Applied Soft Computing 24 (2014) 284–290

Contents lists available at ScienceDirect

Applied Soft Computing

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

Multi-start simulated annealing heuristic for the location routing problem with simultaneous pickup and delivery

Vincent F. Yu [a], Shih-Wei Lin [b,*]

[a] Department of Industrial Management, National Taiwan University of Science and Technology, 43, Section 4, Keelung Road, Taipei 106, Taiwan
[b] Department of Information Management, Chang Gung University, 259 Wen-Hwa 1st Road, Kwei-Shan, Taoyuan 333, Taiwan

ARTICLE INFO

Article history:
Received 24 February 2013
Received in revised form 3 May 2014
Accepted 18 June 2014
Available online 26 June 2014

Keywords:
Location routing problem
Simultaneous pickup and delivery
Metaheuristics
Multi-start simulated annealing

ABSTRACT

The location routing problem with simultaneous pickup and delivery (LRPSPD) is a new variant of the location routing problem (LRP). The objective of LRPSPD is to minimize the total cost of a distribution system including vehicle traveling cost, depot opening cost, and vehicle fixed cost by locating the depots and determining the vehicle routes to simultaneously satisfy the pickup and the delivery demands of each customer. LRPSPD is NP-hard since its special case, LRP, is NP-hard. Thus, this study proposes a multi-start simulated annealing (MSA) algorithm for solving LRPSPD which incorporates multi-start hill climbing strategy into simulated annealing framework. The MSA algorithm is tested on 360 benchmark instances to verify its performance. Results indicate that the multi-start strategy can significantly enhance the performance of traditional single-start simulated annealing algorithm. Our MSA algorithm is very effective in solving LRPSPD compared to existing solution approaches. It obtained 206 best solutions out of the 360 benchmark instances, including 126 new best solutions.

Introduction

Locating depots and planning vehicle routes are two critical issues in the design of distribution systems. These two issues are often tackled seperately, leading to suboptimal configurations for distribution systems [1]. There has thus been a growing interest in the location routing problem (LRP) over the last few decades. The LRP simultaneously determines the facility location problem (FLP) and the vehicle routing problem (VRP) to satisfy each customer's demand for service in a way that minimizes the sum of transportation costs, depot opening costs, and vehicle fixed costs. LRP has applications across a wide variety of fields including newspaper delivery [2,3], drink distribution [4], bill delivery [5], military equipment [6], and retailing [7]. However, it fails to consider another important aspect of the supply chain network: the flow in a supply chain network may be bidirectional. That is, customers may have two types of demands: pickup demand and delivery demand. For example, as more and more customers demand quick after-sales service when products fail, consumer electronics distributors and manufacturers need to both deliver products and parts to retailers and collect failed or recycled products from them. Given this, Karaoglan et al. [8] proposed the location routing problem with simultaneous pickup and delivery (LRPSPD), an extension of the LRP and the vehicle routing problem with simultaneous pickup and delivery (VRPSPD). VRPSPD considers the situation where customers have both pickup and delivery demands, a situation which occurs in many industries such as book distribution [9], laundry service in hotels [10], and soft drink industry [11]. In LRPSPD, customers also have both pickup and delivery demands. These demands are satisfied by vehicles dispatched from multiple potential depots, instead of a single depot as in VRPSPD. LRPSPD has many applications. For example, in the beverage and grocery store chain industries [12], beverage manufacturers not only distribute beverages but also collect empty bottles, while grocery store chains distribute merchandise to and retrieve empty pallets from customers at the same time.

LRPSPD can be stated as follows. There is a set of customers each of which has known coordinates, pickup demand, and delivery demand, and a set of potential depot locations each of which has known coordinates and fixed capacity. The objective is determining locations of open depots and service routes of vehicles to minimize total cost which consists of travel cost of vehicles, opening cost of depots, and fixed cost of vehicles, while satisfying all customers' pickup and delivery demands. Each customer is assigned to an open depot and served exactly once by a vehicle dispatched from the depot. All vehicle routes must originate from and end at the same depot. The total amount of freight in a vehicle cannot exceed

* Corresponding author. Tel.: +886 3 2118800x5811; fax: +886 3 2118020.
E-mail addresses: vincent@mail.ntust.edu.tw (V.F. Yu), swlin@mail.cgu.edu.tw (S.-W. Lin).

http://dx.doi.org/10.1016/j.asoc.2014.06.024
1568-4946/

the vehicle's capacity at any time. The freight includes remaining delivery demand to be satisfied and the cumulated pickup demand satisfied by the vehicle. Total pickup demand or delivery demand of the customers assigned to an open depot cannot exceed the depot's capacity.

The LRPSPD problem is an NP-hard problem since its special case LRP is NP hard [13]. Therefore, it is unlikely that the optimal solution to a large-scale LRPSPD instance can be obtained within a reasonable amount of time. Thus, meta-heuristics seem to be the only viable way for solving large-scale LRPSPD instances. Simulated annealing based heuristics have been successfully applied to problems in a wide variety of fields [14–20]. This study proposes a multi-start simulated annealing (MSA) algorithm for solving the LRPSPD. MSA combines the advantages of the simulated annealing algorithm with the multi-start hill climbing strategy and has been successfully applied to various hard combinatorial optimization problems [21–24].

The remainder of this paper is organized as follows. 'Literature review' section reviews relevant literature. 'Multi-start simulated annealing algorithm for LRPSPD' section describes the proposed MSA algorithm. 'Experimental results' section presents empirical experiments to assess the performance of the proposed MSA algorithm. Finally, 'Conclusions and future research' section concludes the paper with and points out future research directions.

Literature review

Since LRPSPD is closely related to VRPSPD and LRP [8], we first review prior studies on VRPSPD. VRPSPD is a special case of LRPSPD in which customers with both pickup and delivery demands are served by a single depot. We then discuss LRP where customers have only one type of demand and may be served by any one of the potential depots. Next we discuss LRPSPD in which customers have both pickup and delivery demands and may be served by one of the potential depots. Last, we examine the simulated annealing (SA) heuristic and its applications.

Min [9] introduced the VRPSPD to model a book distribution network consisting of several libraries. The problem was solved by a three-stage approach. First, customers are divided into several clusters so that the total pickup demand and total delivery demand are both less than the vehicle's capacity. Then a vehicle is assigned to each cluster. Finally, for each cluster, the orders of customers in which they are serviced are determined. Dethloff [11] developed an insertion-based heuristic for solving VRPSPD which is based on the clustering first, routing second approach. The insertions are performed based on travel distance, residual capacity, and other factors.

Many metaheuristics for solving VRPSPD have been proposed in the literature. For example, Gajpal and Abad [25] proposed an ant colony system (ACS) for VRPSPD. Çatay [10] developed a saving based ant algorithm to solve VRPSPD. Goksal et al. [26] solved VRPSPD by a hybrid discrete particle swarm optimization (PSO) with variable neighborhood descent.

The vehicle routing problem with backhauling (VRPB) is also closely related to LRPSPD. In VRPB, customers may have pickup demand, delivery demand, or both. Vehicles must finish all deliveries before beginning to pick up freight on their way back to the depot. Wassan et al. [27] solved VRPB using a two-stage approach. Clarke and Wright saving algorithm [28] was used to construct an initial solution in the first stage. Then a reactive tabu search (TS) algorithm, originally proposed by Osman and Wassan [29], was applied to improve the initial solution.

LRP minimizes total distribution network costs by determining which potential depots should be opened, how many vehicles should be operated, and how the operated vehicles should serve all customers under routing and capacity constraints. Tuzun and Burke [30] proposed a TS-based two-phase solution approach for LRP. In phase 1, a TS is used to search for a good facility configuration. Then in phase 2, another TS is performed to find a good routing corresponding to the configuration. Wu et al. [31] developed a heuristic based on SA to solve LRP. A tabu list was used in the SA to avoid cycling. Liu and Lee [32] constructed a two-phase heuristic for the LRP with inventory control decisions. In phase 1, a route-first, location-allocation second approach was used to find an initial solution. The initial solution is then improved in the second phase. Prins [33] introduced a two-phase approach for the LRP. The first phase performed a greedy randomized adaptive search procedure (GRASP) based on an extended and randomized version of the Clarke and Wright savings heuristic. A learning process was used in the depot selection strategy in this phase. The phase 2 executes a path relinking method to generate new solutions. Duhamel [34] solved LRP using GRASP with an evolutionary local search to search within two solution spaces: giant tours without trip delimiters and LRP solutions. Barreto [35] developed a distribution-first, location-second heuristic for solving LRP. First, all customers are grouped in clusters so that the total customer demand in each cluster is under vehicle capacity. Then a traveling salesman problem (TSP) is solved for each cluster to determine open depots. These routes are subsequently improved and then assigned to open depots. Many popular metaheuristics have been applied to LRP including SA [15], iterated local search (ILS) [36] and genetic algorithm (GA) [37].

Karaoglan et al. [8] developed two mathematical models of LRPSPD, a node-based model and a flow-based model. They also proposed a hybrid of branch and cut (BC) method and SA (BC-SA) for solving LRPSPD. Karaoglan et al. [12] proposed two SA heuristics to solve LRPSPD. The two heuristics employed different methods for generating initial solutions.

In 1953 Metropolis et al. [38] presented the SA algorithm, an extension of the Markov Chain Monte Carlo algorithm. SA is motivated by the physical analogy between annealing solid metals and optimizing a problem. Kirkpatrick et al. [39] made the first attempt to apply SA to solving optimization problems. Since then, SA has become a popular metaheuristic for solving hard combinatorial problems due to its capability of escaping from local optima. It has been applied to solving routing problems, such as TSP [40] and LRP [15], allocation problems [41], flow shops [42], and job rotation scheduling [43], among others. This study proposes an MSA algorithm based on SA and the multi-start strategy to solve LRPSPD.

Multi-start simulated annealing algorithm for LRPSPD

SA typically starts with a randomly generated initial solution. At each iteration, the algorithm selects a new solution from one of the neighborhoods of the current solution. If the new solution is better than the current one, it replaces the current solution and the search process resumes from this new current solution. There is a small probability that a worse solution is accepted as the new current solution.

The proposed MSA algorithm for LRPSPD combines the advantages of the simulated annealing algorithm with the multi-start hill climbing strategy. The following subsections discuss the components of the proposed MSA algorithm, including solution representation scheme, generation of initial solution, neighborhood structure, parameter setting, and the complete MSA procedure.

Solution representation and initial solution

The solution representation for LRP proposed by Yu et al. [15] is used for this study. It may be described as follows. An LRPSPD solution is encoded as an integer string consisting of a permutation of n customers $\{1, 2, \ldots, n\}$, m potential depots $\{n+1, n+2, \ldots, n+m\}$, and N_{dummy} zeros which are employed to terminate routes, in addition to the vehicle capacity constraints. The ith number in $\{1, 2, \ldots, n\}$ represents the ith customer to be serviced. The first number in a solution must be in $\{n+1, n+2, \ldots, n+m\}$ as it represents the first depot under consideration. The parameter N_{dummy} is calculated as max $\{\lceil\sum_i d_i/Q\rceil, \lceil\sum_i p_i/Q\rceil\}$, where d_i is the delivery demand of customer i, p_i is the pickup demand of customer i, Q denotes vehicle capacity, and $\lceil\bullet\rceil$ represents the smallest integer larger than or equal to the enclosed number.

The solution representation is discussed in greater detail as follows. Each depot services customers between the depot and the next depot in the solution representation. The first route of this depot starts by servicing the first customer after the depot. Subsequently, other customers serviced by this depot are added to the current route one by one. If servicing a customer exceeds the capacity of vehicle, the current route is terminated. A route can also be terminated when the next number in the solution representation is a dummy zero. Whenever a route is terminated, a new route is initiated to service remaining customers for the current depot.

The proposed solution representation scheme always gives an LRPSPD solution that satisfies vehicle capacity constraint. However, depot capacity constraint may be violated. Since the solution representation already determines open depots and vehicle routes, objective function value of the solution can be easily calculated. Because depot capacity is not explicitly considered during the decoding process, a per unit penalty cost P_{cost} is added to the objective function value if depot capacity is violated. The initial solutions are generated randomly.

Neighborhood

Let S denote the set of feasible solutions and let σ_k ($k = 1, \ldots, P_{\text{size}}$) represent the current solution, where $\sigma_k \in S$, and P_{size} denote the number of starting points in the MSA algorithm. The sets $N(\sigma_k)$ ($k = 1, \ldots, P_{\text{size}}$) are then the sets of neighboring solutions of σ_k. $N(\sigma_k)$ can be generated through either swap, insertion, or 2-opt operations.

The swap operation randomly selects two elements of the current solution and then exchanges their positions. The insertion operation randomly selects an element of the current solution and then inserts it before another randomly selected element of the current solution.

The 2-opt operation is used extensively in solving vehicle routing problem and its variants. We modify traditional 2-opt operation to improve existing routes by randomly selecting two customers serviced by the same depot, and then reversing the substring in the solution representation between them. The results are slightly different from those obtained by traditional 2-opt operation because of our special solution representation scheme. If the two selected customers are on the same route, our 2-opt operation is essentially the same as traditional inner-route 2-opt operation. However, the effect of our 2-opt operation becomes more complicated when the selected customers belong to different routes. If the routes of these two customers are adjacent, and both routes are terminated by a dummy zero, rather than vehicle capacity constraint, then the operation is similar to traditional inter-route 2-opt operation. Otherwise, all routes between the two selected customers may also be affected because of vehicle capacity constraint, and it may be necessary to rearrange these routes. Nevertheless, this modified 2-opt operation gives a neighborhood that is a superset of traditional 2-opt neighborhood and facilitates a more diverse search process.

The probability of performing each of the three neighborhood operations is fixed at 1/3.

Parameters

The proposed MSA algorithm uses seven parameters, namely I_{iter}, T_0, α, P_{cost} $N_{\text{non-improving}}$, P_{size} and $MaxT$. I_{iter} is the number of iterations performed at a particular temperature. T_0 represents the initial temperature. α is the controlling coefficient of cooling schedule. P_{cost} is the unit penalty cost incurred when depot capacity is violated, and $N_{\text{non-improving}}$ is the allowable number of consecutive temperature decreases during which σ_{best} has not been improved. $MaxT$ is the maximum computational time. Finally, P_{size} defines the number of current solutions at each iteration.

MSA procedure

Fig. 1 illustrates the proposed MSA algorithm. The current temperature T is set to T_0 initially. Then the initial solutions σ_k ($k = 1, \ldots, P_{\text{size}}$) are generated at random and used as multi-start points. At each iteration, new solutions σ'_k ($k = 1, \ldots, P_{\text{size}}$) are selected from their corresponding $N(\sigma_k)$. Furthermore, let $obj(\sigma_k, P_{\text{cost}})$ denote the objective funciton value of σ_k, and let Δ_k be the difference between the objective function values of σ_k and σ'_k, that is $\Delta_k = obj(\sigma_k, P_{\text{cost}}) - obj(\sigma'_k, P_{\text{cost}})$. If $\Delta_k < 0$, the probability of replacing σ_k with σ'_k is $Exp(\Delta_k/T)$. This is done by randomly generating a number $r \in [0, 1]$ and replacing σ_k with σ'_k when $r < Exp(\Delta_k/T)$. Meanwhile, if $\Delta_k \geq 0$, the algorithm replaces σ_k with σ'_k with probability 1.

After running I_{iter} iterations following the previous temperature decrease, T is decreased using the formula $T \leftarrow \alpha T$, where $0 < \alpha < 1$. Then, a local search procedure is applied to improve the best current solution. The local search procedure sequentially performs swap and insertion operations. The algorithm is terminated when the incumbent best solution, σ_{best}, has not been improved in $N_{\text{non-improving}}$ consecutive temperature reductions. The algorithm is also terminated when the computational time used reaches $MaxT$. Whenever the algorithm finds a new feasible σ_{best}, all the current solutions are set to be σ_{best} and the MSA is continued. After the MSA algorithm is terminated, a (near) global optimal solution can be derived from σ_{best}.

Experimental results

This section describes computational experiments used to evaluate the effectiveness and efficiency of the proposed MSA algorithm for solving LRPSPD. The test problems, parameter setting, and experimental results of the proposed MSA algorithm compared with traditional single-start simulated annealing and state-of-the-art algorithms are discussed in subsequent sections.

Test problems

To assess the proposed MSA algorithm, computational simulations and comparisons were performed based on 360 test problems of Karaoglan et al. [12]. The 360 test problems are derived from the LRP benchmark instances generated by Prodhon [44] using two demand separation approaches proposed in the literature. The number of customers ranges from 20 to 200, while the number of potential depots is either 5 or 10. Each demand follows a uniform distribution in [11,20] and depot capacities are determined under the assumption that at least two or three depots are opened. The vehicle capacity has two levels (70 and 150) and the number of clusters has three levels. Euclidean distances between customers and depots were rounded to real numbers with four digits.

Parameter selection

Comprehensive computational experiments were conducted to determine the appropriate parameter values. The

MSA (I_{iter}, T_0, α, P_{cost}, $N_{non\text{-}improving}$, P_{size}, $MaxT$)
Step 1: Generate initial solutions σ_k $k=1, 2, \ldots, P_{size}$;
Step 2: Let $T=T_0$; R=0; N=0; σ_{best} = the best σ_k among the P_{size} solutions;
    $F_{best}=obj(\sigma_{best},\ P_{\text{cost}})$;
Step 3: N=N+1;
Step 4: For $k=1$ to P_{size} {
    Step 4.1 Generate a solution σ'_k based on σ_k;
    Step 4.2 If $\Delta_i = obj(\sigma'_k,\ P_{\text{cost}}) - obj(\sigma_k,\ P_{\text{cost}}) \le 0$ {Let $\sigma_k = \sigma'_k$;}
        Else {
            Generate r~U(0,1);
            If $r < Exp(-\Delta_k / T)$ { Let $\sigma_k = \sigma'_k$;}
            Else {Discard σ'_k;}
        }
    Step 4.3 If ($obj(\sigma_k,\ P_{\text{cost}}) < F_{best}$ and σ_k is feasible) {
            $\sigma_{best} = \sigma_k$; $F_{best} = obj(\sigma_k, P_{\text{cost}})$; R=0;
            Let $\sigma_j = \sigma_{best}$ ($j=1, \ldots, P_{size}$)
        }
}
Step 5: If $N = I_{iter}$ {
    $T = T\times\alpha$; $N=0$; R=R+1;
    For $k=1$ to P_{size} {
        Perform local search on σ_k;
        If ($obj(\sigma_k) < F_{best}$ and σ_k is feasible) {
            $\sigma_{best} = \sigma_k$; $F_{best} = obj(\sigma_k)$; R=0;
            Let $\sigma_j = \sigma_{best}$ ($j=1, \ldots, P_{size}$)
        }
    }
}
Else {Go to Step 3;}
Step 6: If termination condition is satisfied {Terminate the MSA procedure;}
    Else {Go to Step 3;}

Fig. 1. Pseudo-code of the proposed MSA algorithm.

following combinations of parameter values were evaluated. $I_{\text{iter}} = (n+m-1)\times\eta/P_{\text{size}}$, $N_{\text{non-improving}} = 10$, 20, 30; $T_0 = 3$, 5, 7, 9; $P_{\text{cost}} = 0$, 200, 400; and $\alpha = 0.96$, 0.97, 0.98, 0.99, $MaxT = (n+m-1)\times 2.0$ s, where n denotes the number of locations; m represents the number of potential deposits; $\eta = 2500$, 5000, 7500; P_{size} ranges from 1 to 6. Since the number of iterations (I_{iter}) is inversely proportional to the value of P_{size}, the total number of solutions evaluated for the same problem are almost the same for different P_{size} values, ensuring a fair comparison.

The objective function value for each parameter combination is recorded to compute the *RER* (relative error rate). The *RER* is calculated as $(OBJ^P - BKS)/BKS \times 100$, where OBJ^P denotes the objective function value for the parameter combination P and BKS is the current best known solution value reported in the literature [12]. It can be seen that when I_{iter} or $N_{\text{non-improving}}$ is increased, better solutions are obtained at the expense of longer computational time. By increasing P_{size} value, *RER* values can be reduced. However, when P_{size} is larger than a threshold value, *RER* values will increase. This may be due to the fact that the number of iterations (I_{iter}) is inversely proportional to the value of P_{size}. Thus when P_{size} is too large, the number of iterations is not enough to search enough solutions at a specific temperature.

Temporarily accepting an infeasible solution in the search procedure may help escape from local optima. If P_{cost} is too high, an infeasible solution has a lower chance of being temporarily accepted. Contrarily, when P_{cost} is too low, the algorithm may accept too many infeasible solutions and affect the quality of the final solution.

It is difficult to assess the effect of T_0 and α separately. In general, T_0 affects the probability of accepting a worse solution. The higher the value of T_0, the larger the probability of accepting a worse solution will be. As a result, the convergence of the algorithm is slower. On the other hand, if the value of T_0 is too small, accepting probability of worse solutions will be small and the algorithm is more likely to be stuck at a local optimum. However, parameter combinations significantly impact solution quality. For example, setting $P_{\text{size}} = 4$,

Table 1
Performance comparison on *RPD* (%).

		SA_1			SA_2			SA			MSA		
n	*m*	*Min. RPD*	*Ave. RPD*	*Max. RPD*	*Min. RPD*	*Ave. RPD*	*Max. RPD*	*Min. RPD*	*Ave. RPD*	*Max. RPD*	*Min. RPD*	*Ave. RPD*	*Max. RPD*
10	3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.004	0.007	0.000	0.001	0.003
15	4	0.000	0.001	0.002	0.000	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000
20	5	0.000	0.001	0.002	0.000	0.001	0.005	0.006	0.027	0.051	0.005	0.017	0.033
25	5	0.000	0.106	0.177	0.002	0.109	0.182	−0.003	−0.003	−0.002	−0.003	−0.003	−0.001
30	5	0.002	0.105	0.200	0.041	0.142	0.206	−0.228	−0.226	−0.221	−0.228	−0.226	−0.221
50	5	0.012	0.473	0.674	0.187	0.510	0.681	−0.115	−0.073	0.047	−0.207	−0.125	−0.087
60	5	0.001	0.027	0.050	0.002	0.021	0.050	0.004	0.016	0.032	−0.039	0.011	0.064
60	10	0.170	1.452	3.635	0.040	0.266	0.838	−0.323	−0.247	−0.201	−0.254	−0.226	−0.201
70	5	0.073	0.132	0.188	0.017	0.084	0.109	−0.037	−0.026	−0.011	−0.037	−0.028	−0.017
70	10	0.080	0.375	0.845	0.036	0.203	0.371	−0.130	0.080	0.303	−0.099	0.075	0.254
80	5	0.042	0.106	0.172	0.006	0.106	0.349	−0.010	0.001	0.015	−0.046	−0.005	0.014
80	10	0.060	0.559	1.170	0.040	0.216	0.465	−0.175	−0.085	0.036	−0.181	−0.133	−0.089
90	5	0.000	0.007	0.012	0.000	0.015	0.033	−0.031	−0.023	−0.010	−0.034	−0.019	−0.013
90	10	0.495	1.054	1.917	0.011	0.507	1.282	−0.235	−0.158	−0.059	−0.216	−0.160	−0.100
100	5	0.053	0.155	0.251	0.036	0.138	0.234	−0.166	−0.112	−0.026	−0.169	−0.105	−0.035
100	10	0.100	1.414	3.649	0.046	0.508	1.584	−0.222	−0.151	−0.052	−0.229	−0.159	−0.046
Avg.													

$P_{cost} = 200$, $N_{non\text{-}improving} = 20$, $\eta = 5000$, difference in the objective function values obtained by using various combinations of T_0 and α can be as large as 1.2% higher than the best-known solution value for the Prod_90_5_W_coord100-5-3b.txt instance.

The results indicated that setting $T_0 = 5$, $I_{iter} = (n + m - 1) \times 5000/P_{size}$, $N_{non\text{-}improving} = 20$, $\alpha = 0.96$, $P_{cost} = 200$, $P_{size} = 3$ and $MaxT = (n + m - 1) \times 2.0$ gives the best result in terms of solution quality. This setting was used in the subsequent numerical experiments.

Results and discussion

The proposed MSA algorithm was coded in C programming language and experiments were conducted on a PC with an Intel Core 2 2.67 GHz CPU and 4 GB of RAM. Each problem instance was tested over five trials and compared with existing algorithms and the traditional single-start simulated annealing. To the best of our knowledge, the best-performing algorithms for LRPSPD are the two simulated annealing algorithms proposed by Karaoglan et al. [12]. Two initialization heuristics, Extended Clarke and Wright Algorithm (ECWA) and Facility Location Problem Heuristic (FLPH) are used in two versions of their SA heuristic, denoted by SA_1 and SA_2 in this study, respectively. If $P_{size} = 1$, the proposed MSA algorithm is the same as traditional single-start simulated annealing algorithm, denoted by SA in this study.

The main differences between the proposed MSA algorithm and the simulated annealing approaches proposed by Karaoglan et al. [12] are (1) multi-start strategy is adopted in MSA; (2) the neighborhood solution is different; (3) local search procedure is performed after each temperature reduction in MSA; and (4) the initial solution is generated differently.

Each problem is solved by SA_1, SA_2, SA and MSA for five runs. The best, average, and worst solutions of each problem obtained by each approach are recorded. The computational result of SA_1 and SA_2 were reported in Karaoglan et al. [12] and can be downloaded from http://w3.gazi.edu.tr/_fulyaal/TestCases/LRPSPD_MIPFormulation_and_SA_Solutions.xls.

The minimum, average, and maximum relative percentage deviation for the best-known solution for each problem is calculated as $RPD = (Obj_i - BKS_i)/BKS_i \times 100\%$, where Obj_i is the objective function value of instance i obtained by the algorithm being evaluated, and BKS_i is the best objective function value of instance i obtained by SA_1 and SA_2. As can be seen in Table 1, the *Min. RPD*, *Ave. RPD* and *Max. RPD* for the proposed MSA algorithm are −0.116%, −0.064%, and −0.001%, respectively. Comparison with the other three heuristics shows that the proposed MSA algorithm outperforms all the other approaches in solving LRPSPD.

Table 2
Number of new best solutions obtained by each approach.

n	*m*	# of instances	SA_1	SA_2	SA	MSA
10	3	16	16	16	16	16
15	4	16	16	16	16	16
20	5	16	16	16	15	14
25	5	24	21	20	24	24
30	5	24	16	16	24	24
50	5	24	7	11	17	17
60	5	24	9	12	10	11
60	10	24	3	4	16	12
70	5	24	3	3	13	11
70	10	24	4	8	8	7
80	5	24	2	2	8	12
80	10	24	3	2	8	13
90	5	24	4	5	4	11
90	10	24	0	0	13	14
100	5	24	4	5	4	11
100	10	24	0	3	9	12
Total						

Table 2 shows the number of new best-known solutions obtained by each approach. SA_1, SA_2, SA, and MSA obtained new best solutions for 124 (124/360 = 34.44%), 139 (139/360 = 38.61%), 205 (205/360 = 56.94%) and 225 (225/360 = 62.50%) instances, respectively. The proposed MSA algorithm acquired the most of new best solutions among the four approaches.

Table 3 lists the average computation time (CPU time in seconds) for each algorithm. SA_1 and SA_2 were implemented using C++ programming language and performed on a computer equipped with Intel Xeon 3.16 GHz and 1 GB of RAM. Because computational time may be affected by hardware, software and programming skills, comparison of computational efficiency of different approaches is not the focus of this study. Nonetheless, the proposed MSA algorithm is highly effective in terms of solution quality.

For a more rigorous performance analysis, a group of one-sided paired *t*-tests with respect to the *Min. RPD*, *Ave. RPD*, and *Max. RPD* for each instance was performed to compare the proposed MSA algorithm with SA_1, SA_2, and SA. At confidence level $\alpha = 0.05$, results in Table 4 showed that the proposed MSA algorithm significantly outperformed SA_1 and SA_2 in terms of *Min. RPD*. Further, it significantly outperformed SA_1, SA_2, and traditional single-start simulated annealing heuristic on *Ave. RPD* and *Max. RPD*. These statistical results also showed that the multi-start hill climbing

Table 3
Computational time required by each approach (seconds).

n	m	# of instances	SA_1	SA_2	SA	MSA
10	3	16	20.2	19.9	1.5	1.5
15	4	16	45.3	45.6	2.6	2.6
20	5	16	54.1	54.3	5.1	4.8
25	5	24	65.7	65.3	9.1	9.1
30	5	24	71.1	71.6	15.3	14.6
50	5	24	110.7	111.8	47.5	49.5
60	5	24	131.3	131.7	77.4	80.7
60	10	24	113.0	112.2	86.1	90.2
70	5	24	153.4	153.7	111.7	117.3
70	10	24	117.6	120.8	100.2	106.2
80	5	24	170.3	171.6	149.8	155.9
80	10	24	141.2	145.2	148.7	158.6
90	5	24	129.3	132.0	209.9	205.4
90	10	24	162.4	169.5	194.1	201.3
100	5	24	190.3	192.3	211.5	217.2
100	10	24	371.2	381.8	220.1	232.8
Avg.						

Table 4
Analytical results of the paired t-tests with respect to *RPD* for each benchmark instance.

MSA V.S.	SA_1	SA_2	SA
Test on *Min. RPD*			
Paired difference	−0.19773	−0.15632	−0.00768
t-value	−6.47020	−5.81700	−0.84471
Degree of freedom	359	359	359
P-value	<0.00001	<0.00001	0.199418
Test on *Ave. RPD*			
Paired difference	−0.47860	−0.26763	−0.00922
t-value	−7.73147	−6.98207	−1.86402
Degree of freedom	359	359	359
P-value	<0.00001	<0.00001	0.031567
Test on *Max. RPD*			
Paired difference	−0.90847	−0.46529	−0.02294
t-value	−7.39773	−6.62658	−1.99885
Degree of freedom	359	359	359
P-value	<0.00001	<0.00001	0.023189

strategy significantly improved the performance of the single-start simulated annealing algorithm.

Conclusions and future research

LRPSPD simultaneously considers two critical issues in the design of distribution networks: determining locations of depots and planning vehicle routes. Moreover, it also takes into consideration both the forward and reverse directions of flows in the logistics network. As the reverse logistics is gaining more and more attention from the research community and the industry, LRPSPD is likely to attract much interest from academics and practitioners. To solve the LRPSPD, this study proposes an MSA algorithm which uses a special solution representation and a multi-start mechanism. The computational results indicate that the proposed MSA algorithm is effective and efficient in solving LRPSPD. It outperforms the single-start simulated annealing algorithm and other state-of-the-art approaches proposed in the literature.

Future research may focus on developing different heuristics for solving LRPSPD and check the performance of the heuristics on the datasets generated by Karaoglan et al. [12]. Researchers may also consider LRPSPD with more practical constraints such as a time windows to bring the problem closer to reality.

Acknowledgements

This research was partially supported by the Ministry of Science and Technology of the Republic of China (Taiwan) under grants NSC 100-2410-H-011-005-MY2 and NSC 101-2410-H-182-004-MY2. This support is gratefully acknowledged.

References

[1] S. Salhi, G. Rand, The effect of ignoring routes when locating depots, Eur. J. Oper. Res. 39 (1989) 150–156.
[2] S.K. Jacobsen, O.B.G. Madsen, A comparative study of heuristics for a two-level routing-location problem, Eur. J. Oper. Res. 5 (1980) 378–387.
[3] O.B.G. Madsen, Methods for solving combined two level location-routing problems of realistic dimensions, Eur. J. Oper. Res. 12 (1983) 295–301.
[4] C. Watson-Gandy, P. Dohrn, Depot location with van salesman – a practical approach, Omega 1 (1973) 321–329.
[5] C.K.Y. Lin, C.K. Chow, A. Chen, A location-routing-loading problem for bill delivery services, Comput. Ind. Eng. 43 (2002) 5–25.
[6] K.G. Murty, P.A. Djang, The U.S. army national guard's mobile training simulators location and routing problem, Oper. Res. 47 (1999) 175–182.
[7] D. Aksen, K. Altinkemer, A location-routing problem for the conversion to the click-and-mortar retailing: the static case, Eur. J. Oper. Res. 186 (2008) 554–575.
[8] I. Karaoglan, F. Altiparmak, I. Kara, B. Dengiz, A branch and cut algorithm for the location-routing problem with simultaneous pickup and delivery, Eur. J. Oper. Res. 211 (2011) 318–332.
[9] H. Min, The multiple vehicle routing problem with simultaneous delivery and pick-up points, Transp. Res. A 23 (1989) 377–386.
[10] B. Çatay, A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery, Expert Syst. Appl. 37 (2010) 6809–6817.
[11] J. Dethloff, Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick up, OR Spectr. 23 (2001) 79–96.
[12] I. Karaoglan, F. Altiparmak, I. Kara, B. Dengiz, The location-routing problem with simultaneous pickup and delivery: formulations and a heuristic approach, Omega 40 (2012) 465–477.
[13] G. Laporte, Y. Nobert, D. Arpin, An exact algorithm for minimizing routing and operating costs in depot location, Eur. J. Oper. Res. 6 (1986) 293–310.
[14] S.W. Lin, V.F. Yu, S.Y. Chou, Solving the truck and trailer routing problem based on a simulated annealing heuristic, Comput. Oper. Res. 36 (2009) 1683–1692.
[15] V.F. Yu, S.-W. Lin, W. Lee, C.-J. Ting, A simulated annealing heuristic for the capacitated location routing problem, Comput. Ind. Eng. 58 (2010) 288–299.
[16] V.F. Yu, S.W. Lin, S.Y. Chou, The museum visitor routing problem, Appl. Math. Comput. 216 (2010) 719–729.
[17] S.W. Lin, V.F. Yu, A simulated annealing heuristic for the team orienteering problem with time windows, Eur. J. Oper. Res. 217 (2012) 94–107.
[18] S.W. Lin, V.F. Yu, S.Y. Chou, A note on the truck and trailer routing problem, Expert Syst. Appl. 37 (2010) 899–903.
[19] S.W. Lin, V.F. Yu, C.C. Lu, A simulated annealing heuristic for the truck and trailer routing problem with time windows, Expert Syst. Appl. 38 (2011) 15244–15252.
[20] H.C. Huang, J.S. Pan, Z.M. Lu, S.H. Sun, H.M. Hang, Vector quantization based on genetic simulated annealing, Signal Process. 81 (2001) 1513–1523.
[21] S.-W. Lin, K.-C. Ying, C.-C. Lu, J.N.D. Gupta, Applying multi-start simulated annealing to schedule a flowline manufacturing cell with sequence dependent family setup times, Int. J. Prod. Econ. 130 (2011) 246–254.
[22] S.-W. Lin, K.-C. Ying, Minimizing makespan and total flowtime in permutation flowshops by a bi-objective multi-start simulated-annealing algorithm, Comput. Oper. Res. 40 (2011) 1625–1647.
[23] S.-W. Lin, K.-C. Ying, Scheduling a bi-criteria flowshop manufacturing cell with sequence-dependent family setup times, Eur. J. Ind. Eng. 6 (2012) 474–496.
[24] S.-W. Lin, Solving the team orienteering problem using effective multi-start simulated annealing, Appl. Soft Comput. 13 (2013) 1064–1073.
[25] Y. Gajpal, P. Abad, An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup, Comput. Oper. Res. 36 (2009) 3215–3223.
[26] F.P. Goksal, I. Karaoglan, F. Altiparmak, A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery, Comput. Ind. Eng. 65 (2013) 39–53.
[27] N.A. Wassan, A.H. Wassan, G. Nagy, A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries, J. Comb. Optim. 15 (2007) 368–386.
[28] G. Clarke, J.W. Wright, Scheduling of vehicles from a central depot to a number of delivery points, Oper. Res. 12 (1964) 568–581.
[29] I.H. Osman, N.A. Wassan, A reactive tabu search meta-heuristic for the vehicle routing problem with back-hauls, J. Sched. 5 (2002) 263–285.
[30] D. Tuzun, L.I. Burke, A two-phase tabu search approach to the location routing problem, Eur. J. Oper. Res. 116 (1999) 87–99.
[31] T.-H. Wu, C. Low, J.-W. Bai, Heuristic solutions to multi-depot location-routing problems, Comput. Oper. Res. 29 (2002) 1393–1415.
[32] S.C. Liu, S.B. Lee, A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into consideration, Int. J. Adv. Manuf. Technol. 22 (2003) 941–950.
[33] C. Prins, C. Prodhon, R.W. Calvo, Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking, 4OR 4 (2006) 221–238.
[34] C. Duhamel, P. Lacomme, C. Prins, C. Prodhon, A GRASP × ELS approach for the capacitated location-routing problem, Comput. Oper. Res. 37 (2010) 1912–1923.

[35] S.S. Barreto, Análise e Modelização de Problemas de localização-distribuição (Analysis and Modelling of Location-routing Problems), Campus universit' ario de Santiago, University of Aveiro, Aveiro, Portugal, 2004.
[36] V.-P. Nguyen, C. Prins, C. Prodhon, A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem, Eng. Appl. Artif. Intell. 25 (2012) 56–71.
[37] H. Derbel, B. Jarboui, S. Hanafi, H. Chabchoub, Genetic algorithm with iterated local search for solving a location-routing problem, Expert Syst. Appl. 39 (2012) 2865–2871.
[38] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (1953) 1087–1092.
[39] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[40] C.-S. Jeong, M.-H. Kim, Fast parallel simulated annealing for traveling salesman problem on SIMD machines with linear interconnections, Parallel Comput. 12 (1991) 221–228.
[41] S. Sofianopoulou, Simulated annealing applied to the process allocation problem, Eur. J. Oper. Res. 60 (1992) 327–334.
[42] I. Osman, C. Potts, Simulated annealing for the permutation flowshop problem, Omega 17 (2003) 551–557.
[43] S.U. Seçkiner, M. Kurt, A simulated annealing approach to the solution of job rotation scheduling problems, Appl. Math. Comput. 188 (2007) 31–45.
[44] C. Prodhon, Classical Iinstances for LRP., 2008, http://prodhonc.free.fr/homepage (accessed 10.02.13).