Expert Systems with Applications 37 (2010) 4073–4083

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A scatter search algorithm for solving vehicle routing problem with loading cost

Jiafu Tang *, Jun Zhang, Zhendong Pan

Key Laboratory of Integrated Automation of Process Industry of MOE, Northeastern University, Shenyang, PR China

ARTICLE INFO

Keywords:
Vehicle routing problem
Meta-heuristic algorithm
Scatter search
Loading cost

ABSTRACT

The vehicle routing problem (VRP) is an important scientific problem addressed in distribution management. In classical VRP and its variants, the vehicle load is often regarded as a constant during transportation; therefore the loading cost associated with the amount of the load on the vehicle, are neglected in the objective function when optimizing a vehicle routine. However, in real-world, the vehicle load varies from one customer to another in a vehicle route. Thus, the vehicle route without considering the effect of loading cost may lead to sub-optimal routes. In this paper, we investigate VRP with loading cost (VRPLC), which considers the costs associated with the amount of the load on the vehicle when determining the vehicle routes. Considering the features of the VRPLC, a scatter search (SS) is proposed. By introducing customer-oriented three-dimensional encoding method, sweep algorithm and optimal splitting procedure are combined to obtain better trial solutions. The arc combination and improved nearest neighbor heuristic are adopted as a solution combination method and an improvement method to generate and improve new solutions, respectively. Computational experiments were carried out on benchmark problems of capacitated VRP with seven categories of distribution scenarios. The computational results show that the SS is competitive and superior to other algorithms on most instances, and that the VRPLC can more reasonably and exactly formulate the vehicle routing problem with more cost savings than general VRP models.

1. Introduction

Vehicle routing problem (VRP) is an important scientific problem proposed by Dantzig and Ramser (1959) firstly. It can be defined as the problem of designing the optimal routes for a fleet of homogeneous vehicles to serve some geographically scattered customers to minimize the total operation cost. Since then, VRP has received much attention with respects to various aspects of the problem. They are: (1) capacitated vehicle routing problem (CVRP) that is an elementary version of VRP and only considers capacity constraint (Laporte & Nobert, 1983); (2) vehicle routing problem with time windows (VRPTW) that considers the due service time bound of every customer (Bent & Van Hentenryck, 2004); (3) multi-depot vehicle routing problem (MDVRP) that considers multiple depots instead of a single depot (Lim & Wang, 2005); (4) periodic vehicle routing problem (PVRP) that considers the service time as a period instead of a day (Francis & Smilowitz, 2006); (5) split delivery vehicle routing problem (SDVRP) in which customer can be served more than once (Belenguer, Martinez, & Mota, 2000) and so on. The above-mentioned problems are special classes of general VRP and they correspond to some real scenarios of transportation in distribution management. Most of the aforementioned VRPs are formulated as 0-1 integer programming problems (Chu, 2005; Lim & Wang, 2005; Lu & Dessouky, 2004; Mingozzi, Giorgi, & Baldacci, 1999), 0-1 mixed integer programming problems (Bookbinder & Reece, 1988; Tavakkoli, Safaei, Kah, & Rabbani, 2007) and network optimization models (Yi & Ozdamar, 2007).

In general VRP models and its variants, the transportation cost usually includes two parts: fixed cost and variable cost. The fixed part is the dispatching cost for a vehicle, which depends only on the volume of goods loaded in the vehicle and distance of a trip. The variable cost is in proportion to the sum of the distance traveled. In summary of the literature, the aforementioned models for various forms of VRP consider the vehicle load as constant on a whole route, and hence the cost associated with the amount of the load, referred to loading cost hereafter in this paper, is a constant and neglected in the objective function. The loading cost is an important part of charge for consumption of gas that changes due to the amount of the load of the vehicle. However, in real-world distribution activities, the vehicle load varies greatly from one customer to another on a whole route, because the goods will be unloaded when a vehicle visited the customer. The vehicle load will decrease as the vehicle visits customers one by one and finally equals the quantity of demand at the last customer on a trip. In practice, when transporting some special freight, such as hazardous materials, perishable food, livestock or something with special

* Corresponding author. Tel.: +86 24 83673810.
E-mail addresses: jftang@mail.neu.edu.cn (J. Tang), water_zj2000@yahoo.com.cn (J. Zhang), panzhendong@ise.neu.edu.cn (Z. Pan).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.11.027

handling, extra cost can be charged in real-world distribution activity. This part of cost is commonly a function of weight of the load and accounts for about fifty percent of the transportation cost.

In fact, the vehicle load is considered as a kind of variable, called flow variable, to formulate traveling salesman problem and a variety of related transportation problems, e.g. network flow problem (Dumas, Desrosiers, & Soumis, 1991; Gavish & Graves, 1978; Gouveia, 1995). The load variable appears in constraints or appears as a threshold in objective function. Thus, the aforementioned VRP models without considering the loading cost could only provide an approximate and simplified formulation of practical VRP, and subsequently the vehicle routes determined without considering the effect of loading cost may lead to inexact optimal routes.

This paper focuses on a new formulation of VRP, short for VRPLC hereafter in the paper, taking into account the loading cost in the objective function when optimizing vehicle routes. Distinguished from the traditional models of VRP, the loading cost is regarded as a variable in objective function rather than zero. When the loading cost is zero, the VRPLC is CVRP.

CVRP is a NP-hard problem and has been intensively studied. As a generalization of CVRP, VRPLC is a NP-hard problem. There has been no report on solution algorithms for VRPLC in the literature. However, algorithms for solving various CVRP received much attention. The algorithms for solving CVRP can be divided into two categories: exact algorithms and heuristics. The detailed survey of exact algorithms for CVRP can be found in Cordeau, Laporte, Savelsbergh, and Vigo (2007). The largest CVRP instance that has been solved optimally using an exact algorithm is of 134 customers (Fukasawa et al., 2006). With the size of instances increasing, the computation time of exact algorithm becomes intolerable. Thus, heuristic algorithms have received more attention and they are suitable to solve practical instances. The heuristics for solving CVRP and its variants are classified into two categories: classical heuristic and meta-heuristic. The famous classical heuristics are those like CW saving heuristic (Clarke & Wright, 1964), sweep algorithm (Gillett & Miller, 1974), insertion heuristic (Gendreau, Hertz, Laporte, & Stan, 1998) and other iterative improvement heuristics (Tan, Lee, Zhu, & Ou, 2001; Waters, 1987). The classical heuristic algorithms terminate with satisfactory solution quickly and are easy to be implemented; but, in general, the gap between the heuristic solution and the optimal solution is large. A systematic review on classical heuristics can be found in the literatures (Laporte, Gendreau, Potvin, & Semet, 2000; Laporte, 2007; Laporte & Semet, 2002).

Compared with classical heuristic, meta-heuristic algorithms can obtain better solutions than classical heuristics or even global optimal solutions in a reasonable time. The well-known meta-heuristic algorithms include tabu search (TS) (Glover, 1986), simulated annealing (SA) (Kirkpatrick & Gelatt, 1983), genetic algorithm (GA) (Mitchell, 1996), ant colony optimization algorithm (ACO) (Doerner, Hartl, Benkner, & Lucka, 2006) and neural networks (Smith, 1999). TS and SA are based on local search principle (Kytöjoki, Nuortio, Bräysy, & Gendreau, 2007) and usually start searching from one initial solution and explore more promising solutions in the solution region. Some efficient implementation of TS and SA for CVRP can be found in Tavakkoli-Moghaddam et al. (2007) and Brandão (2009). GA and ACO are representative methods grounding in population search principle (Smith, 1999). The meta-heuristics based on population search principle maintain a pool of solutions and update the solutions in the pool by rules. Recently, several ACO algorithms are developed to solve CVRP successfully (Doerner et al., 2006). Over the last 2 year, some meta-heuristic algorithms, like variable neighborhood search (VNS) (Kytöjoki et al., 2007), large neighborhood search (LNS) (Goel & Gruhn, 2008), adaptive large neighborhood search(ALNS) (Pisinger & Ropke, 2007), scatter search (SS) (Russell & Chiang, 2006) are applied to the vehicle routing problem. Gendreau, Laporte, and Potvin (2002) Cordeau, Gendreau, Hertz, Laporte, and Sormany (2005) gave an extensive review of the above-mentioned meta-heuristics in the past decades and recent years.

Among the aforementioned meta-heuristics, scatter search (SS) can be regarded as an efficient algorithms for VRP (Glover, Laguna, & Marti, 2000). SS is a population-based meta-heuristic and introduced by Glover (1977) firstly for solving integer programming. Distinguished from other population-based meta-heuristic, SS operates on some solutions called reference set (Glover, Laguna, & Marti, 2003). It selects more than two solutions of reference set in a systematic way aiming at producing new solutions. In the domain of solving routing problem the SS can yield better results than some often used meta-heuristics (Alegre, Laguna, & Pacheco, 2007; Mota, Campos, & Corberán, 2007; Russell & Chiang, 2006).

In this paper, we study VRP with loading cost (VRPLC), which determines a route in order to minimize the total transportation cost considering the costs incurred by vehicle load in objective function. With the formulation of VRPLC, the vehicle load is viewed as varying from a customer to another one instead of a constant during a trip. Thus, the loading cost brought by the volumes of goods loaded in a vehicle is considered as a part of the transportation cost. Considering the features of VRPLC, a scatter search (SS) algorithm is developed. The SS algorithm adapts customer-oriented three-dimensional encoding and combines the sweep and optimal splitting procedure (Prins, 2004) to construct the initial trial solutions pool. New solutions are generated by combining the arc selected. The improved nearest neighbor method is used to improve new solutions. Seven types of benchmark datasets of CVRP with different node distribution are chosen, in the form of random, cluster (one and several), mixing random and cluster, regulation(dispersive and dense), and similar magnetic field. The experiments to test the performance, stability and sensitivity of parameters of the SS on benchmark problems are conducted.

Following the introduction, the paper is organized as follows. The mathematical model is described in Section 2, along with the problem assumption. In Section 3, combining the features of the model, a SS is developed to solve the VRPLC. In Section 4, computational results of the SS on seven types of benchmark instances are reported. Finally, conclusions are made in Section 5.

2. Mathematical model for VRPLC

Consider a distribution network in which one product is shipped from a depot to a set of customers. The VRPLC can be defined on a graph $G = (V,A)$, where V is vertex set and A is the arc set. The vertex set V includes the depot v_0 and customers $V_c = \{v_1, v_2, \ldots, v_N\}$. The index of the depot is 0 while customers are indexed from 1 to N. $A = \{(v_i, v_j) | v_i,\ v_j \in V,\ i \neq j\}$ is the arc set. A distance matrix $D = (d_{ij})$ is defined on A, which is associated to each arc (v_i, v_j). We assume that D is symmetric and satisfies the triangle inequality. The depot owns homogeneous vehicles, and the vehicle number is assumed to be unlimited. Each customer's demand is known and less than the vehicle capacity. The following notations are used throughout the paper to formulate the model:

N = the number of the customers;
K = the number of the vehicles;
Q = the capacity of the vehicle;
C_d = the cost of traveling per unit-distance by a vehicle, called distance coefficient hereafter;
C_g = the cost of delivering product of per unit-weight and per unit-distance, called load coefficient hereafter;
C_v = the fixed cost of dispatching a vehicle called vehicle coefficient hereafter;
d_{ij} = the distance from node i to j, $i, j = 0,1,2,\ldots,N$;

q_i = the demand of customer i, $i = 1,2,\ldots,N$;
x_{ijk} = a binary variable, which is equal to 1 when arc (i,j) is traversed by vehicle k, otherwise is equal to 0, i, $j = 0,1,2,\ldots,N$, $k = 0,1,2,\ldots,K$;
y_{ijk} = the load of the vehicle k on arc (i,j), the value is nonnegative real number and less than Q when vehicle k pass the arc form i to j, otherwise y_{ijk} is equal to 0, i, $j = 0,1,2,\ldots,N$, $k = 0,1,2,\ldots,K$.

The VRPLC aims to minimize the transportation cost including distance cost, loading cost brought by the weight of goods distributed and the dispatching cost. It is formulated as follows:

$$\text{Min} \quad \cos t = \sum_{i=0}^{N}\sum_{j=0}^{N} d_{ij}\sum_{k=1}^{K} x_{ijk}(C_d + C_g y_{ijk}) + C_v\sum_{j=1}^{N}\sum_{k=1}^{K} x_{0jk} \tag{1}$$

$$\text{Subject to} \quad \sum_{i=1}^{N} q_i \sum_{j=0}^{N} x_{ijk} \leqslant Q, \quad k \in \{1,2,\ldots,K\} \tag{2}$$

$$\sum_{j=0}^{N}\sum_{k=1}^{K} x_{ijk} = 1, \quad i \in \{1,2,\ldots,N\} \tag{3}$$

$$\sum_{i=0}^{N}\sum_{k=1}^{K} \boldsymbol{x}_{ijk} = 1, \quad j \in \{1,2,\ldots,N\} \tag{4}$$

$$\sum_{j=1}^{N} x_{0jk} \leqslant 1, \quad k \in \{1,2,\ldots,K\} \tag{5}$$

$$\sum_{j=1}^{N} x_{0jk} - \sum_{j=1}^{N} x_{j0k} = 0, \quad k \in \{1,2,\ldots,K\} \tag{6}$$

$$\sum_{i\in S}\sum_{j\in S} \boldsymbol{x}_{ijk} \leqslant |S| - 1 \quad \forall S \subseteq V_c,\ 2 \leqslant |S| \leqslant n, \quad k \in \{1,2,\ldots,K\} \tag{7}$$

$$\sum_{\substack{j=0\\ i\neq j}}^{N}\sum_{k=1}^{K} (\boldsymbol{x}_{jik} y_{jik} - x_{ijk} y_{ijk}) = q_i, \quad i \in \{1,2,\ldots,N\} \tag{8}$$

$$0 \leqslant y_{ijk} \leqslant Q x_{ijk}, \quad i,j \in \{0,1,\ldots,N\}, \quad k \in \{1,2,\ldots,K\} \tag{9}$$

$$x_{ijk} \in \{0,1\}, \quad i,j \in \{0,1,\ldots,N\},\ k \in \{1,2,\ldots,K\} \tag{10}$$

The objective function includes the distance cost, the loading cost brought by the weight of freight and the dispatching cost. The constraint (2) is the vehicle capacity constraint. The formula (3)–(5) represent that every customer is visited exactly once and that served only by one vehicle. The constraint (6) guarantees that the vehicles starts from and return to the depot. The constraint (7) is the sub-tour elimination constraint. Eq. (8) shows the logic relationship between the demand of customer i and the vehicle loads on the two arcs linking customer i.

The mathematical formulation of VRPLC can be regarded as the generalization of CVRP, since the model of VRPLC can be transformed to the classical CVRP when the C_g and C_v are equal to 0 and neglecting the constraints (8) and (9). In the next section, a scatter search algorithm is proposed.

3. A scatter search algorithm to solve VRPLC

In recent years, much attention have been attracted on SS for solving various kinds of VRP. Corberán, Fernández, Laguna, and Marti (2002) proposed a SS to solve a real-life problem with multiple objectives. Two different heuristics construct the initial trial solutions in the SS. SWAP and INSERT, two simple exchange procedures are used to improve solutions. The combination method is based on a voting scheme. The experiments testing on real data show that the SS can solve the practical problem efficiently. Russell and Chiang (2006) solved a classical VRPTW by SS. Their SS also adopts two different construction heuristics to generate the initial trial solutions. A common arc method and an optimization-based set covering model are used to combine solutions. A more recent application of SS is Mota et al. (2007), who presented a SS for SDVRP. Local search is adopted as the improvement method. Four kinds of critical clients is defined to produce new solutions.

Most SS algorithms generally use heuristics to obtain initial trial solutions and adopt methods with few runtime to improve solutions. The procedure of generating new solutions is usually developed according to the characteristic of problem, which is the essential of an SS algorithm. However, generating the initial trial solutions and the methods of improving solution are important in SS. Combining the features of the VRPLC, a customer-oriented three-dimensional encoding method, arc combination and solution improvement methods based on the nearest neighbor heuristic are developed, and they are given in detail as follows.

3.1. Customer-oriented three-dimensional encoding method

According to the characteristics of the VRPLC, we design a structure to represent the solution based on the customer. The structure is shown in Fig. 1.

In our implementation, the solution is encoded as an n-dimensional vector of customers. Every entry of the vector denotes customer information that includes three components: the customer number; the vehicle serving the customer, and the position in the sequence of the customer visited. The example of Fig. 1 can tell us that the customer 1 is visited secondly by vehicle 4. However, the solution encoding cannot display the route directly. When the solution is obtained, the route can be known from the customers' information by ordering the vehicle number and the sequence number in turn.

Fig. 2 is an example to show how to translate encoded solution to route. The solution displayed each customer's information in Fig. 2a, which has five customers. By ordering the second and third items as above-mentioned, we can obtain the routes: 0-1-4-0, 0-3-0 and 0-2-5-0 as shown in Fig. 2b, where 0 denote the depot.

3.2. Diversification generation method

The initial trial solution set of SS usually needs solutions with higher quality or better diversity in the solution space. The random initialization based on optimal splitting procedure (RIOSP) is proposed to generate the initial trial solution. RIOSP comprise three steps: (1) select a start point randomly; (2) construct a TSP route by sweep algorithm (Gillett & Miller, 1974) beginning from the start point; (3) apply the optimal splitting procedure (Prins, 2004) to split the TSP. Repeating the above process p times one can build a trial solution set denoted by P. All trial solutions are feasible.

For example, we use the date of benchmark problem (P-n16-k8) of CVRP to demonstrate the process of RIOSP. In Fig. 3a, the algorithm selects the 8th customer randomly as start point and the line from the depot to the 8th customer is set as the polar axis. Then use sweep method to obtain a visiting sequence shown in Fig. 3b, which is 8-3-10-1-12-15-4-11-14-5-7-9-6-13-2. After that, apply the optimal splitting procedure to generate the initial trail solution as shown in Fig. 3c, which is 0-8-0, 0-3-10-0, 0-1-0, 0-12-15-0, 0-4-11-0, 0-14-5-0, 0-7-9-0, 0-6-0, 0-13-0 and 0-2-0 (0 denote the depot). When a route is decided, y_{ijk} the vehicle load

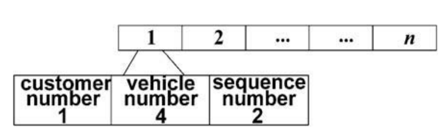

Fig. 1. The solution encoding structure.

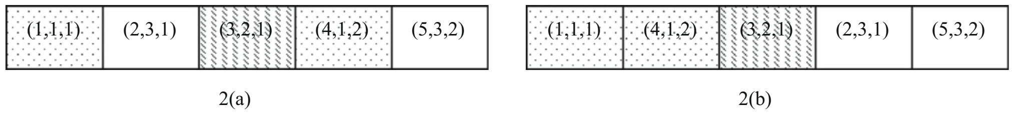

Fig. 2. An example of the conversion from solution to route.

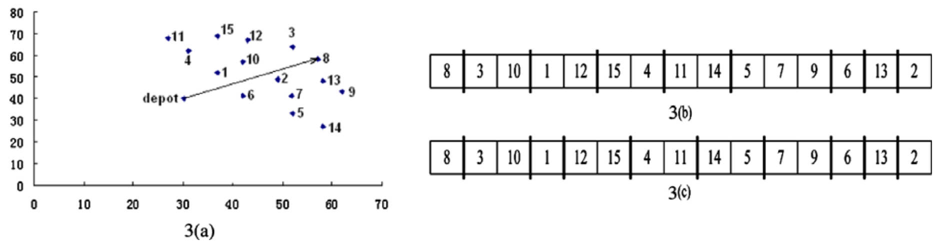

Fig. 3. An example of the optimal splitting procedure.

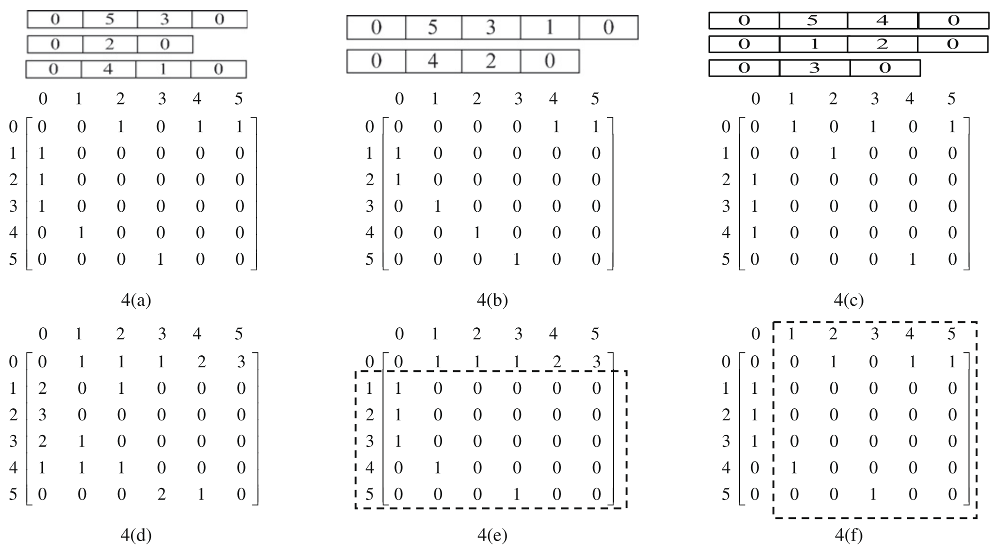

Fig. 4. An example of solution combination method.

variable can be determined. For example, given the route 0-3-10-0 served by vehicle 2, $Q = 35$, $q_3 = 16$ and $q_{10} = 8$. The computational results are $y_{0,3,2} = 24$, $y_{3,10,2} = 8$ and $y_{10,0,2} = 0$. Although y_{ijk} is obtained according to the route, y_{ijk} is considered simultaneously during the route decision.

Diversification generation method can construct initial trial solution set quickly. The start point selected by random enhances the diversity of the solution. The sweep algorithm and the optimal splitting procedure can help obtain the route of high quality fast. For instance, Fig. 3, 0-8-0, 0-3-10-0, 0-1-12-0, 0-15-4-0, 0-11-14-0, 0-5-7-9-0, 0-6-0, 0-13-0 and 0-2-0, is the result of simple splitting procedure with objective value of 3266.4, while the result of optimal splitting procedure is with the objective value of 3072.4.

3.3. Improvement method

The improvement method is used after generating new solutions. The new solutions are the initial trial solutions or solutions generated by combination method. Hence, the improvement method need treat with feasible and infeasible solutions. Improving the solution generated by the solution combination method can affect the reference set updated and a tradeoff between the quality and computational time is needed. Hence, we apply a simple method to complete improvement towards reducing the runtime of SS. The procedure is given below.

1. **If** there are routes of the input solution violating vehicle capacity limit **then**
   For (i = 1 ***to*** the number of the routes which violate vehicle capacity limit)
2. Order all the customers of the route i according to insertion cost.
   do
3. Remove the customer with the highest insertion cost.
   until (satisfy the vehicle capacity limit)
   End for
4. Insert each customer removed in the position with satisfying vehicle capacity limit and having the smallest insertion cost. If no such position exists, apply the Optimal Splitting Procedure to the entire customer unplaced to construct the new route.
   End if
5. Order every route of the solution using the improved nearest neighbor method.

The improved nearest neighbor method select next visiting customer by heuristic rule.

Rule: The vertex j with the highest q_j/d_{ij} value and the lowest d_{ij} is preferential, when vertex i ($i \in V$) select the next vertex visited from $J = \{j|j \in V\}$. q_j equals 0, when $j = 0$.

The rule is derived from the observation of the objective function. For a given route, the total demands of all customers are determined. Given m customers in one route and the corresponding demand set $Q' = \{q_1, q_2, \ldots, q_m\}$, the arcs comprised the route belong to arc set $A' = \{r_1, r_2, \ldots, r_{m+1}\}$. The length of each arc is represented by $d_{r_{i'}}$, $i' \in \{1, 2, \ldots, m+1\}$. Then, the objective function of this route can be computed by following formula:

$$\cos t = C_d(d_{r_1} + d_{r_2} + \cdots + d_{r_{m+1}}) + C_g[(q_1 + q_2 + \cdots + q_m)d_{r_1} + (q_2 + \cdots + q_m)d_{r_2} + \cdots + q_m d_{r_m}] + C_v \quad (11)$$

To reduce the value of objective function, the idea is to reduce the value of each item in the function. The intuitional method is to reduce each element in the first and second item of the function. The third item is constant. The most complex part is the second non-linear item. When a route is determined, the total demand, i.e. $q_1 + q_2 + \cdots + q_m$, is determined. To reduce the value of the second item, we want to make the value of each element in this item be lower. Hence, we hope $q_1 > q_2 > \cdots > q_m$ and $d_{r_1} < d_{r_2} < \cdots < d_{r_m}$. The improved nearest neighbor method can satisfy it. The method may result in the value of $d_{r_{m+1}}$ being high, but it influences on the objective function little.

3.4. Reference set update method

The reference set is the foundation of generating new solutions and is very important part of SS. Reference set update method completes building the initial reference set and updating the reference set when the new solution is created. The detailed descriptions for the two parts of reference set update method are given in the following.

3.4.1. Building initial reference set

As usual, the reference set, *Refset*, consists of two subsets. One is *Refset1* with b_1 higher quality solutions and the other is *Refset2* which includes b_2 solutions with better diversification. To find diverse solutions, how to measure the diversification degree, i.e. distance, between two solutions need be defined. Given two solutions x_1 and x_2 with e_1 and e_2 arcs, respectively, there are e_c common arcs between the two solutions. The distance can be defined as $\boldsymbol{d}(\boldsymbol{x}_1, \boldsymbol{x}_2) = 1 - \frac{2\times \boldsymbol{e}_c}{\boldsymbol{e}_1 + \boldsymbol{e}_2}$. The value of the distance is in the range $[0, 1]$. The degree of the diversification is biggest when the value is equal to 1. If the two solutions are same, the distance value is 0.

Take first 3 solutions with the minimum objective function value from P to build *Refset1*. During determining *Refset2*, calculate the distance between solutions in *Refset* and *P-Refset* and maximize the distance minimum (Glover et al., 2003) to obtain a solution and repeat this process b_2 times.

3.4.2. Updating reference set

Updating *Refset* starts from the new solution generated by combination and improvement method. The basic static update process of the reference set is chosen (Marti, Laguna, & Glover, 2006). The reference set is updated when the objective function value of a new solution is better than the worst solution in the *Refset*.

3.5. Subset generation method

Subset generation method is the foundation to construct new solutions in SS. The subsets are built based on the reference set. The general rule (Glover et al., 2003) to generate subsets is suit for most problems solved by SS. The only difference is the type of the subsets applied in different problems. In our implementation, we use four types of subsets, which are 2-element subsets, 3-element subsets, 4-element subsets and the subsets consisting of the best i (for $i = 5$ to b) elements.

3.6. Solution combination method

Solution combination method is used to create new solutions on the subsets. The basic SS can obtain more than one solutions, but only one new solution is generated in our implementation. The new solution is perhaps infeasible; however, they all satisfy the other constraints except the vehicle capacity limit. The combination method is a method of arc combination with selecting high quality arcs and common arcs. "The arcs appears with a high frequency in elite solutions will have a higher probability of being in an optimal solution" (Rochat & Taillard, 1995). The subset is based on the reference set and the solutions in the reference set are the elite solutions of set *P*. It is the reason of selecting common arcs. The criterion of high quality arcs are those with the minimum distance and according with the rule described in Section 3.3.

To be convenient for calculation, we use a $(1 + N) \times (1 + N)$ 0-1 matrix, $A = (a_{ij})$ $i, j \in 0, 1, \ldots, N$, to represent a solution. If there is an arc from vertex i to j, a_{ij} equals 1. Otherwise a_{ij} equals 0. Each solution can only be represented by one matrix according to the order of customers visited.

The implementation process of combination method can be shown below.

1. Transform all the solutions in a subset to matrixes $A_l = (a_{ij})$, respectively, $l \in [1, num]$, num represent the number of solutions in the subset
2. Add up all the matrixes transformed in step1, $A' = \sum_{l=1}^{num} A_l$
3. **For** i = the second row of A' **to** the row 1 + n of A'
4. Save the column number with the biggest value a_{ij} located in the variable *column*
   If there are more than one column with the same biggest value **then**
5. Update variable *column* with the column number where q_j/d_{ij} with the biggest value is located
   End if
6. Set j to equal the number which is saved in the variable *column*. Then, set a_{ij} equals 1 and others equal 0.
   End for
7. **For** j = the second column of A' **to** the 1 + n column of A'
   If there is no any location of the column j being set in step 3 to step 6 **then**
8. Save the row number which a_{ij} with the biggest value is located in the variable *row*
   If there are more than one column with the same biggest value **then**
9. Update variable *row* with the row number where q_i/d_{ji} with the biggest value is located
   End if
10. Set i to equal the number which is saved in the variable *row*. Then, set a_{ij} equals 1 and others equal 0.
    Else
11. Maintain the location set, the other rows in this column equal 0.
    End if
    End for
12. Traverse the route starting from the distribution center, and cut the relationship between the customers which aren't visited and other customers. Construct a new route between each customer which is not visited and the distribution center.

In the process of combination, the constraints (3)–(6) are met by step 3 to step 11. The step 12 makes the route without the sub-tour.

An example is illustrated to explain the implementation process in Fig. 4.

The example in Fig. 4 is the solution combination process in a subset which includes three solutions. Firstly, we use three matrixes to represent the three solutions, respectively, as shown in Fig. 4a–c. Secondly, by adding the three matrixes, we can obtain the matrix shown in Fig. 4d. Thirdly, we determine the value of each row in the frame. In the frame only one position can be equal to 1 in each row to ensure the out-degree of the customer equals 1. The position is decided according to the rule described from step 3 to step 6 in the implementation process. In Fig. 4d, given the value of q_1/d_{41} is the biggest, so customer 1 will be visited following customer 4. Finally, set the value of each column in the frame of Fig. 4f as described from step 7 to step 11 in the implementation process. The result of the solution combination method is 0-2-0, 0-4-1-0 and 0-5-3-0 where 0 denote the depot.

4. Computational experiments and analysis

In this section, we examine the performance of the proposed SS described above. The SS is coded in JAVA and all the tests were performed on Pentium 4 at 3.0 GHz with 1 GB RAM under the Microsoft Windows XP operation system.

4.1. Introduction of testing problems

We adopt the benchmark instances of CVRP to test the proposed SS. The benchmark problems selected as the testing problems are obtained from the webpage of http://www.neo.lcc.uma.es/radi-aeb/WebVRP/Problem_Instances/instances.html. These testing

Table 1
Description of testing problems.

Testing problem no.	Instance	Authors	Size		Distribution type	Distribution characteristic of benchmark data
			N	Capacity		
P1	P-n76-k4	Augerat et al.	75	350	1	Random distribution
P2	P-n76-k5	Augerat et al.	75	280		
P3	E-n101-k8	Christofides and Eilon	100	200		
P4	E-n101-k14	Christofides and Eilon	100	112		
P5	M-n200-k17	Christofides, Mingozzi and Toth	199	200		
P6	E-n33-k4	Christofides and Eilon	32	8000	2	One cluster distribution with small scale data and bigger vehicle capacity
P7	Breedam-P1[a]	Breedam	100	50	3	Several clusters distribution
P8	Breedam-P2[a]	Breedam	100	100		
P9	F-n135-k7	Fisher	134	2210	4	Mixes the cluster and random distribution
P10	F-n72-k4	Fisher	71	30,000	5	Dispersive regular distribution with the same abscissas or ordinates
P11	GWKC-P1[b]	Golden, Wasil, Kelly and Chao	252	1000	6	Dense regular distribution with the same abscissas or ordinates
P12	GWKC-P2[b]		323	1000		
P13	GWKC-P3[b]		480	1000		
P14	GWKC-P4[b]	Golden, Wasil, Kelly and Chao	240	200	7	Depot locating in the center and the customers distribution like a magnetic field
P15	GWKC-P5[b]		360	200		
P16	GWKC-P6[b]		420	200		

[a] Breedam-P1 and Breedam-P2 are instance 19.
[b] GWKC-P1 is instance 13, GWKC-P2 is instance 10, GWKC-P3 is instance 16, GWKC-P4 is instance 17, GWKC-P5 is instance 19, and GWKC-P6 is instance 20.

Table 2
Comparison of the results for VRPLC among SS, ACOs and best known solutions.

Testing problem no.	Instance	Improved C-W		CVRP	IMMAS[a]		PMMAS[a]		SS[b]	
		$v(s)$	t (s)	$v(s)$	$v(s)$	t (s)	$v(s)$	t (s)	$v(s)$	t (s)
P6	E-n33-k4	622,590	2.7	577,739	613,206	4.3	**467,149**	4.5	467,302	0.1
P10	F-n72-k4	600,671	49.5	589,297	598,322	11.3	314,446	11.4	**313,368**	0.8
P1	P-n76-k4	24,346	59.5	21,349	27,706	11.6	11,885	12.1	**11,112**	0.5
P2	P-n76-k5	19,912	59.4	19,690	22,230	11.5	12,018	12.0	**11,130**	0.5
P3	E-n101-k8	18,016	194.5	17,664	21,140	18.1	13,914	18.7	**12,624**	0.9
P4	E-n101-k14	15,102	183.8	14,898	16,335	21.6	13,861	18.9	**12,715**	2.8
P7	Breedam-P1	12,181	184.8	–	12,496	18. 2	12,476	19.0	**11,867**	7.7
P8	Breedam-P2	11,093	184.1	–	11,680	18.3	11,600	18.1	11,515	4.9
P9	F-n135-k7	250,795	592.6	243,786	261,070	29.9	162,142	31.3	**161,282**	1.5
P5	M-n200-k17	30,285	995.2	–	32,846	66.1	25,264	67.1	**25,001**	5.1
P14	GWKC-P4	17,723	2564.8	18,084	18,239	87.7	17,100	96.5	**16,442**	24.4
P11	GWKC-P1	94,196	2451.9	89,415	96,830	96.6	71,451	101.0	**63,561**	15.5
P12	GWKC-P2	82,539	7310.6	75,781	86,469	156.6	59,635	164.8	**57,144**	34.0
P15	GWKC-P5	33,232	13,173.6	33,639	37,756	193.5	**30,721**	209.9	31,050	48.6
P16	GWKC-P6	44,343	24,775.9	43,897	47,681	262.9	**40,147**	306. 2	40,742	177.4
P13	GWKC-P3	175,952	99,552.1	165,053	188,155	341. 3	132,766	356.1	**121,641**	64.1

[a] Coded in Matlab, seconds using Pentium 4, 2.66 GHz with 512 MB RAM under the Microsoft Windows XP operating system.
[b] Time unit is second.

problems are classified into seven types. Each type represent a scenario of customer's distribution, hence they have different distribution characteristics. These testing problems are from small sizes with 32 customers to large size with 480 customers and have different vehicle capacity. Detailed information of these testing problems is shown in Table 1. The parameter values of the model are set with $C_d = 1.5$, $C_g = 0.2$ and $C_v = 100$. The parameter values are determined according to an investigation in some local transportation company.

All the benchmark problems only have vehicle capacity restriction.

Table 3
The results of SS over 50 independent runs.

Instance	MAX		MIN		AVG
	$v(s)$	Avg gap (%)	$v(s)$	Avg gap (%)	$v(s)$
E-n33-k4	467,549	0.1	467,302	0.0	467,307
F-n72-k4	313,664	0.0	313,368	0.0	313,516
P-n76-k4	11,514	1.7	11,112	1.9	11,326
P-n76-k5	11,524	1.7	11,130	1.8	11,334
E-n101-k8	13,012	0.7	12,624	2.3	12,923
E-n101-k14	13,179	1.4	12,715	2.1	12,993
Breedam-P1	12,103	1.9	11,867	0.1	11,875
Breedam-P2	11,819	0.8	11,515	1.8	11,722
F-n135-k7	161,599	0.0	161,282	0.2	161,554
M-n200-k17	25,703	1.2	25,001	1.5	25,388
GWKC-P4	17,032	2.0	16,442	1.5	16,691
GWKC-P1	65,015	1.2	63,561	1.0	64,233
GWKC-P2	59,327	2.2	57,144	1.6	58,053
GWKC-P5	31,806	1.1	31,050	1.3	31,446
GWKC-P6	42,171	1.8	40,742	1.7	41,432
GWKC-P3	124,488	0.9	121,641	1.4	123,344

4.2. The comparison analysis of the SS

To test the performance of SS, the best known solutions of the benchmark problems, improved C-W algorithm and two Ant Colony Optimization (ACO) algorithms are used to compare with SS. One ACO algorithm is Improved MAX-MIN Ant System (IMMAS) and the other is Partition based MAX-MIN Ant System (PMMAS). The improved C-W algorithm adopts the objective function of VRPLC and considers the influence of sequence of visiting. IMMAS is based on the MAX-MIN Ant System for TSP proposed by Stützle and Hoos (2000). It solves the VRPLC by partitioning the TSP route. The partition only considers vehicle capacity constraint. The partition of the PMMAS is by optimal splitting (Prins, 2004).

The comparison results are reported in Table 2, the column of "$v(s)$" represents the objective function value of the solution and "t" represents the computational time. The column of "CVRP" is the objective function value of the VRPLC corresponding to the best known solutions of CVRP problem. The last column is the best solution of SS over 50 independent runs. The computational results indicate that the SS is much better than other four algorithms for the 12 testing problems among the 16 testing problems. As for the testing problems P6, P8, P15 and P16, the solution gap between the SS and the best solutions of other algorithm is not more than 3.6%, while the other algorithm consumed more time. Thus, one can conclude that the SS is competitive both in quality and computational time for most instances in Table 2.

From the perspective of running time, the testing problems with the same customers' distribution and different vehicle capacity, computational time reduces along with increase of vehicle capacity. That is because in the problem with the relatively small vehicle capacity, more feasible solutions are generated. The results in Table 2 also show that for the instances with the same

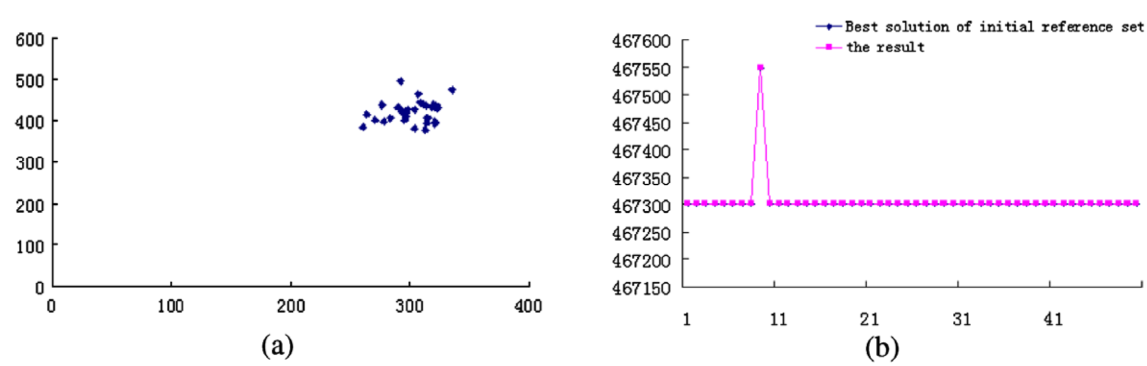

Fig. 5. Data set of E-n33-k4 and the corresponding results over 50 independent runs.

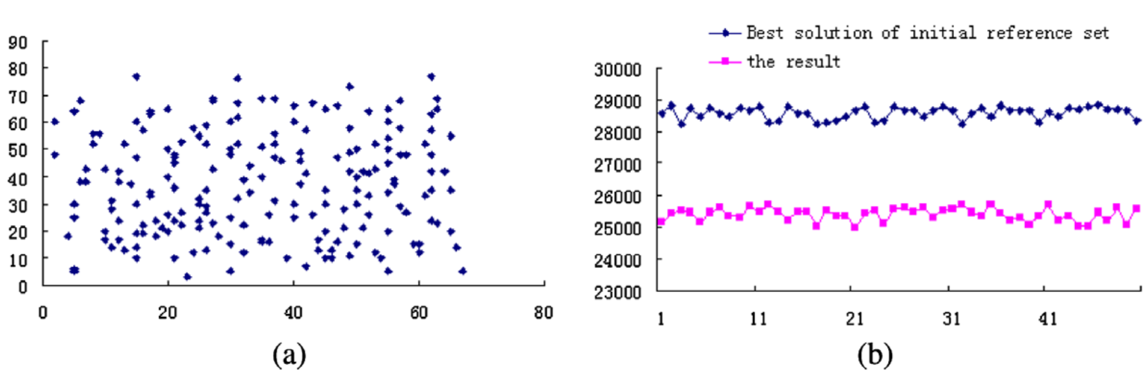

Fig. 6. Data set of M-n200-k17 and the corresponding results over 50 independent runs.

distribution, the computational time increases with the size of the customers on the whole. From the results in Table 2, we can conclude that algorithms with best known solutions of CVRP are not suitable for the VRPLC. It can be observed that the methods for CVRP cannot produce better solution than SS.

4.3. The experiment on the stability of the SS

To test the stability of the SS, we run the SS algorithm independently over 50 times for each testing problem. For each testing problem, the best solution (MIN), worst solution (MAX) and the

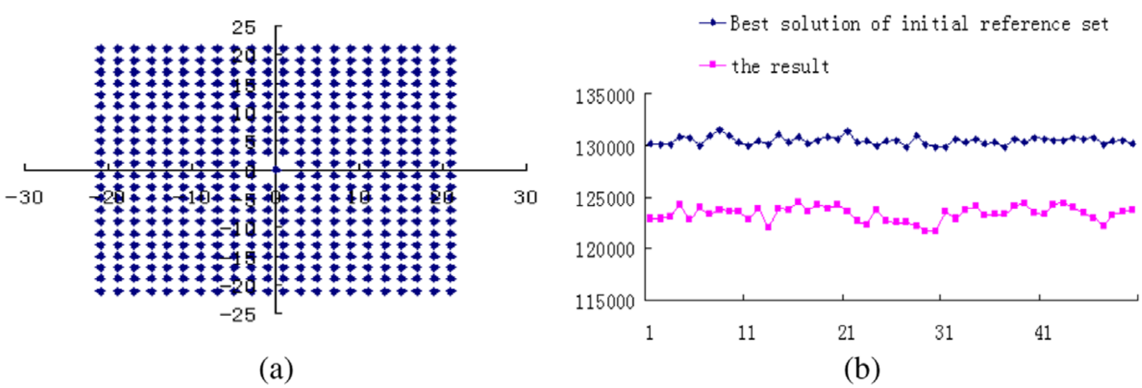

Fig. 7. Data set of GWKC-P3 and the corresponding results over 50 independent runs.

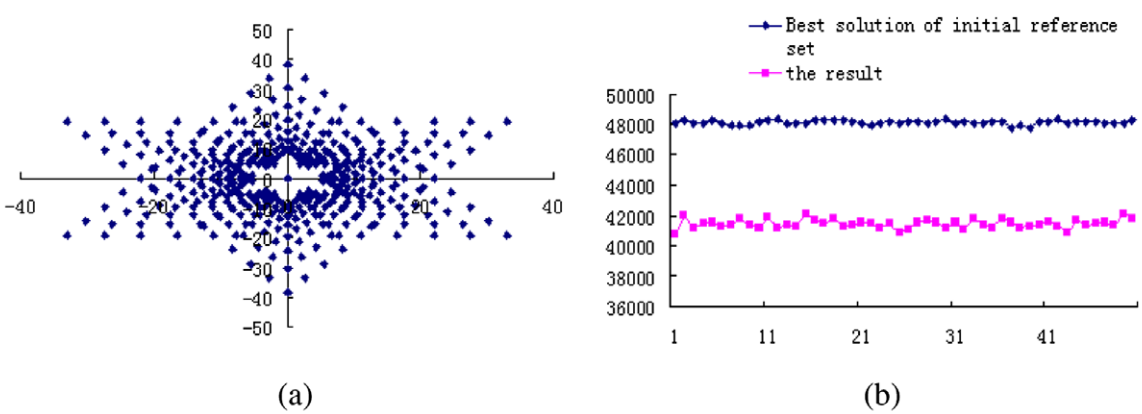

Fig. 8. Data set of GWKC-P6 and the corresponding results over 50 independent runs.

Table 4
The results of SS with different parameter setting of model.

Instance	C_d	C_g	C_v	CVRP	SS		
					MAX	MIN	AVG
E-n101-k14	1.5	0.2	100	14,898	13,075	**12,863**	12,986
	1.0	1.0	100	61,851	44,523	**44,215**	44,362
	1.0	0.0	100	**2483**	2830	2700	2752
	1.5	0.2	0	13,498	11,276	**11,029**	11,176
	1.0	1.0	0	60,451	40,200	**40,145**	40,177
	1.0	0.0	0	**1083**	1342	1288	1316
	1.5	0.2	500	20,498	19,566	**18,742**	19,112
	1.0	1.0	500	67,451	55,399	**54,047**	54,791
	1.0	0.0	500	**8083**	8447	8306	8356
GWKC-P1	1.5	0.2	100	89,415	64,661	**63,925**	64,216
	1.0	1.0	100	430,549	276,008	**275,038**	275,506
	1.0	0.0	100	**3581**	4050	3905	3968
	1.5	0.2	0	86,715	57,381	**57,097**	57,222
	1.0	1.0	0	427,849	264,396	**264,230**	264,334
	1.0	0.0	0	**881**	1277	1170	1222
	1.5	0.2	500	100,215	85,502	**81,474**	83,494
	1.0	1.0	500	441,349	308,151	**303,687**	305,376
	1.0	0.0	500	14,381	14,594	**14,281**	14,445

*Time unit is second. MAX, MIN and AVG are the maximum, minimum and average values of the solutions among the 50 independent runs.

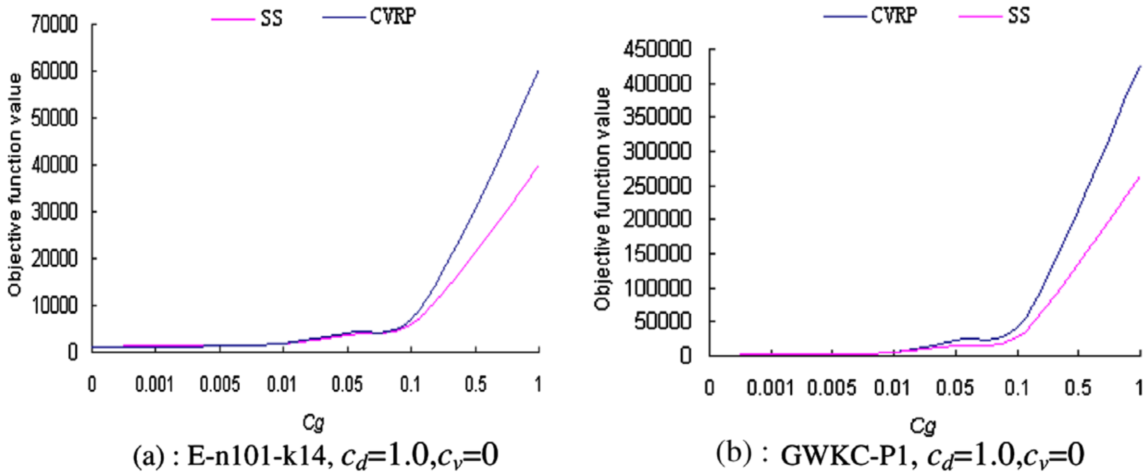

Fig. 9. Total costs varying with load cost coefficient under SS and best solution of CVRP.

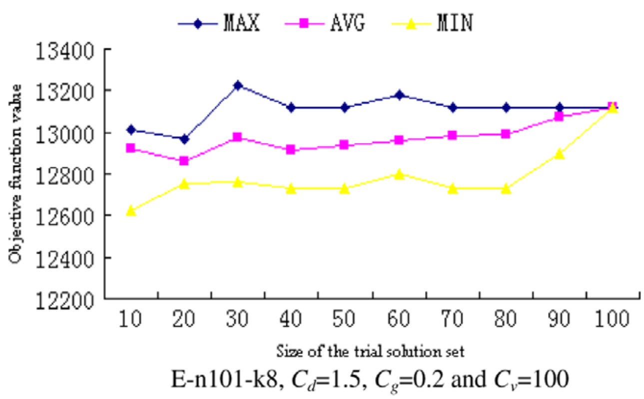

Fig. 10. The total costs varying with the size of trial solution set.

average values of the objective function (AVG) are given in Table 3. The ''Avg gap'' is defined as ((MAX − AVG)/AVG) × 100% in the ''Max'' column and as ((AVG − MIN)/AVG) × 100% in the ''MIN'' column. The results in Table 3 indicates that the gaps either between maximum and average or between minimum and average are not more than 3% among the testing problems, which lead us to conclude that the SS is stable.

4.4. The improvement effect of SS

This part of experiments is carried out to illustrate the roles of the improvement method by comparing the best trial solution with the final solution. Four typical distributions are selected to observe the performance of SS. Figs. 5–8 report the best trial solution and final result over 50 independent runs. The left ones show the data distribution of the benchmark instances. From the result in Fig. 5b, it can be observed that the initial best solution is not improved further when the data size is small. Owing to the randomization of SS, there exists one solution distinguished from others. However, from Figs. 6–8, it indicates that the improved rates of the other three instances with large sizes data sets are about 10%, 5% and 14%, respectively. The effect of improvement is not good for the regular distribution.

4.5. The influence of cost coefficients variation on solution

This part of experiments is to illustrate how the cost coefficients affect the solution to demonstrate that VRPLC is more reasonable and practical than VRP in formulating distribution management. $C_d = 1.0,\ 1.5$, $C_g = 0,\ 0.001,\ 0.005,\ 0.01,\ 0.05,\ 0.1,\ 0.5,\ 1.0$ and $C_v = 0,\ 100,\ 500$ are selected and the experiments are done on two benchmark instances, E-n101-k14 and GWKC-P1 . When C_g and C_v are equal to 0, VRPLC is transformed to the classical CVRP.

Table 5
The results of SS with different size initial solution sets.

Instance	Size of initial trail solution = 20			Size of initial trail solution = 30		
	MAX	MIN	AVG	MAX	MIN	AVG
E-n33-k4	467,302	467,302	467,302	467,302	467,302	467,302
F-n72-k4	313,648	313,418	313,542	313,573	313,367	313,452
P-n76-k4	11,514	11,249	11,371	11,391	11,261	11,365
P-n76-k5	11,390	11,233	11,304	11,390	11,355	11,385
E-n101-k8	12,968	12,754	12,857	13,226	12,765	12,977
E-n101-k14	13,072	12,732	12,932	13,197	12,810	13,026
Breedam-P1	11,866	11,866	11,866	11,866	11,866	11,866
Breedam-P2	11,819	11,685	11,772	11,819	11,685	11,795
F-n135-k7	161,599	161,458	161,551	161,598	161,458	161,585
M-n200-k17	25,534	25,185	25,340	25,574	24,976	25,268
GWKC-P4	16,871	16,583	16,703	16,792	16,469	16,670
GWKC-P1	64,569	63,762	64,096	64,814	63,868	64,317
GWKC-P2	57,996	57,405	57,760	58,498	57,113	57,899
GWKC-P5	31,642	31,036	31,485	31,581	31,092	31,383
GWKC-P6	41,917	41,315	41,586	41,850	41,077	41,384
GWKC-P3	123,872	122,582	123,131	123,649	121,675	122,850

The best known solution of CVRP and the solution of SS under different values of parameters are given in Table 4. The total cost varying with load coefficients for the instance E-n101-k14 and GWKC-P1 under CVRP and SS is shown in Fig. 9. It is observed from Table 4 that the best known solutions of CVRP are poorer than SS when C_g is not equal to 0. In Fig. 9, the ''CVRP'' represent the costs of the best known solution of CVRP, and the ''SS'' represent the proposed SS algorithm. Fig. 9 shows that, when the value of C_g is less than 0.05, the cost of ''CVRP'' is less than the ''SS''. It indicates that the load cost affects the transportation cost slightly, whereas the costs of ''SS'' are much less than ''CVRP'' with the value of C_g increasing. It means that considering the factor of load cost can lead a reduction of the total cost. The results demonstrate the SS is more suitable for VRPLC. As a consequence, this SS for VRPLC is not good for solving general CVRP, if there is no any modification. The experiments demonstrate that VRPLC not only can more reasonably and exactly formulate the vehicle routing problem, but also can provide better solution than VRP.

4.6. Experiments on solution varying with the size of initial trial solution

Reference sets is an important concept of SS, the initial trial solution sets affect the building of initial reference sets. So we test the effect of different size of the initial trial solution sets on the solution in this section. Fig. 10 depicts the solutions varying with different sizes of the initial trial solution. Because the results in Table 3 are obtained when the size of the reference set equals 10. Considering Table 3, Table 5 and Fig. 10 together, it is concluded that the sizes have no considerable effect on the solution.

5. Conclusions

Taking into account the loading cost in objective function, a new formulation of vehicle routing problem (VRPLC) is proposed. A scatter search algorithm is developed for solving VRPLC in this paper. Distinguished from the traditional VRP models, the vehicle load is viewed as a variable rather than a constant from one customer to another in a trip. Experimenting on some benchmark instances demonstrated that VRPLC can more reasonably and exactly formulate the vehicle routing problem, but also can provide cost saving than VRP formulation.

Acknowledgements

The paper was financially supported by the Natural Science Foundation of China (NSFC 70625001, 70721001), 973 Program (2009CB320601), and 111 project of Ministry of Education (MOE) in China with number B08015. Those supports are gratefully acknowledged.

References

Alegre, J., Laguna, M., & Pacheco, J. (2007). Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts. *European Journal of Operational Research, 179*(3), 736–746.

Belenguer, J. M., Martinez, M. C., & Mota, E. (2000). Lower bound for the split delivery vehicle routing problem. *Operations Research, 48*(5), 801–810.

Bent, R., & Van Hentenryck, P. (2004). A two-stage hybrid local search for the vehicle routing problem with time windows. *Transportation Science, 38*(4), 515–530.

Bookbinder, J. H., & Reece, K. E. (1988). Vehicle routing considerations in distribution system design. *European Journal of Operational Research, 37*(2), 204–213.

Brandão, J. (2009). A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research, 195*(3), 716–728.

Chu, C. W. (2005). A heuristic algorithm for the truckload and less-than-truckload problem. *European Journal of Operational Research, 165*(3), 657–667.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*(4), 568–581.

Corberán, A., Fernández, E., Laguna, M., & Marti, R. (2002). Heuristic solutions to the problem of routing school buses with multiple objectives. *Journal of the Operational Research Society, 53*(4), 427–435.

Cordeau, J. F., Laporte, G., Savelsbergh, M. W. P., & Vigo, D. (2007). Vehicle routing. In C. Barnhart & G. Laporte (Eds.), *Transportation, handbooks in operations research and management science* (pp. 367–428). Amsterdam: Elsevier.

Cordeau, J. F., Gendreau, M., Hertz, A., Laporte, G., & Sormany, J.-S. (2005). New heuristics for the vehicle routing problem. In A. Langevin & D. Riopel (Eds.), *Logistics systems: Design and optimization* (pp. 279–297). New York: Springer.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science, 6*(1), 80–91.

Doerner, K. F., Hartl, R. F., Benkner, S., & Lucka, M. (2006). Parallel cooperative Savings based Ant colony optimization – Multiple search and decomposition approaches. *Parallel Processing Letters, 16*, 351–369.

Dumas, Y., Desrosiers, J., & Soumis, F. (1991). The pickup and delivery problem with time windows. *European Journal of Operational Research, 54*(1), 7–22.

Francis, P., & Smilowitz, K. (2006). Modeling techniques for periodic vehicle routing problems. *Transportation Research Part B: Methodological, 40*(10), 872–884.

Fukasawa, R., Lysgaard, J., De Aragao, M. P., Reis, M., Uchoa, E., & Werneck, R. F. (2006). Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming, 106*(3), 491–511.

Gavish, B., & Graves, S. C. (1978). *The traveling salesman problem and related problems*.

Gendreau, M., Hertz, A., Laporte, G., & Gendreau, M. (1998). A generalized insertion heuristic for the traveling salesman problem with time windows. *Operations Research, 46*(3), 330–346.

Gendreau, M., Laporte, G., & Potvin, J. Y. (2002). Metaheuristics for the capacitated VRP. In P. Toth & D. Vigo (Eds.), *The vehicle routing problem* (pp. 129–154). Philadelphia: SIAM Monograph on Discrete Mathematics and Applications.

Gillett, B. E., & Miller, L. R. (1974). A heuristic algorithm for the vehicle dispatch problem. *Operations Research, 22*(2), 340–349.

Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences, 8*(1), 156–166.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research, 13*(5), 533–549.

Glover, F., Laguna, M., & Marti, R. (2000). Fundamentals of scatter search and path relinking. *Control and Cybernetics, 29*(3), 652–684.

Glover, F., Laguna, M., & Marti, R. (2003). Scatter search. In G. Ashish & T. Shigeyoshi (Eds.), *Advances in evolutionary computing: Theory and applications* (pp. 519–537). New York: Springer-Verlag.

Goel, A., & Gruhn, V. (2008). A general vehicle routing problem. *European Journal of Operational Research, 191*(3), 650–660.

Gouveia, L. (1995). A result on projection for the vehicle routing problem. *European Journal of Operational Research, 85*(3), 610–624.

Kirkpatrick, S., & Gelatt, C. (1983). Optimisation by simulated annealing. *Science, 20*, 671–680.

Kytöjoki, J., Nuortio, T., Bräysy, O., & Gendreau, M. (2007). An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Computers and Operations Research, 34*(9), 2743–2757.

Laporte, G. (2007). What you should know about the vehicle routing problem. *Naval Research Logistics, 54*(8), 811–819.

Laporte, G., Gendreau, M., Potvin, J. Y., & Semet, F. (2000). Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research, 7*(4–5), 285–300.

Laporte, G., & Nobert, Y. (1983). A branch and bound algorithm for the capacitated vehicle routing problem. *OR Spektrum, 5*(2), 77–85.

Laporte, G., & Semet, F. (2002). Classical heuristics for the capacitated VRP. In P. Toth & D. Vigo (Eds.), *The vehicle routing problem* (pp. 109–128). Philadelphia: SIAM Monograph on Discrete Mathematics and Applications.

Lim, A., & Wang, F. (2005). Multi-depot vehicle routing problem: A one-stage approach. *IEEE Transactions on Automation Science and Engineering, 2*(4), 397–402.

Lu, Q., & Dessouky, M. (2004). An exact algorithm for the multiple vehicle pickup and delivery problem. *Transportation Science, 38*(4), 503–514.

Marti, R., Laguna, M., & Glover, F. (2006). Principles of scatter search. *European Journal of Operational Research, 169*(2), 359–372.

Mingozzi, A., Giorgi, S., & Baldacci, R. (1999). Exact method for the vehicle routing problem with backhauls. *Transportation Science, 33*(3), 315–329.

Mitchell, M. (1996). *An introduction to genetic algorithms*. Bradford Books.

Mota, E., Campos, V., & Corberán, Á. (2007). A new metaheuristic for the vehicle routing problem with split demands. In C. Cotta & J. van Hemert (Eds.), *Lecture Notes in Computer Science* (pp. 121–129). Springer Valencia.

Pisinger, D., & Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers and Operations Research, 34*(8), 2403–2435.

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers and Operations Research, 31*(12), 1985–2002.

Rochat, Y., & Taillard, É (1995). Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics, 1*(1), 147–167.

Russell, R. A., & Chiang, W. C. (2006). Scatter search for the vehicle routing problem with time windows. *European Journal of Operational Research, 169*(2), 606–622.

Smith, K. A. (1999). Neural networks for combinatorial optimization: A review of more than a decade of research. *INFORMS Journal on Computing, 11*, 15–34.

Stützle, T., & Hoos, H. H. (2000). MAX-MIN ant system. *Future Generation Computer System, 16*(8), 889–914.

Tan, K. C., Lee, L. H., Zhu, Q. L., & Ou, K. (2001). Heuristic methods for vehicle routing problem with time windows. *Artificial Intelligence in Engineering, 15*(3), 281–295.

Tavakkoli, M. R., Safaei, N., Kah, M. M. O., & Rabbani, M. (2007). A new capacitated vehicle routing problem with split service for minimizing fleet cost by simulated annealing. *Journal of the Franklin Institute, 344*(5), 406–425.

Waters, C. D. J. (1987). A solution procedure for the vehicle scheduling problem based on iterative route improvement. *Journal of the Operational Research Society, 38*(9), 833–839.

Yi, W., & Ozdamar, L. (2007). A dynamic logistics coordination model for evacuation and support in disaster response activities. *European Journal of Operational Research, 179*(3), 1177–1193.