**	4.78	−8.40	−0.14	236.32	**23,501.10**	4.78	−8.40	−0.14
coord50-5-3	Z	23,432.69	24,536.37	10,677	271	14,400.00	23,501.50	0.29	−4.22	14,400.00	**23,485.30**	0.22	−4.28	−0.07	212.68	**23,485.30**	0.22	−4.28	−0.07
coord50-5-3b	W	17,136.31	17,197.59	12,167	216	14,400.00	17,182.00	0.27	−0.09	14,400.00	17,177.00	0.24	−0.12	−0.03	156.57	**17,176.80**	0.24	−0.12	−0.03
coord50-5-3b	Z	17,137.13	17,193.04	9284	270	14,400.00	17,174.10	0.22	−0.11	14,400.00	**17,173.30**	0.21	−0.11	0.00	150.85	**17,173.30**	0.21	−0.11	0.00
coord100-5-1	W	158,094.33	164,382.57	1034	469	14,400.00	160,277.00	1.38	−2.50	14,400.00	159,229.00	0.72	−3.14	−0.65	969.61	**159,219.00**	0.71	−3.14	−0.66
coord100-5-1	Z	158,032.23	161,433.17	2097	403	14,400.00	159,200.00	0.74	−1.38	14,400.00	158,150.00	0.07	−2.03	−0.66	847.04	**158,147.00**	0.07	−2.04	−0.66
coord100-5-1b	W	144,554.86	147,801.60	822	489	14,400.00	145,697.00	0.79	−1.42	14,400.00	**145,633.00**	0.75	−1.47	−0.04	602.87	**145,633.00**	0.75	−1.47	−0.04
coord100-5-1b	Z	144,530.42	147,821.61	1938	374	14,400.00	145,650.00	0.77	−1.47	14,400.00	145,612.00	0.75	−1.49	−0.03	688.59	**144,646.00**	0.08	−2.15	−0.69
coord100-5-2	W	120,875.53	125,253.97	3598	415	14,400.00	168,371.00	39.29	34.42	14,400.00	123,016.00	1.77	−1.79	−26.94	990.69	**123,009.00**	1.77	−1.79	−26.94
coord100-5-2	Z	120,848.84	124,011.93	4680	422	14,400.00	172,418.00	42.67	39.03	14,400.00	122,957.00	1.74	−0.85	−28.69	891.51	**122,956.00**	1.74	−0.85	−28.69
coord100-5-2b	W	108,333.13	110,742.71	6202	569	14,400.00	153,827.00	41.99	38.90	14,400.00	109,434.00	1.02	−1.18	−28.86	609.97	**109,432.00**	1.01	−1.18	−28.86
coord100-5-2b	Z	108,324.37	274,423.61	3809	391	14,400.00	153,776.00	41.96	−43.96	14,400.00	109,400.00	0.99	−60.13	−28.86	882.30	**109,396.00**	0.99	−60.14	−28.86
coord100-5-3	W	110,328.47	117,664.19	2778	381	14,400.00	114,517.00	3.80	−2.67	14,400.00	112,453.00	1.93	−4.43	−1.80	877.24	**112,451.00**	1.92	−4.43	−1.80
coord100-5-3	Z	110,322.00	123,841.16	1917	351	14,400.00	112,467.00	1.94	−9.18	14,400.00	112,420.00	1.90	−9.22	−0.04	954.78	**112,411.00**	1.89	−9.23	−0.05
coord100-5-3b	W	97,867.20	102,205.05	1921	357	14,400.00	100,006.00	2.19	−2.15	14,400.00	98,947.80	1.10	−3.19	−1.06	546.41	**98,945.00**	1.10	−3.19	−1.06
coord100-5-3b	Z	97,868.70	102,104.54	1725	399	14,400.00	98,972.50	1.13	−3.07	14,400.00	**98,937.20**	1.09	−3.10	−0.04	567.22	**98,937.20**	1.09	−3.10	−0.04
coord100-10-1	W	221,249.53	232,850.39	598	195	14,400.00	224,445.00	1.44	−3.61	14,400.00	**223,398.00**	0.97	−4.06	−0.47	965.98	**223,398.00**	0.97	−4.06	−0.47
coord100-10-1	Z	221,220.35	227,448.79	307	211	14,400.00	222,434.00	0.55	−2.20	14,400.00	222,337.00	0.50	−2.25	−0.04	1259.03	**222,336.00**	0.50	−2.25	−0.04
coord100-10-1b	W	208,868.35	218,179.21	118	235	14,400.00	209,084.00	0.10	−4.17	14,400.00	209,004.00	0.06	−4.21	−0.04	958.82	**208,976.00**	0.05	−4.22	−0.05
coord100-10-1b	Z	208,861.58	217,155.91	125	192	14,400.00	208,958.00	0.05	−3.78	14,400.00	208,942.00	0.04	−3.78	−0.01	834.17	**208,935.00**	0.04	−3.79	−0.01
coord100-10-2	W	169,666.72	184,044.64	0	187	14,400.00	218,395.00	28.72	18.66	14,400.00	171,807.00	1.26	−6.65	−21.33	943.72	**171,791.00**	1.25	−6.66	−21.34
coord100-10-2	Z	169,646.82	238,563.89	219	214	14,400.00	218,243.00	28.65	−8.52	14,400.00	171,760.00	1.25	−28.00	−21.30	1120.02	**171,740.00**	1.23	−28.01	−21.31
coord100-10-2b	W	157,396.54	320,794.44	224	234	14,400.00	159,507.00	1.34	−50.28	14,400.00	158,480.00	0.69	−50.60	−0.64	826.42	**158,474.00**	0.68	−50.60	−0.65
coord100-10-2b	Z	157,391.02	210,328.57	49	219	14,400.00	202,371.85	28.58	−3.78	14,400.00	158,452.00	0.67	−24.66	−21.70	908.20	**158,446.00**	0.67	−24.67	−21.71
coord100-10-3	W	167,428.95	220,766.50	24	301	14,400.00	210,355.00	25.64	−4.72	14,400.00	205,159.00	22.53	−7.07	−2.47	1019.56	**179,825.00**	7.40	−18.55	−14.51

(*Continued*)

Table 8. (*Continued*).

		CPLEX					BC-SA				SA					SA*			
Instance	Type	LB[a]	Int Sol[b]	Nodes	Cuts	Soltime	BC-SA	Gap_1[c] (%)	Gap_2[d] (%)	Soltime	SA	Gap_1 (%)	Gap_2 (%)	Gap_3[e] (%)	Soltime	SA*[f]	Gap_1 (%)	Gap_2 (%)	Gap_3 (%)
coord100-10-3	Z	162,496.16	216,545.24	82	236	14,400.00	166,404.22	2.41	−23.15	14,400.00	**165,366.00**	1.77	−23.63	−0.62	1054.37	**165,366.00**	1.77	−23.63	−0.62
coord100-10-3b	W	154,085.58	318,016.56	39	180	14,400.00	196,017.00	27.21	−38.36	14,400.00	191,762.00	24.45	−39.70	−2.17	660.60	**165,436.00**	7.37	−47.98	−15.60
coord100-10-3b	Z	150,174.76	206,294.09	0	191	14,400.00	152,088.46	1.27	−26.28	14,400.00	**151,967.00**	1.19	−26.33	−0.08	897.20	**151,967.00**	1.19	−26.33	−0.08
Avg.		94,596.46	114,961.24	14,822	263	12,216.83	104,797.69	7.67	−2.98	11,953.14	97,071.68	1.79	−7.77	−4.36	530.82	95,872.00	1.04	−8.23	−4.96

Notes: Italic numbers presents the solution values which have been approved to be the optimum solution to the problem by CPLEX.
Bold numbers are the best known solution values obtained by the corresponding approach.
[a]LB: the lower bound obtained from CPLEX under the LP relaxation.
[b]Int Sol: the best solution which CPLEX can solve as an integer programming problem.
[c]Gap_1: the percentage gap can be calculated as (the solution values – LB)/LB.
[d]Gap_2: the percentage gap can be calculated as (the solution values – Int Sol)/Int Sol.
[e]Gap_3: the percentage gap can be calculated as (SA – BC-SA)/BC-SA.
[f]SA*: best solution gained during all parameter analysis.

Table 9. Computational results for the KPSN data-set.

		CPLEX					BC-SA				SA					SA*			
Instance	Type	LB[a]	Int Sol[b]	Nodes	Cuts	Soltime	BC-SA	Gap_1[c] (%)	Gap_2[d] (%)	Soltime	SA	Gap_1 (%)	Gap_2 (%)	Gap_3[e] (%)	Soltime	SA*[f]	Gap_1 (%)	Gap_2 (%)	Gap_3 (%)
coord20-5-1	X	***16,816.50***	***16,816.50***	31,970	96	425.89	**16,816.50**	0.00	0.00	84.51	**16,816.50**	0.00	0.00	0.00	0.82	**16,816.50**	0.00	0.00	0.00
coord20-5-1	Y	***16,816.50***	***16,816.50***	38,173	131	522.88	**16,816.50**	0.00	0.00	1900.62	**16,816.50**	0.00	0.00	0.00	0.63	**16,816.50**	0.00	0.00	0.00
coord20-5-1b	X	***9167.15***	***9167.15***	0	9	0.56	9167.14	0.00	0.00	0.44	**9167.15**	0.00	0.00	0.00	0.66	**9167.15**	0.00	0.00	0.00
coord20-5-1b	Y	***9167.15***	***9167.15***	0	11	0.50	9167.14	0.00	0.00	0.27	**9167.15**	0.00	0.00	0.00	0.63	**9167.15**	0.00	0.00	0.00
coord20-5-2	X	***17,814.70***	***17,814.70***	217,138	172	1879.69	**17,814.70**	0.00	0.00	27.93	**17,814.70**	0.00	0.00	0.00	1.43	**17,814.70**	0.00	0.00	0.00
coord20-5-2	Y	***17,814.70***	***17,814.70***	117,984	113	1338.80	**17,814.70**	0.00	0.00	16.02	**17,814.70**	0.00	0.00	0.00	0.97	**17,814.70**	0.00	0.00	0.00
coord20-5-2b	X	***10,257.30***	***10,257.30***	2779	114	5.95	**10,257.30**	0.00	0.00	1.61	**10,257.30**	0.00	0.00	0.00	1.11	**10,257.30**	0.00	0.00	0.00
coord20-5-2b	Y	***10,257.30***	***10,257.30***	1998	70	9.78	**10,257.30**	0.00	0.00	1.38	**10,257.30**	0.00	0.00	0.00	1.32	**10,257.30**	0.00	0.00	0.00
coord50-5-1	X	16,298.43	17,431.79	8890	199	14,400.00	16,350.00	0.32	−6.21	14,400.00	**16,346.70**	0.30	−6.22	−0.02	120.66	**16,346.70**	0.30	−6.22	−0.02
coord50-5-1	Y	16,300.94	16,489.50	61,184	215	14,400.00	16,355.20	0.33	−0.81	14,400.00	**16,346.70**	0.28	−0.87	−0.05	148.23	**16,346.70**	0.28	−0.87	−0.05
coord50-5-1b	X	13,110.95	13,283.05	6946	145	14,400.00	**13,132.90**	0.17	−1.13	14,400.00	**13,132.90**	0.17	−1.13	0.00	96.67	**13,132.90**	0.17	−1.13	0.00
coord50-5-1b	Y	13,116.72	13,132.93	226,509	210	14,400.00	**13,132.90**	0.12	0.00	14,400.00	**13,132.90**	0.12	0.00	0.00	117.08	**13,132.90**	0.12	0.00	0.00
coord50-5-2	X	26,348.41	26,407.73	113,959	254	14,400.00	26,395.60	0.18	−0.05	14,400.00	**26,390.90**	0.16	−0.06	−0.02	152.08	**26,390.90**	0.16	−0.06	−0.02
coord50-5-2	Y	26,348.06	26,397.91	35,306	244	14,400.00	26,392.70	0.17	−0.02	14,400.00	**26,390.90**	0.16	−0.03	−0.01	144.66	**26,390.90**	0.16	−0.03	−0.01
coord50-5-2b	X	22,264.71	22,268.53	425,028	402	14,400.00	**22,268.50**	0.02	0.00	213.75	**22,268.50**	0.02	0.00	0.00	106.55	**22,268.50**	0.02	0.00	0.00
coord50-5-2b	Y	22,261.06	22,269.36	42,996	239	14,400.00	**22,268.50**	0.03	0.00	1147.76	**22,268.50**	0.03	0.00	0.00	108.95	**22,268.50**	0.03	0.00	0.00
coord50-5-3	X	11,580.67	12,641.04	11,272	205	14,400.00	11,624.20	0.38	−8.04	14,400.00	**11,621.20**	0.35	−8.07	−0.03	117.18	**11,621.20**	0.35	−8.07	−0.03
coord50-5-3	Y	11,581.18	12,676.57	11,014	215	14,400.00	11,626.60	0.39	−8.28	14,400.00	**11,621.20**	0.35	−8.33	−0.05	115.31	**11,621.20**	0.35	−8.33	−0.05
coord50-5-3b	X	8461.42	8472.55	431,542	297	14,400.00	8472.39	0.13	0.00	14,400.00	**8469.87**	0.10	−0.03	−0.03	157.60	**8469.87**	0.10	−0.03	−0.03
coord50-5-3b	Y	8460.77	8472.80	292,903	249	14,400.00	**8469.87**	0.11	−0.03	14,400.00	**8469.87**	0.11	−0.03	0.00	120.08	**8469.87**	0.11	−0.03	0.00
coord100-5-1	X	102,182.44	104,919.19	2187	292	14,400.00	102,388.00	0.20	−2.41	14,400.00	102,310.00	0.12	−2.49	−0.08	771.44	**102,304.00**	0.12	−2.49	−0.08
coord100-5-1	Y	102,180.50	103,716.66	2962	253	14,400.00	102,381.00	0.20	−1.29	14,400.00	102,308.00	0.12	−1.36	−0.07	747.70	**102,305.00**	0.12	−1.36	−0.07
coord100-5-1b	X	94,746.03	97,111.74	1859	367	14,400.00	94,884.00	0.15	−2.29	14,400.00	**94,842.30**	0.10	−2.34	−0.04	634.16	**94,842.30**	0.10	−2.34	−0.04
coord100-5-1b	Y	94,748.91	95,090.23	5976	432	14,400.00	94,878.80	0.14	−0.22	14,400.00	**94,842.60**	0.10	−0.26	−0.04	585.05	**94,842.60**	0.10	−0.26	−0.04
coord100-5-2	X	105,541.73	107,314.40	6096	334	14,400.00	105,655.00	0.11	−1.55	14,400.00	105,620.00	0.07	−1.58	−0.03	720.51	**105,616.00**	0.07	−1.58	−0.04
coord100-5-2	Y	105,542.24	106,991.04	7584	335	14,400.00	105,655.00	0.11	−1.25	14,400.00	**105,616.00**	0.07	−1.29	−0.04	631.69	**105,616.00**	0.07	−1.29	−0.04
coord100-5-2b	X	97,131.94	97,350.95	1806	245	14,400.00	97,213.80	0.08	−0.14	14,400.00	97,188.90	0.06	−0.17	−0.03	684.05	**97,187.70**	0.06	−0.17	−0.03
coord100-5-2b	Y	97,132.40	97,257.12	6234	297	14,400.00	97,206.00	0.08	−0.05	14,400.00	97,188.50	0.06	−0.07	−0.02	692.02	**97,187.70**	0.06	−0.07	−0.02
coord100-5-3	X	56,425.57	57,737.83	8877	310	14,400.00	56,552.10	0.22	−2.05	14,400.00	56,513.80	0.16	−2.12	−0.07	583.00	**56,513.50**	0.16	−2.12	−0.07
coord100-5-3	Y	56,423.90	60,751.32	2585	259	14,400.00	56,581.90	0.28	−6.86	14,400.00	**56,513.50**	0.16	−6.98	−0.12	753.22	**56,513.50**	0.16	−6.98	−0.12
coord100-5-3b	X	50,141.94	51,540.30	1885	220	14,400.00	50,224.60	0.16	−2.55	14,400.00	50,201.60	0.12	−2.60	−0.05	680.30	**50,200.70**	0.12	−2.60	−0.05
coord100-5-3b	Y	50,141.30	51,534.84	1788	203	14,400.00	50,220.80	0.16	−2.55	14,400.00	50,201.60	0.12	−2.59	−0.04	532.92	**50,201.50**	0.12	−2.59	−0.04
coord100-10-1	X	108,585.58	116,992.57	793	490	14,400.00	109,785.00	1.10	−6.16	14,400.00	109,710.00	1.04	−6.22	−0.07	519.61	**108,750.00**	0.15	−7.05	−0.94
coord100-10-1	Y	108,590.07	118,061.36	931	438	14,400.00	109,787.00	1.10	−7.01	14,400.00	**109,704.30**	1.03	−7.08	−0.08	740.72	**109,704.30**	1.03	−7.08	−0.08
coord100-10-1b	X	102,337.82	104,546.73	6266	389	14,400.00	102,430.00	0.09	−2.02	14,400.00	102,407.00	0.07	−2.05	−0.02	541.27	**102,404.00**	0.06	−2.05	−0.03
coord100-10-1b	Y	102,337.29	104,555.47	2810	355	14,400.00	102,426.00	0.09	−2.04	14,400.00	**102,404.00**	0.07	−2.06	−0.02	462.81	**102,404.00**	0.07	−2.06	−0.02
coord100-10-2	X	105,384.93	109,844.35	2855	363	14,400.00	155,190.00	47.26	41.28	14,400.00	105,486.00	0.10	−3.97	−32.03	969.21	**105,484.00**	0.09	−3.97	−32.03
coord100-10-2	Y	105,385.02	109,692.42	4426	422	14,400.00	107,521.00	2.03	−1.98	14,400.00	105,491.00	0.10	−3.83	−1.89	743.11	**105,479.00**	0.09	−3.84	−1.90
coord100-10-2b	X	98,059.11	99,263.63	3838	329	14,400.00	99,140.10	1.10	−0.12	14,400.00	99,134.60	1.10	−0.13	−0.01	735.14	**99,129.30**	1.09	−0.14	−0.01
coord100-10-2b	Y	98,059.49	99,187.81	10,680	375	14,400.00	99,138.40	1.10	−0.05	14,400.00	**99,129.30**	1.09	−0.06	−0.01	520.49	**99,129.30**	1.09	−0.06	−0.01

(*Continued*)

Table 9. (*Continued*).

		CPLEX					BC-SA				SA					SA*			
Instance	Type	LB[a]	Int Sol[b]	Nodes	Cuts	Soltime	BC-SA	Gap_1[c] (%)	Gap_2[d] (%)	Soltime	SA	Gap_1 (%)	Gap_2 (%)	Gap_3[e] (%)	Soltime	SA*[f]	Gap_1 (%)	Gap_2 (%)	Gap_3 (%)
coord100-10-3	X	99,566.66	105,186.37	859	384	14,400.00	100,702.00	1.14	−4.26	14,400.00	99,694.20	0.13	−5.22	−1.00	747.13	**99,688.40**	0.12	−5.23	−1.01
coord100-10-3	Y	99,566.43	104,807.64	1937	435	14,400.00	99,913.60	0.35	−4.67	14,400.00	99,697.30	0.13	−4.88	−0.22	947.14	**99,691.60**	0.13	−4.88	−0.22
coord100-10-3b	X	93,372.01	94,874.21	6973	392	14,400.00	93,450.50	0.08	−1.50	14,400.00	93,422.70	0.05	−1.53	−0.03	631.74	**93,420.60**	0.05	−1.53	−0.03
coord100-10-3b	Y	93,371.03	93,654.99	5122	332	14,400.00	93,475.00	0.11	−0.19	14,400.00	93,422.20	0.05	−0.25	−0.06	573.07	**93,420.60**	0.05	−0.25	−0.06
Avg.		57,527.25	59,091.78	49,430	269	11,876.91	58,895.46	1.36	−0.83	11,204.42	57,673.20	0.19	−1.95	−0.82	401.37	57,650.16	0.17	−1.97	−0.84

Notes: Italic numbers presents the solution values which have been approved to be the optimum solution to the problem by CPLEX.
Bold numbers are the best known solution values obtained by the corresponding approach.
[a]LB: the lower bound obtained from CPLEX under the LP relaxation.
[b]Int Sol: the best solution which CPLEX can solve as an integer programming problem.
[c]Gap_1: the percentage gap can be calculated as (the solution values – LB)/LB.
[d]Gap_2: the percentage gap can be calculated as (the solution values – Int Sol)/Int Sol.
[e]Gap_3: the percentage gap can be calculated as (SA – BC-SA)/BC-SA.
[f]SA*: best solution gained during all parameter analysis.

Table 10. Computational results for the YLPAM data-set.

		CPLEX					SA				SA*		
Instance	Type	LB[a]	Int Sol[b]	Nodes	Cuts	Soltime	SA	Gap_1[c] (%)	Gap_2[d] (%)	Soltime	SA*[e]	Gap_1 (%)	Gap_2 (%)
coord50-5-2BIS	W	28,129.55	31,257.03	6975	254	14,400.00	**29,188.10**	3.76	−6.62	260.85	**29,188.10**	3.76	−6.62
coord50-5-2BIS	Z	28,121.04	30,405.52	8384	218	14,400.00	**29,172.80**	3.74	−4.05	242.08	**29,172.80**	3.74	−10.00
coord50-5-2bBIS	W	23,932.26	24,020.51	7369	170	14,400.00	**23,984.20**	0.22	−0.15	191.67	**23,984.20**	0.22	−2.00
coord50-5-2bBIS	Z	23,928.66	25,081.64	5992	231	14,400.00	23,982.00	0.22	−4.38	184.66	**23,981.90**	0.22	−2.00
Avg.		26,027.88	27,691.17	7180	218	14,400.00	26,581.78	1.99	−3.80	219.82	26,581.75	1.99	−5.15
coord200-10-1	W	N/A	N/A	N/A	N/A	14,400.00	287,017.00	N/A	N/A	3056.46	**286,920.00**	N/A	N/A
coord200-10-1	Z	N/A	N/A	N/A	N/A	14,400.00	286,726.40	N/A	N/A	3174.47	**286,724.00**	N/A	N/A
coord200-10-1b	W	N/A	N/A	N/A	N/A	14,400.00	**259,776.00**	N/A	N/A	2665.28	**259,776.00**	N/A	N/A
coord200-10-1b	Z	N/A	N/A	N/A	N/A	14,400.00	259,595.00	N/A	N/A	2669.62	**259,587.00**	N/A	N/A
coord200-10-2	W	N/A	N/A	N/A	N/A	14,400.00	326,217.00	N/A	N/A	3157.42	**326,208.00**	N/A	N/A
coord200-10-2	Z	N/A	N/A	N/A	N/A	14,400.00	326,097.00	N/A	N/A	3050.06	**325,101.00**	N/A	N/A
coord200-10-2b	W	N/A	N/A	N/A	N/A	14,400.00	300,440.00	N/A	N/A	1926.47	**299,463.00**	N/A	N/A
coord200-10-2b	Z	N/A	N/A	N/A	N/A	14,400.00	299,399.00	N/A	N/A	2399.09	**299,395.00**	N/A	N/A
coord200-10-3	W	N/A	N/A	N/A	N/A	14,400.00	284,786.00	N/A	N/A	3179.75	**284,766.00**	N/A	N/A
coord200-10-3	Z	N/A	N/A	N/A	N/A	14,400.00	283,697.00	N/A	N/A	3081.28	**283,688.00**	N/A	N/A
coord200-10-3b	W	N/A	N/A	N/A	N/A	14,400.00	**257,826.00**	N/A	N/A	2547.47	**257,826.00**	N/A	N/A
coord200-10-3b	Z	N/A	N/A	N/A	N/A	14,400.00	257,751.00	N/A	N/A	2716.25	**257,744.00**	N/A	N/A
Avg.		N/A	N/A	N/A	N/A	14,400.00	285,777.28	N/A	N/A	2801.97	285,599.83	N/A	N/A

Notes: N/A: the problem cannot be solved by CPLEX.
Bold numbers are the best known solution values obtained by the corresponding approach.
[a]LB: the lower bound obtained from CPLEX under the LP relaxation.
[b]Int Sol: the best solution which CPLEX can solve as an integer programming problem.
[c]Gap_1: the percentage gap can be calculated as (the solution values – LB)/LB.
[d]Gap_2: the percentage gap can be calculated as (the solution values – Int Sol)/Int Sol.
[e]SA*: best solution gained during all parameter analysis.

Table 11. Computational results for the YLPSN data-set.

		CPLEX					SA				SA*		
Instance	Type	LB[a]	Int Sol[b]	Nodes	Cuts	Soltime	SA	Gap_1[c] (%)	Gap_2[d] (%)	Soltime	SA*[e]	Gap_1 (%)	Gap_2 (%)
coord50-5-2BIS	X	20,339.97	21,432.15	8952	241	14400.00	**20,414.70**	0.37	−4.75	214.77	**20,414.70**	0.37	−4.75
coord50-5-2BIS	Y	20,343.56	20,414.74	104,223	318	14400.00	**20,414.70**	0.35	0.00	213.58	**20,414.70**	0.35	0.00
coord50-5-2bBIS	X	22,901.03	22,910.30	102,956	405	14400.00	**22,908.70**	0.03	−0.01	200.63	**22,908.70**	0.03	−0.01
coord50-5-2bBIS	Y	22,882.00	22,916.93	10,331	210	14400.00	**22,908.70**	0.12	−0.04	183.67	**22,908.70**	0.12	−0.04
Avg.		21,616.64	21,918.53	56,616	294	14400.00	21,661.70	0.22	−1.20	203.16	21,661.70	0.22	−1.20
coord200-10-1	X	N/A	N/A	N/A	N/A	14400.00	174,345.00	N/A	N/A	2323.82	**173,420.00**	N/A	N/A
coord200-10-1	Y	N/A	N/A	N/A	N/A	14400.00	174,367.00	N/A	N/A	2384.69	**173,416.00**	N/A	N/A
coord200-10-1b	X	N/A	N/A	N/A	N/A	14400.00	159,680.00	N/A	N/A	1867.89	**159,669.00**	N/A	N/A
coord200-10-1b	Y	N/A	N/A	N/A	N/A	14400.00	159,675.00	N/A	N/A	2024.64	**159,664.00**	N/A	N/A
coord200-10-2	X	N/A	N/A	N/A	N/A	14400.00	200,906.00	N/A	N/A	2734.11	**200,898.00**	N/A	N/A
coord200-10-2	Y	N/A	N/A	N/A	N/A	14400.00	200,902.00	N/A	N/A	2590.55	**200,896.00**	N/A	N/A
coord200-10-2b	X	N/A	N/A	N/A	N/A	14400.00	183,419.00	N/A	N/A	2357.53	**183,416.00**	N/A	N/A
coord200-10-2b	Y	N/A	N/A	N/A	N/A	14400.00	**183,416.00**	N/A	N/A	2128.67	**183,416.00**	N/A	N/A
coord200-10-3	X	N/A	N/A	N/A	N/A	14400.00	178,613.00	N/A	N/A	2510.21	**178,610.00**	N/A	N/A
coord200-10-3	Y	N/A	N/A	N/A	N/A	14400.00	**178,604.00**	N/A	N/A	2656.55	**178,604.00**	N/A	N/A
coord200-10-3b	X	N/A	N/A	N/A	N/A	14400.00	**164,964.00**	N/A	N/A	2287.55	**164,964.00**	N/A	N/A
coord200-10-3b	Y	N/A	N/A	N/A	N/A	14400.00	164,974.00	N/A	N/A	2215.80	**164,962.00**	N/A	N/A
Avg.		N/A	N/A	N/A	N/A	14400.00	176,988.75	N/A	N/A	2340.17	176,827.92	N/A	N/A

Notes: N/A: the problem cannot be solved by CPLEX.
Bold numbers are the best known solution values obtained by the corresponding approach.
[a]LB: the lower bound obtained from CPLEX under the LP relaxation.
[b]Int Sol: the best solution which CPLEX can solve as an integer programming problem.
[c]Gap_1: the percentage gap can be calculated as (the solution values – LB)/LB.
[d]Gap_2: the percentage gap can be calculated as (the solution values – Int Sol)/Int Sol.
[e]SA*: best solution gained during all parameter analysis.

5. Conclusion and further research

LRPSPD is widely used in the beverage industry and after-sales services, and thus attracts researchers' and practitioners' interest in the problem. This study proposed an SA heuristic with a special solution encoding scheme for solving LRPSPD. The solution encoding scheme can enlarge the search space and facilitate the search for better solutions.

We compared the results of the BC-SA to those of our SA heuristic on four sets of existing LRPSPD instances – KBAM, KBSN, KPAM and KPSN. The computational results attest that the proposed SA outperforms the BC-SA in terms of solution quality. We also generated four sets of large-scale LRPSPD instances – YLBAM, YLBSN, YLPAM and YLPSN, and applied our SA algorithm to these data-sets. The solution quality and computational time are satisfactory.

Further research may focus on using different heuristics for solving the LRPSPD and comparing their performance on the data-sets generated by Karaoglan et al. (2011) or this study, especially on large-scale problems. Future studies may also consider the LRPSPD with more practical constraints, such as time windows, inter-depot routes, etc., to bring the problem closer to the reality.

Acknowledgement

This work was partially supported by the National Science Council of Taiwan [grant number NSC 100-2410-H-011-005-MY2]. This support is gratefully acknowledged.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Vincent F. Yu http://orcid.org/0000-0001-8975-0606

References

Aksen, D., and K. Altinkemer. 2008. "A Location-routing Problem for the Conversion to the "Click-and-mortar" Retailing: The Static Case." *European Journal of Operational Research* 186 (2): 554–575.

Andel, T. 1997. "Reverse Logistics: A Second Chance to Profit: Whether through Refurbishment or Recycling, Companies Are Finding Profit in Returned Products." *Transportation & Distribution* 38 (7): 61–64.

Angelelli, E., and R. Mansini. 2002. "The Vehicle Routing Problem with Time Windows and Simultaneous Pick-up and Delivery." In *Quantitative Approaches to Distribution Logistics and Supply Chain Management*, edited by A. Klose, M. G. Speranza, and L. N. VanWassenhove, 249–267. Berlin: Springer.

Barreto, S. S. 2004. "Análise e Modelização de Problemas de localização-distribuição [Analysis and Modelling of Location-routing Problems]." Unpublished Ph.D. Thesis, University of Aveiro.

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.

Christofides, N., A. Mingozzi, and P. Toth 1979. "The Vehicle Routing Problem." In *Combinatorial Optimization*, edited by N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, 315–338. Chichester: Wiley.

Clarke, G., and J. W. Wright. 1964. "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points." *Operations Research* 12 (4): 568–581.

Derbel, H., B. Jarboui, S. Hanafi, and H. Chabchoub. 2012. "Genetic Algorithm with Iterated Local Search for Solving a Location-routing Problem." *Expert Systems with Applications* 39 (3): 2865–2871.

Dethloff, J. 2001. "Vehicle Routing and Reverse Logistics: The Vehicle Routing Problem with Simultaneous Delivery and Pick-up." *OR Spektrum* 23 (1): 79–96.

Duhamel, C., P. Lacomme, C. Prins, and C. Prodhon. 2010. "A GRASP×ELS Approach for the Capacitated Location-routing Problem." *Computers & Operations Research* 37 (11): 1912–1923.

Gajpal, Y., and P. Abad. 2009. "An Ant Colony System (ACS) for Vehicle Routing Problem with Simultaneous Delivery and Pickup." *Computers & Operations Research* 36 (12): 3215–3223.

Goksal, F. P., I. Karaoglan, and F. Altiparmak. 2013. "A Hybrid Discrete Particle Swarm Optimization for Vehicle Routing Problem with Simultaneous Pickup and Delivery." *Computers & Industrial Engineering* 65 (1): 39–53.

Jacobsen, S. K., and O. B. G. Madsen. 1980. "A Comparative Study of Heuristics for a Two-level Routing-location Problem." *European Journal of Operational Research* 5 (6): 378–387.

Jeong, C.-S., and M.-H. Kim. 1991. "Fast Parallel Simulated Annealing for Traveling Salesman Problem on SIMD Machines with Linear Interconnections." *Parallel Computing* 17 (2–3): 221–228.

Karaoglan, I., F. Altiparmak, I. Kara, and B. Dengiz. 2011. "A Branch and Cut Algorithm for the Location-routing Problem with Simultaneous Pickup and Delivery." *European Journal of Operational Research* 211 (2): 318–332.

Karaoglan, I., F. Altiparmak, I. Kara, and B. Dengiz. 2012. "The Location-routing Problem with Simultaneous Pickup and Delivery: Formulations and a Heuristic Approach." *Omega* 40 (4): 465–477.

Kirkpatrick, S., C. D. Gelatt Jr., and M. P. Vecchi. 1983. "Optimization by Simulated Annealing." *Science* 220: 671–680.

Laporte, G., Y. Nobert, and D. Arpin. 1986. "An Exact Algorithm for Minimizing Routing and Operating Costs in Depot Location." *European Journal of Operational Research* 6 (9): 293–310.

Lin, S., and B. W. Kernighan. 1973. "An Effective Heuristic Algorithm for the Traveling-salesman Problem." *Operations Research* 21: 498–516.

Lin, C. K. Y., C. K. Chow, and A. Chen. 2002. "A Location-routing-loading Problem for Bill Delivery Services." *Computers & Industrial Engineering* 43 (1–2): 5–25.

Lopes, R. B., C. Ferreira, B. S. Santos, and S. Barreto. 2013. "A Taxonomical Analysis, Current Methods and Objectives on Location-routing Problems." *International Transactions in Operational Research* 20 (6): 795–822.

Madsen, O. B. G. 1983. "Methods for Solving Combined Two Level Location-routing Problems of Realistic Dimensions." *European Journal of Operational Research* 12 (3): 295–301.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. 1953. "Equation of State Calculations by Fast Computing Machines." *The Journal of Chemical Physics* 21 (6): 1087–1092.

Min, H. 1989. "The Multiple Vehicle Routing Problem with Simultaneous Delivery and Pick-up Points." *Transportation Research Part A: General* 23 (5): 377–386.

Murty, K. G., and P. A. Djang. 1999. "The U.S. Army National Guard's Mobile Training Simulators Location and Routing Problem." *Operations Research* 47 (2): 175–182.

Nguyen, V.-P., C. Prins, and C. Prodhon. 2012. "A Multi-start Iterated Local Search with Tabu List and Path Relinking for the Two-echelon Location-routing Problem." *Engineering Applications of Artificial Intelligence* 25 (1): 56–71.

Osman, I., and C. Potts. 2003. "Simulated Annealing for the Permutation Flowshop Problem." *Omega* 17 (6): 551–557.

Osman, I. H., and N. A. Wassan. 2002. "A Reactive Tabu Search Meta-heuristic for the Vehicle Routing Problem with Back-hauls." *Journal of Scheduling* 5 (4): 263–285.

Prins, C., C. Prodhon, and R. Wolfler Calvo. 2004. "Nouveaux algorithmes pour le problème de localisation et routage sous contraintes de capacité [New algorithms for the location and routing problem under capacity constraints]." In *MOSIM'04*, edited by A. Dolgui and S. Dauzère-Pérès, 1115–1122. Nantes:Lavoisier, Ecole des Mines de Nantes.

Prins, C., C. Prodhon, and R. Wolfler Calvo. 2006. "Solving the Capacitated Location-routing Problem by a GRASP Complemented by a Learning Process and a Path Relinking." *4OR* 4 (3): 221–238.

Salhi, S., and G. Nagy. 1999. "A Cluster Insertion Heuristic for Single and Multiple Depot Vehicle Routing Problems with Backhauling." *Journal of the Operational Research Society* 50 (10): 1034–1042.

Seçkiner, S. U., and M. Kurt. 2007. "A Simulated Annealing Approach to the Solution of Job Rotation Scheduling Problems." *Applied Mathematics and Computation* 188 (1): 31–45.

Sofianopoulou, S. 1992. "Simulated Annealing Applied to the Process Allocation Problem." *European Journal of Operational Research* 60 (3): 327–334.

Subramanian, A., E. Uchoa, A. A. Pessoa, and L. S. Ochi. 2011. "Branch-and-cut with Lazy Separation for the Vehicle Routing Problem with Simultaneous Pickup and Delivery." *Operations Research Letters* 39 (5): 338–341.

Ting, C.-J., and C.-H. Chen. 2013. "A Multiple Ant Colony Optimization Algorithm for the Capacitated Location Routing Problem." *International Journal of Production Economics* 141 (1): 34–44.

Tuzun, D., and L. I. Burke. 1999. "A Two-phase Tabu Search Approach to the Location Routing Problem." *European Journal of Operational Research* 116: 87–99.

Wassan, N. A., A. H. Wassan, and G. Nagy. 2007. "A Reactive Tabu Search Algorithm for the Vehicle Routing Problem with Simultaneous Pickups and Deliveries." *Journal of Combinatorial Optimization* 15 (4): 368–386.

Watson-Gandy, C., and P. Dohrn. 1973. "Depot Location with Van Salesman – A Practical Approach." *Omega* 1 (3): 321–329.

Yu, V. F., S.-W. Lin, W. Lee, and C.-J. Ting. 2010. "A Simulated Annealing Heuristic for the Capacitated Location Routing Problem." *Computers & Industrial Engineering* 58 (2): 288–299.