IEMS Vol. 10, No. 1, pp. 24-33, March 2011.

A Hybrid Genetic Algorithm for the Location-Routing Problem with Simultaneous Pickup and Delivery

Ismail Karaoglan
Department of Industrial Engineering
Selcuk University, Konya, 42075, Turkey
E-mail: ikaraoglan@selcuk.edu.tr

Fulya Altiparmak†
Department of Industrial Engineering
Gazi University, Ankara, 06570, Turkey
Tel: +90-312-582-3856, E-mail: fulyaal@gazi.edu.tr

Received, January 22, 2011; Revised, February 15, 2011; Accepted, February 21, 2011

Abstract. In this paper, we consider the Location-Routing Problem with simultaneous pickup and delivery (LRPSPD) which is a general case of the location-routing problem. The LRPSPD is defined as finding locations of the depots and designing vehicle routes in such a way that pickup and delivery demands of each customer must be performed with same vehicle and the overall cost is minimized. Since the LRPSPD is an NP-hard problem, we propose a hybrid heuristic approach based on genetic algorithms (GA) and simulated annealing (SA) to solve the problem. To evaluate the performance of the proposed approach, we conduct an experimental study and compare its results with those obtained by a branch-and-cut algorithm on a set of instances derived from the literature. Computational results indicate that the proposed hybrid algorithm is able to find optimal or very good quality solutions in a reasonable computation time.

Keywords: Location-routing Problem, Simultaneous Pickup and Delivery, Genetic Algorithms, Simulated Annealing

1. INTRODUCTION

The distribution network design (DND) is one of the most important problems in supply chain and logistics management. Distribution costs often represent an important portion of overall system cost and substantial savings can be achieved by improving distribution system. Among the others, location of facilities (i.e. factory, depot, warehouse etc.) can be thought of a crucial component in DND because of the large setup and operating costs. In general, the classical facility location problem requires that the customers must be served directly. This situation is true if the demand of customers are equal to the vehicle capacity. However, in many applications in practice, demand of customers are less than the vehicle capacity and deliveries are made on a route in which two or more customers are visited sequentially. Therefore, locating the facilities without considering vehicle routes may lead to suboptimal solutions (Salhi and Rand, 1989). The Location Routing Problem (LRP) overcomes this drawback by considering vehicle routes while locating the facilities.

In the general form, the LRP deals with determining the location of facilities and the routes of the vehicles for serving the customers under some constraints such as facility and vehicle capacity, route lengths, etc. to satisfy demands of all customers and to minimize total cost including routing costs, vehicle fixed costs, facility fixed costs and facility operating costs.

The facility location problem (FLP) and vehicle routing problem (VRP) are two main components of the LRP. Since both problems belong to the class of NP-hard problems, the LRP is also NP-hard problem. Several mathematical models and exact solution procedures have been developed for small-and medium-size LRPs in the literature (Laporte *et al*., 1983, Laporte *et al*., 1986, Belenguer *et al*., 2006, Berger *et al*., 2007). Different heuristic approaches have been also proposed in the literature to solve larger LRPs. Perl and Daskin (1984, 1985), Srivastava and Benton (1990), Srivastava (1993) and Hansen *et al*. (1994) use classic heuristic approaches for the problem. Moreover, meta-heuristic approa-

† : Corresponding Author

ches have been successfully implemented for the problem. Several examples for the application of meta-heuristic approaches can be given as: tabu search (Tuzun and Burke, 1999; Albareda-Sambola *et al*., 2005), simulated annealing (Wu *et al*., 2002; Yu *et al*., 2010), greedy randomized adaptive search procedure (GRASP) (Prins *et al*., 2006a; Duhamel *et al*., 2010), memetic algorithms (Prins *et al*., 2006b), variable neighborhood search algorithms, (Melechovsky *et al*., 2005) and particle swarm optimization (Marinakis and Marinak, 2008). Comprehensive reviews of location-routing models and their applications are provided in Laporte (1988), Min *et al*. (1998) and Nagy and Salhi (2007).

In the literature, it is seen that the researchers have considered classical VRP, i.e. each vehicle starts from a depot, traverses through a number of customers, delivers goods to each customer and returns the same depot, within the LRP. However, in practice, customers can have pickup and delivery demands and they request that both demands should be met at the same time. This kind of problem is known in the literature as vehicle routing problem with simultaneous pickup and delivery, VRPSPD (Berbeglia *et al*., 2007; Parragh *et al*., 2008).

In this paper, we consider a variant of the LRP called LRP with simultaneous pickup and delivery (LRP SPD). The LRPSPD is an extension of LRP in terms of types of the customers' demand. The LRPSPD can be also considered as an extension of travelling salesman location problem with pickup and delivery (TSPPD) introduced by Mosheiov (1994) in terms of the number of depots to be located and the capacity of vehicles. Finally, the LRPSPD can be considered as a special case of many to many LRP introduced by Nagy and Salhi (1998) in which several customers wish to send goods to others. The LRPSPD arises in a number of reverse logistics context. For example, perishable product distribution systems (i.e. milk, fresh foods, newspaper etc.) are the most important application areas of the LRPSPD. In these systems, firms are responsible for not only distributing the fresh foods or daily newspapers but also collecting the outdated foods or newspapers for destroying or reusing. The beverage and automotive industries and also grocery store chains are given other examples for the application of the LRPSPD.

Although the LRP has been studied extensively in the literature, the LRPSPD has received no attention from researchers so far. To the best of our knowledge, we are first to address the LRPSPD. In our previous study, we have proposed two MIP formulations, which are two-index node-based and flow-based formulations, for the problem and presented several polynomial-size valid inequalities adapted from literature to strengthen the formulations (Karaoglan *et al*., 2009). Moreover, we have developed branch and cut (B&C) algorithm, which is an exact solution procedure, based on flow-based formulation. Our experimental studies have revealed that our B&C algorithm solves small and some medium-size LRPSPD instances to optimality within four hours of computation time (Karaoglan *et al*., 2011).

In this paper, we propose a hybrid heuristic approach based on genetic algorithms (GA) and simulated annealing (SA), called hGA, to solve medium and large-size LRPSPDs where SA is used as a local search algorithm after each recombination procedure of GA. We investigate the performance of the hGA on a set of instances derived from the literature and compare it with the B&C algorithm with respect to solution quality and computation time. Computational results indicate that the proposed hGA is able to find optimal or very good quality solutions in a reasonable computation time.

The paper is organized as follows: Problem definition and mathematical formulation are given in Section 2. The detailed description of proposed hybrid algorithm is given in Section 3. Section 4 reports computational results and conclusions follow in Section 5.

2. PROBLEM DEFINITION AND MATHEMATICAL MODEL

The location-routing problem with pickup and delivery (LRPSPD) can be defined as follows: let $G = (N, A)$ be a complete directed network where $N = N_0 \cup N_C$ is a set of nodes in which N_0 and N_C represent the potential depot nodes and customers, respectively, and $A = \{(i, j): i, j \in N\}$ is the set of arcs. Each arc $(i, j) \in N$ has a non-negative cost (distance) c_{ij} and triangular inequality holds (i.e., $c_{ij}+c_{jk} \geq c_{ik}$). A capacity CD_k and a fixed cost FD_k are associated with each potential depot $k \in N_0$. An unlimited fleet of homogeneous vehicles with capacity CV and fixed operating cost FV including the cost of acquiring the vehicles used in the routing is available to serve the customers. Each customer $i \in N_C$ has pickup (p_i) and delivery (d_i) demands, with $0 < d_i, p_i \leq CV$. The problem is to determine the locations of depots, the assignment of customers to opened depots and the corresponding vehicle routes with minimum total cost under following constraints:

- Each vehicle is used at most one route,
- Each customer is served by exactly one vehicle,
- Each route begins and ends at the same depot,
- The total vehicle load at any point of the route does not exceed the vehicle capacity,
- The total pickup and total delivery load of the customers assigned to a depot does not exceed the capacity of the depot.

To formulate the LRPSPD, following decision variables are used:

$$x_{ij} = \begin{cases} 1 & \textit{if a vehicle travels directly from} \\ & \textit{node } i \textit{ to node } j \ (\forall i, j \in N) \\ 0 & \textit{otherwise} \end{cases}$$

$$y_k = \begin{cases} 1 & \textit{if depot } k \textit{ is opened } (\forall k \in N_0) \\ 0 & \textit{otherwise} \end{cases}$$

$$z_{ik} = \begin{cases} 1 & \textit{if customer } i \textit{ is assigned to} \\ & \textit{depot } k \ (\forall i \in N_C, \forall k \in N_0) \\ 0 & \textit{otherwise} \end{cases}$$

U_{ij}; demand to be delivered to customers routed after node i and transported in arc (i, j) if a vehicle travels directly from node i to node j $(\forall i, j \in N)$, otherwise 0

V_{ij}; demand to be picked-up from customers routed up to node i (including node i) and transported in arc (i, j) if a vehicle travels directly from node i to node j $(\forall i, j \in N)$, otherwise 0

The two-index flow-based formulation of the LRP SPD is given as follows (Karaoglan *et al*., 2009, 2011):

$$\text{Min} \sum_{i \in N}\sum_{j \in N} c_{ij}x_{ij} + \sum_{k \in N_0} FD_k y_k + \sum_{k \in N_0}\sum_{i \in N_C} FVx_{ki} \quad (1)$$

$$\text{s.t.} \sum_{j \in N} x_{ij} = 1 \qquad \forall i \in N_C \quad (2)$$

$$\sum_{j \in N} x_{ji} = \sum_{j \in N} x_{ij} \qquad \forall i \in N \quad (3)$$

$$\sum_{j \in N} U_{ji} - \sum_{j \in N} U_{ij} = d_i \qquad \forall i \in N_C \quad (4)$$

$$\sum_{j \in N} V_{ij} - \sum_{j \in N} V_{ji} = p_i \qquad \forall i \in N_C \quad (5)$$

$$U_{ij} + V_{ij} \le CVx_{ij} \qquad \forall i, j \in N,\ i \ne j \quad (6)$$

$$\sum_{j \in N_C} U_{kj} = \sum_{j \in N_C} z_{jk} d_j \qquad \forall k \in N_0 \quad (7)$$

$$\sum_{j \in N_C} U_{jk} = 0 \qquad \forall k \in N_0 \quad (8)$$

$$\sum_{j \in N_C} V_{jk} = \sum_{j \in N_C} z_{jk} p_j \qquad \forall k \in N_0 \quad (9)$$

$$\sum_{j \in N_C} V_{kj} = 0 \qquad \forall k \in N_0 \quad (10)$$

$$U_{ij} \le (CV - d_i)x_{ij} \qquad \forall i \in N_C, \forall j \in N \quad (11)$$

$$V_{ij} \le (CV - p_j)x_{ij} \qquad \forall i \in N, \forall j \in N_C \quad (12)$$

$$U_{ij} \ge d_j x_{ij} \qquad \forall i \in N, \forall j \in N_C \quad (13)$$

$$V_{ij} \ge p_i x_{ij} \qquad \forall i \in N_C, \forall j \in N \quad (14)$$

$$\sum_{k \in N_0} z_{ik} = 1 \qquad \forall i \in N_C \quad (15)$$

$$\sum_{i \in N_C} d_i z_{ik} \le CD_k y_k \qquad \forall k \in N_0 \quad (16)$$

$$\sum_{i \in N_C} p_i z_{ik} \le CD_k y_k \qquad \forall k \in N_0 \quad (17)$$

$$x_{ik} \le z_{ik} \qquad \forall i \in N_C, \forall k \in N_0 \quad (18)$$

$$x_{ki} \le z_{ik} \qquad \forall i \in N_C, \forall k \in N_0 \quad (19)$$

$$x_{ij} + z_{ik} + \sum_{m \in N_0, m \ne k} z_{jm} \le 2 \qquad \forall i, j \in N_C,\ i \ne j, \forall k \in N_0 \quad (20)$$

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

$$z_{ik} \in \{0, 1\} \qquad \forall i \in N_C, \forall k \in N_0 \quad (22)$$

$$y_k \in \{0, 1\} \qquad \forall k \in N_0 \quad (23)$$

$$U_{ij}, V_{ij} \ge 0 \qquad \forall i, j \in N \quad (24)$$

where x_{ij} is set to zero when $\max\{d_i + d_j;\ p_i + p_j;\ d_j + p_i\} > CV,\ \forall i, j \in N_C,\ i \ne j$. This restriction guaranties that any incompatible customer pair whose total demands are greater than the vehicle capacity is not appeared in the same route. In this formulation, objective function (1) minimizes the total system cost including transportation, depot and vehicle fixed costs. Constraints (2) and (3) are known as degree constraints. While constraints (2) ensure that each customer must be visited exactly once, constraints (3) guarantee that entering and leaving arcs to each node are equal. Constraints (4) and (5) are flow conservation constraints for delivery and pickup demands, respectively. These constraints eliminate subtour and guarantee that pickup and delivery demands are satisfied for each customer. Constraints (6) imply that total load on any arc must not exceed the vehicle capacity. While constraints (7) ensure that total delivery load dispatching from each depot equals to total delivery demand of customers which are assigned to the corresponding depot, constraints (8) guarantee that the total amount of delivery load returning to the depots must be equal to zero. Similarly, while constraints (9) ensure that total pickup load entering to each depot equals to total pickup demand of customers which are assigned to the corresponding depot, constraints (10) guarantee that the total amount of pickup load dispatching from the depots must be equal to zero. Constraints (11)~(14) are bounding constraints for additional variables. Constraints (15) ensure that each customer must be assigned to only one depot. Constraints (16) and (17) guarantee that total delivery and pickup loads on any depot must not exceed the corresponding depot capacity, respectively. Constraints (18)~(20) forbid the illegal routes such that beginning and ending depots are not same. Finally, constraints (21)~(24) are known as integrality constraints which define the nature of the decision variables. This formulation includes $O\left((|N_c| + |N_0|)^2\right)$ binary variables, $O\left((|N_c| + |N_0|)^2\right)$ additional variables and $O\left(|N_c|^2 |N_0|\right)$ constraints.

3. PROPOSED HYRBID GENETIC ALGORITHM

Genetic Algorithms (GA) refer to a class of adaptive search procedures based on the principles derived from natural evolution and genetics. GA offer significant advantages over conventional methods by using simultaneously several search principles and heuristics. The most important ones include a population-wide search, a continuous balance between convergence and diversity, and the principles of building-block combination. GA can be implemented in several different ways to solve any problem (Gen and Cheng, 1997). In our implementation, we use steady-state replacement strategy in which only one new solution is obtained by genetic operators and inserted to the current population, and then the worst chromosome is removed from population. Moreover, simulated annealing (SA) algorithm is applied after genetic operators to improve the solution. This section presents the detailed description of components of the proposed hybrid algorithm, hGA, for the LRPSPD.

3.1 Representation

Representation is one of the important issues that affect the performance of GA. Usually different problems have different data structures or genetic representations. In this paper, we implement encoding scheme originally proposed by Prins *et al*. (2006b) for the LRP. In this encoding scheme, chromosome consists of two parts: *i*) depot status and *ii*) customer sequence. Depot status is a vector of $|N_0|$ potential depots. Each gene in the vector represents the status of corresponding depot, i.e. closed or opened. If the depot i is closed, its gene takes the value of zero, otherwise it has an integer number that represents the index in customer sequence of the first customer assigned to that depot. Customer sequence is a permutation of $|N_C|$ customers in which lists of customers assigned to opened depots are connected in series. The decoding of a chromosome includes: *i*) determining the opened depots using depot status part and *ii*) assigment of customers to the opened depots considering depot and customers statuses and also depot capacities. After feasible assignment of customers to the opened depots, optimal routes for each opened depot are obtained. In this paper, route partitioning procedure proposed by Prins (2002) for the VRP is adapted to the VRPSPD. The adapted route partitioning procedure checks the route feasibility before linking two nodes in the auxiliary graph. An illustrative example for a chromosome is given in Figure 1. In this example, the LRPSPD consists of 3 potential depots and 11 customers. When the depot status part of the chromosome is considered, it is seen that depots 1 and 3 are opened while depot 2 is closed. The indexes of first customers assigned the depots 1 and 3 are 7 and 1, respectively. It means that the customers indexed between 1 and 6 in the customer sequence, i.e. (7-6-10-11-4-8), are assignned to depot 1 and the customers indexed between 7 and 11, i.e. (9-2-1-3-5), are assigned to depot 3. After these assignments, optimal routes for each opened depot are obtained by using adapted route partitioning proce-dure briefly explained above. In this example, vehicle capacity is set to 50**.**

3.2 Initialization of Population

Usually, the initial population is generated randomly. However, selection a suitable initial population accelerates the convergence of the GA. To reach optimal/near optimal solution, we generate an initial population including heuristic solutions as well as random solutions to explore the different regions of solution space. While half of the initial population consists of heuristic solutions obtained by a greedy heuristic approach called Extended Clarke and Wright Algorithm (ECWA), the rest of the initial population includes randomly generated solutions. The ECWA is an extension of the well-known Clarke and Wright heuristic (1964) and firstly introduced by Prins *et al*. (2006b) for the LRP. The implementation of this heuristic to the LRPSPD can be summarized as follows: initially, a penalty cost (pc_i) arising when the customer is assigned to the second closest depot instead of the closest one is calculated for each customer, and the customers are listed in non-in-

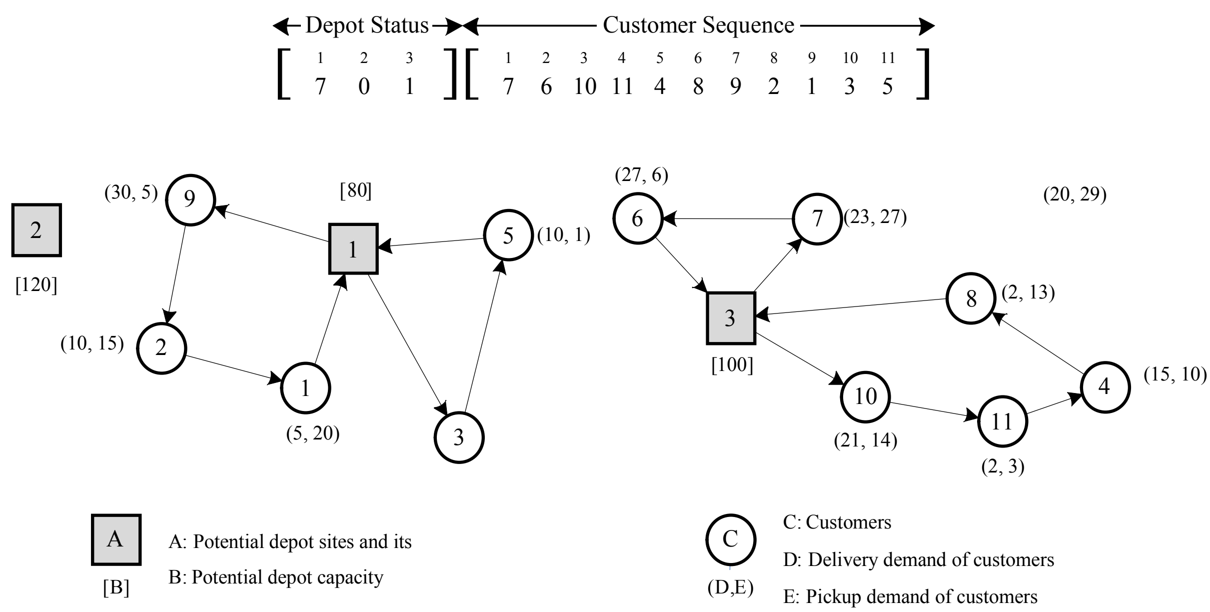

Figure 1. An example for a chromosome and LRPSDP solution.

creasing order of their penalty costs. Then, each customer is assigned to the closest depot starting from top of the list. If some of the customers are not assigned to their closest depots because of the depots' capacity, their penalty costs are recalculated considering closest ones with enough capacity and the assignment of these customers to appropriate depots is tried again. These steps, i.e. calculating penalty costs and assignment of customers to closest depots with enough capacity, are repeated until all customers are assigned. After that depots without customers are closed and a simple route for each customer from whose depot is constructed. Then, each pair of routes (R_1, R_2) in the feasible solution are examined in terms of cost saving obtained by their combination and a pair of nodes providing largest cost saving is combined (see merge operator explained in moving strategies). This strategy is repeated until no capacity-feasible combination is found. In order to generate the different good solutions for the initial population, route combination alternatives are listed in non-increasing order according to their cost savings and one of them is selected randomly from the first $|N_0|$ alternatives in this list.

3.3 Crossover and Mutation Operators

The crossover and mutation operators are used to explore new solutions in the search space. In general, crossover operator is applied to obtain new solutions by exchanging some information between selected parents. Since the chromosome in the hGA composes of two different encoding types, i.e. integer number encoding in depot status and permutation encoding in customers' sequence, we utilize two different crossover operators for the chromosomes: one-point crossover for depot status and partially-matched crossover (PMX) for customers sequence. One-point crossover is a well-known classical genetic operator in which a cut point is selected randomly between 1 and $|N_0|$-1, and then the depot status of the offspring is obtained by taking left and right parts of the cut point from the first and second parent, respectively. In PMX, two different crossing points are randomly selected. These two points enclose a matching section of one or more genes of a chromosome. This matching section of the first parent is directly transferred into the offspring without changing the index of each gene. Then, the offspring is filled up by copying the elements of the second parent. At this point, if any gene is already present in the offspring then it is replaced according to mappings. It is worthy to note that the parents undergoing crossover operation in each generation of the hGA are selected by tournament selection strategy. Figure 2 depicts the application of the proposed crossover operator to two chromosomes.

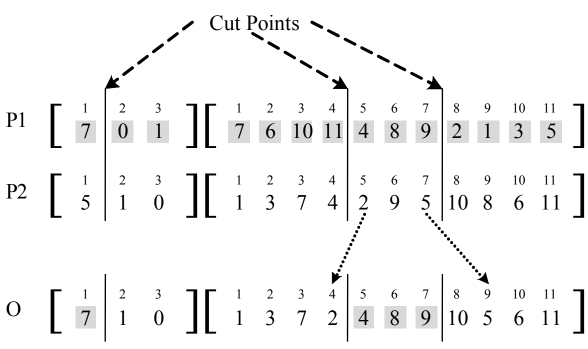

Figure 2. An example for crossover operator.

Unlike the crossover, mutation is usually done by modifying gene within a chromosome. We implement classical swap mutator for the second part of choromosome. In this operator, randomly selected two customers are exchanged. The application of the mutation operator is given in Figure 3.

Crossover and mutation operators may lead to infeasible solution. Therefore, after each recombination operation, the feasibility check should be performed. Firstly, it must be checked that all the customers are assigned to a depot (i.e. depot status includes the index 1). If it fails, then the first closed depot is opened and the depot status of this depot is changed from 0 to 1. Secondly, capacity violation of the opened depots must be checked (i.e. total delivery or total pickup demand of customers assigned to a opened depot is greater than the capacity of the depot). In this situation, the last assigned customer is removed from the depot and assigned to the first opened depot having enough remaining capacity. If none of the depots have enough capacity to assign the corresponding customer, one of the closest depots is opened randomly.

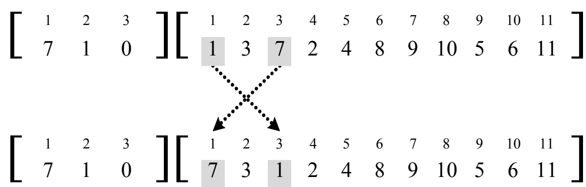

Figure 3. An example for mutation operator.

3.4 Local Search Based on Simulated Annealing

In each generation of the hGA, simulated annealing (SA) algorithm is performed as a local search algorithm to improve the offspring obtained by genetic operators. SA, which stems from the simulation of the annealing of solids, is a stochastic search technique that is able to escape local optima using a probability function (Suman and Kumar, 2006). It starts with a feasible solution and improves it until stopping criterion is met. We employ following four moving strategies, some of which are well-known in the VRP literature and also extended to the LRP by Prins *et al.* (2006b), to define neighborhoods for the SA.

• *Insert*: One customer is inserted to a new position, which is in the same route or in two different routes belonging to same depot or two different depots, from its current position.
• *Swap*: Two customers, which are in the same route or in two different routes belonging to same depot or two different depots, are exchanged.
• *Opt*: Two non-consecutive arcs, which are in the same route or in two different routes belonging to same depot or different depots, are deleted. If the deleted arcs are in the same route, two new arcs are created and the path lying between the created arc pair is reversed. If the deleted arcs are in two different routes of the same depot, each route is divided into two parts as starting and terminating part. Then two new arcs are created in such way that starting and terminating parts of two different routes are connected. Otherwise, i.e. the deleted arcs belong to two different routes of different depots, after two different parts coming from different routes are connected two new arcs, they are revised such that each route starts and finishes at the same depot.
• *Merge*: Two routes belonging to same or different depots are selected and merged into new one considering following alternatives: new route is assigned to the depot of first or second route, or to another depot opened or not. Hence, after these alternatives are evaluated in each merge operation, the one with largest cost saving is applied.

It should be noted that these moving strategies are implemented to the LRPSPD solution obtained by decoding of the chromosome. At each iteration of the SA, neighbors of the current solution (S_{cur}) are generated by using all of the moving strategies and the best one among the neighbors is chosen as a new solution (S_{new}) for the problem. The SA algorithm implements a candidate list strategy in generating the neighbors since searching whole neighborhood of the current solution by a moving strategy is a very time consuming process. According to this strategy, each moving strategy generates randomly a subset of neighbors that satisfying vehicle and depot capacity constraints and they are gathered in a pool to select a new solution. If the new solution is better than current solution then it is accepted as current solution, otherwise it is accepted with probability of $exp(-\Delta s / T_{itr})$ as a current solution. Δs is a relative percent deviation of quality of the new solution from the current solution and calculated by $[(f(S_{new}) - f(S_{cur}))/f(S_{cur})]*100$. In each iteration of SA, temperature (T_{itr}) is reduced using a geometric cooling schedule, i.e. $T_{itr} = \alpha T_{itr-1}$ where α is a cooling rate. SA stops when the temperature reaches to final temperature.

3.5 Procedure of the Proposed hGA

The overall pseudo-code procedure for solving the LRPSPD is outlined in Figure 4.

4. COMPUTATIONAL RESULTS

In order to investigate the performance of the hGA, the results of hGA are compared with the upper bounds obtained by B&C algorithm on a set of test instances. Since there are no benchmark instances in the literature for the LRPSPD, we have derived our test problems considering the LRP test set generated by Prodhon (2008) and two demand separation approaches proposed in the literature.

```
algorithm: hGA for the LRPSPD
input: LRPSPD data, GA and SA parameters
output: best solution
begin
   k ← 0;
   initialize P(k);
   evaluate P(k);
   while not (termination condition) do
     select P1 and P2 by binary tournament from P(k);
     apply crossover to P1 and P2 for obtaining C;
     apply mutation to C;
     evaluate the C;
     improve the C by SA;
     update P(k) by replacing the C with the worst solution
   end
   output best solution
end
```

Figure 3. Overall procedure of the hGA.

Prodhon's test set consists of 28 LRP instances with capacitated routes and depots, and coordinates are given for each node (potential depot sites and customers). The cost between two nodes is the Euclidean distance rounded to the real number with four digits. The main characteristics of the set are given as follows: the number of customers $|N_C|$ in {20, 50, 100, 200}, the number of potential depots $|N_0|$ in{5, 10}, uniform demands in [11, 20], the number of clusters *clu* in {1, 2, 3} (1 means that all nodes scatter on Euclidean plane), vehicle capacity *CV* in {70, 150}. Depot capacities have been determined in such a way that at least two or three depots are opened. In rest of the paper, we utilize following coding structure to denote an instance from this set: $|N_C|$-$|N_0|$- *cluCV*. We consider 22 test instances up to 100 customers (4 instances with 20 customers and 5 depots; 6 instances with 50 customers and 5 depots, 6 instances with 100 customers and 5 depots, 6 instances with 100 customers and 10 depots) in the LRP test set.

In order to generate the delivery and pickup demands of the customers in each test instance, we utilize demand separation approaches proposed by Salhi and Nagy (1999) and Angelelli and Mansini (2002) (denoted as SN and AM, respectively). These approaches are briefly defined as follows. In SN approach, a ratio $r_i = min(x_i / y_i; y_i / x_i)$, where x_i and y_i are the coordinates of customer *i*, is calculated for each customer *i*, and then the delivery and pickup demands are generated as $d_i = r_i * q_i$ and $p_i = q_i - d_i$, where q_i is the original demand of customer *i*. We refer this type of problems as *X*. Similarly, another type of problem, called *Y*, is generated by exchanging delivery and pickup demands of each customer. In AM approach, the original demand of each customer *i* is considered as delivery demand ($d_i = q_i$) and the pickup demand of the corresponding customer is generated as $p_i = \lfloor (1-\gamma) q_i \rfloor$ if *i* is even and $p_i = \lfloor (1+\gamma) q_i \rfloor$ if *i* is odd. In this paper, we consider two γ values as 0.2 and 0.8 to generate two different types of problems called *Z* and *W*, respectively. As a result, 88 LRPSPD test instances have been generated by using 22 original LRP test instances and 4 different separation strategies (*X, Y, Z* and *W* type).

The proposed algorithm was coded using C++ programming language and all experiments were performed on Intel Xeon 3.16 GHz equipped with 1 GB RAM computer. Based on our preliminary experiments, we implement following parameters in the hGA: population size is taken as $|N_C|+|N_0|$, crossover and mutation rates are set to 1 and 0.05, respectively, finally the number of generations is set to $|N_C|*|N_0|$. The initial temperature (T_0) in SA is taken as 665 in which an inferior solution (inferior by 70% relative to current solution) is accepted with a probability of 0.90, the final temperature (T_f) is set to 0.15 such that a solution which is inferior by 1% relative to current solution is accepted with a probability of 0.001. The number of neighbors (LS) generated by each moving strategy is set to $|N_C|$ and finally cooling rate is considered as 0.95. Because of the stochastic nature of the hGA, it is run 10 times for each instance with different random seeds. Meantime, it is worthy to node that B&C algorithm in Karaoglan et al. 2011 for the LRPSPD had been also run at the same computer and its computation time had been limited with four hours for each LRPSPD instance.

In the comparison of the results of the hGA and B&C algorithm, following performance measures are used: percentage gap and computation time. The percentage gap is calculated as $100 \cdot [(UB - LB)/LB]$ where LB is the lower bound obtained by B&C algorithm within four hours and upper bound (UB) is the optimal or best solution obtained by B&C algorithm or the proposed hGA. Table 1 and Table 2 present computational results for the hGA and the B&C algorithm on LRPSPD instances obtained by using SN and AM approach, respectively. In these tables, the first column represents the name of the LRPSPD instances. The next column, named *DSS*, stands for the demand separation strategy. Subsequent three columns report the minimum, average and maximum values of percentage gap obtained by the hGA. In the sixth column, percentage gap of upper bound obtained by B&C algorithm is reported. Finally, last two columns stand for the computation times of both solution methods. As seen from tables, good quality solutions are obtained by the proposed hGA in a very short computation time. The hGA improves upper bounds ob-

Table 1. Computational results for the LRPSPD test instances obtained by SN approach.

Problem Name	DSS	hGA gap %			B&C gap %	Computation Time	
		Min	Avg	Max		GA	B&C
20-5-1a	X	0.00	0.00	0.00	0.00	8.29	84.51
	Y	0.00	0.01	0.01	0.00	8.24	73.77
20-5-1b	X	0.00	0.00	0.00	0.00	7.73	1.86
	Y	0.00	0.00	0.00	0.00	7.61	0.27
20-5-2a	X	0.00	0.01	0.01	0.00	7.34	27.93
	Y	0.00	0.00	0.00	0.00	7.41	16.02
20-5-2b	X	0.00	0.00	0.00	0.00	6.76	1.61
	Y	0.00	0.00	0.00	0.00	7.18	1.38
50-5-1a	X	0.17	0.20	0.25	0.17	43.29	14400.00
	Y	0.17	0.19	0.21	0.22	43.63	14400.00
50-5-1b	X	0.04	0.04	0.04	0.04	45.87	14400.00
	Y	0.05	0.05	0.05	0.04	46.02	14400.00
50-5-2a	X	0.09	0.09	0.09	0.10	46.81	14400.00
	Y	0.09	0.09	0.10	0.09	45.91	14400.00
50-5-2b	X	0.00	0.00	0.00	0.00	44.11	213.75
	Y	0.00	0.00	0.00	0.00	44.55	1147.76
50-5-3a	X	0.24	0.25	0.26	0.24	35.42	14400.00
	Y	0.23	0.24	0.25	0.25	35.74	14400.00
50-5-3b	X	0.09	0.09	0.09	0.12	42.30	14400.00
	Y	0.03	0.03	0.03	0.03	42.25	14400.00
100-5-1a	X	0.16	0.18	0.20	0.18	232.71	14400.00
	Y	0.16	0.18	0.20	0.17	231.53	14400.00
100-5-1b	X	0.09	0.10	0.11	0.12	185.47	14400.00
	Y	0.09	0.10	0.11	0.12	185.42	14400.00
100-5-2a	X	0.11	0.13	0.15	0.11	296.54	14400.00
	Y	0.12	0.13	0.14	0.11	299.33	14400.00
100-5-2b	X	0.06	0.07	0.07	0.08	286.23	14400.00
	Y	0.06	0.07	0.08	0.07	288.04	14400.00
100-5-3a	X	0.15	0.18	0.22	0.17	155.41	14400.00
	Y	0.15	0.53	1.93	0.22	157.50	14400.00
100-5-3b	X	0.08	0.09	0.09	0.11	114.41	14400.00
	Y	0.08	0.08	0.09	0.09	115.28	14400.00
100-10-1a	X	1.05	1.06	1.08	1.07	265.60	14400.00
	Y	1.04	1.05	1.07	1.07	256.65	14400.00
100-10-1b	X	0.06	0.06	0.06	0.08	232.63	14400.00
	Y	0.05	0.05	0.06	0.07	231.38	14400.00
100-10-2a	X	0.13	0.71	1.08	32.08	228.73	14400.00
	Y	0.15	0.90	1.13	1.97	230.26	14400.00
100-10-2b	X	1.08	1.09	1.09	1.08	194.59	14400.00
	Y	1.09	1.09	1.09	1.08	191.48	14400.00
100-10-3a	X	0.21	0.76	1.12	1.11	136.15	14400.00
	Y	0.16	0.76	1.15	0.33	144.39	14400.00
100-10-3b	X	0.06	0.06	0.07	0.07	93.76	14400.00
	Y	0.05	0.05	0.06	0.10	96.88	14400.00
Average		**0.17**	**0.24**	**0.31**	**0.98**	**123.34**	**11162.93**

Table 2. Computational results for the LRPSPD test instances obtained by AM approach.

Problem Name	DSS	hGA gap %			B&C gap %	Computation Time	
		Min	Avg	Max		GA	B&C
20-5-1a	W	0.04	0.04	0.04	0.00	8.33	232.48
	Z	0.00	0.00	0.01	0.00	7.27	75.24
20-5-1b	W	0.00	0.00	0.01	0.00	8.45	3.22
	Z	0.00	0.00	0.00	0.00	8.04	0.94
20-5-2a	W	0.00	0.00	0.00	0.00	7.61	37.79
	Z	0.00	0.00	0.00	0.00	7.34	73.81
20-5-2b	W	0.00	0.00	0.00	0.00	7.22	7.09
	Z	0.00	0.00	0.00	0.00	7.56	4.40
50-5-1a	W	10.79	12.75	13.98	12.28	46.46	14400.00
	Z	0.21	0.23	0.24	0.30	44.24	14400.00
50-5-1b	W	0.16	0.16	0.17	0.21	43.98	14400.00
	Z	0.14	0.15	0.15	0.18	39.53	14400.00
50-5-2a	W	2.57	2.59	2.60	2.54	45.06	14400.00
	Z	0.12	0.12	0.12	0.16	42.31	14400.00
50-5-2b	W	0.12	0.12	0.12	0.13	44.71	14400.00
	Z	0.11	0.11	0.11	0.11	42.69	14400.00
50-5-3a	W	4.77	4.81	4.83	4.65	34.91	14400.00
	Z	0.22	0.24	0.25	0.26	32.56	14400.00
50-5-3b	W	0.21	0.21	0.21	0.21	37.08	14400.00
	Z	0.13	0.13	0.13	0.13	36.46	14400.00
100-5-1a	W	0.78	1.11	1.33	1.36	228.54	14400.00
	Z	0.69	0.75	0.77	0.73	244.42	14400.00
100-5-1b	W	0.78	0.79	0.80	0.78	224.39	14400.00
	Z	0.75	0.76	0.76	0.76	204.08	14400.00
100-5-2a	W	1.82	2.21	2.68	28.20	235.75	14400.00
	Z	2.48	2.49	2.50	30.38	278.27	14400.00
100-5-2b	W	1.03	1.04	1.05	29.57	289.37	14400.00
	Z	1.00	1.01	1.02	29.55	280.93	14400.00
100-5-3a	W	2.31	2.83	3.00	3.65	179.62	14400.00
	Z	1.98	2.01	2.02	1.89	197.99	14400.00
100-5-3b	W	1.18	1.21	1.24	2.13	167.97	14400.00
	Z	1.12	1.14	1.15	1.11	156.90	14400.00
100-10-1a	W	0.97	1.34	1.44	1.41	222.86	14400.00
	Z	0.51	0.51	0.52	0.53	247.63	14400.00
100-10-1b	W	0.05	0.15	0.52	0.10	258.27	14400.00
	Z	0.03	0.04	0.04	0.04	246.39	14400.00
100-10-2a	W	1.27	1.28	1.31	22.30	217.62	14400.00
	Z	1.45	1.46	1.47	22.42	234.41	14400.00
100-10-2b	W	1.18	1.19	1.20	1.81	213.72	14400.00
	Z	1.39	1.39	1.39	22.78	200.96	14400.00
100-10-3a	W	8.02	17.56	23.91	20.88	179.52	14400.00
	Z	1.94	2.29	2.52	2.44	193.26	14400.00
100-10-3b	W	7.59	11.03	24.78	21.58	143.55	14400.00
	Z	1.31	1.33	1.34	1.37	132.17	14400.00
Average		**1.39**	**1.79**	**2.31**	**6.11**	**130.24**	**11791.70**

tained by the B&C algorithm for 52 out of 88 LRPSPD test instances. While average gaps of the hGA are 0.24% and 1.79% for the SN and AM demand separation procedures, respectively, these values increase to 0.98% and 6.11%, respectively, for the B&C algorithm. Moreover, as seen from the average of the maximum percentage gaps, (0.31% and 2.31% for the SN and AM demand separation procedures, respectively) even the worst case performance of the hGA is better than the B&C algorithm. When both approaches are compared in terms of the average computation time, it is seen that the hGA needs 125 seconds in average to reach optimal or near optimal solutions while the B&C algorithm consumes approximately 11000 seconds in average to solve LRPSPD instances. These results show that the hGA is superior to the B&C algorithm. Moreover, it is seen that the minimum and maximum percentage gaps of the hGA for each instance are very close to each other. This result is also an indicator that the hGA is robust to initial random numbers.

5. CONCLUSION

In this paper, we have considered general case of the location-routing problem called location-routing problem with simultaneous pickup and delivery, LRPSPD, and proposed a hybrid heuristic approach based on GA and SA to solve the problem. We have investigated the performance of the proposed algorithm on a set of instances derived from the literature and compared its results with those obtained by branch and cut algorithm developed for the LRPSPD. Computational results indicate that good quality solutions (average gaps are 0.24% and 1.79%) are obtained by the proposed hGA in a reasonable computation time. Further research can be performed on the development new crossover and mutation operators for the encoding structure and new moving strategies for the SA to increase the performance of the hGA. Moreover, the proposed hGA can be used to obtain an initial solution for any exact algorithm (i.e. branch and bound, branch and cut) to shorten the optimization process of exact algorithm.

ACKNOWLEDGMENT

This research is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under Grant No 108E069.

REFERENCES

Albareda-Sambola, M., Diaz, J. A., and Fernandez, E. (2005), A compact model and tight bounds for a combined location-routing problem, *Computers and*

Operations Research, **32**, 407-428.

Angelelli Angelelli, E. and Mansini, R. (2002), The vehicle routing problem with time windows and simultaneous pick-up and delivery, In Klose, A., Speranza, M. G., and Van Wassenhove, L. N. (Eds.), *Quantitative Approaches to Distribution Logistics and Supply Chain Management* (249-267). Berlin: Springer-Verlag.

Belenguer, J.-M., Benaaventt, E., Prins, C., Prodhon, C., and Wolfler-Calvo, R. (2006), A Branch and Cut method for the Capacitated Location-Routing Problem, *International Conference on Service Systems and Service Management*, 1541-1546.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., and Laporte, G. (2007), Static pickup and delivery problems: a classification scheme and survey, *TOP*, **15**, 1-31.

Berger, R. T., Coullard, C. R., and Daskin, M. S. (2007), Location-Routing Problems with Distance Constraints, *Transportation Science*, **41**, 29-43.

Clarke, C. and Wright, J. Q. (1964), Scheduling of vehicle from a central depot to a number of delivery points. *Operations Research*, **12**, 568-581.

Duhamel, C., Lacomme, P., Prins, C., and Prodhon, C. (2010), A GRASP×ELS approach for the capacitated location-routing problem, *Computers and Operations Research*, **37**, 1912-1923.

Gen, M. and Cheng, R. (1997), *Genetic algorithms and engineering design*, NewYork: Wiley.

Hansen, P. H., Hegedahl, B., Hjortkjær, S., and Obel, B. (1994), A heuristic solution to the warehouse location-routing problem, *European Journal of Operational Research*, **76**, 111-127.

Karaoglan, I., Altiparmak, F., Kara, I., and Dengiz, B. (2009a), Formulations for Location-Routing Problem with Simultaneous Pickup and Delivery, Research Paper, http://w3.gazi.edu.tr/~fulyaal/Papers/LRPSPD_MIPFormulations.pdf.

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

Laporte, G., Nobert, Y., and Pelletier, P. (1983), Hamiltonian location problems, *European Journal of Operational Research*, **12**, 82-89.

Laporte, G., Nobert, Y., and Arpin, D. (1986), An exact algorithm for solving a capacitated location-routing problem, *Annals of Operations Research*, **6**, 293-310.

Laporte, G. (1988), Location-routing problems. In Golden, B. L. and Assad, A. A. (Eds.), *Vehicle Routing*: *Methods and Studies* (163-198), Amsterdam: North-Holland.

Marinakis, Y. and Marinaki, M. (2008), A Particle Swarm Optimization Algorithm with Path Relinking for the Location Routing Problem, *Journal of Mathematical Modelling and Algorithms*, **7**, 59-78.

Melechovsky, J. and Prins, C. (2005), A metaheuristic to solve a location routing problem with non-linear costs, *Journal of Heuristics*, **11**, 375-391.

Min, H., Jayaraman, V., and Srivastava, R. (1998), Combined location-routing problems: A synthesis and future research directions, *European Journal of Operational Research*, **108**, 1-15.

Mosheiov, G. (1994), The Travelling Salesman Problem with pick-up and delivery, *European Journal of Operational Research*, **79**, 299-310.

Nagy, G. and Salhi, S. (1998), The many-to-many location-routing problem. *TOP*, **6**, 261-275.

Nagy, G. and Salhi, S. (2007), Location-routing: Issues, models and methods, *European Journal of Operational Research*, **177**, 649-672.

Parragh, S. N., Doerner, K. F., and Hartl, R. F. (2008), A survey on pickup and delivery problems, *Journal für Betriebswirtschaft*, **58**, 81-117.

Perl, J. and Daskin, M. S. (1984), A unified warehouse location-routing methodology, *Journal of Business Logistics*, **5**, 92-111.

Perl, J. and Daskin, M. S. (1985), A warehouse location-routing problem, *Transportation Research Part B*, **19**, 381-396.

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

Prins, C., Prodhon, C., and Calvo, R. (2006a), A Memetic Algorithm with Population Management (MA|PM) for the Capacitated Location-Routing Problem, In Gottlieb, J. and Raidl, G. (Eds.), *Evolutionary Computation in Combinatorial Optimization*, Springer Berlin/Heidelberg, 183-194.

Prins, C., Prodhon, C., and Calvo, R. W. (2006b), Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking, *4OR*, **4**, 221-238.

Prodhon, C. (2008), http://prodhonc.free.fe/homepag.

Salhi, S. and Nagy, G. (1999), A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling, *Journal of the Operational Research Society*, **50**, 1034-1042.

Salhi, S. and Rand, G. K. (1989), The effect of ignoring routes when locating depots, *European Journal of Operational Research*, **39**, 150-156.

Srivastava, R. (1993), Alternate solution procedures for the location-routing problem, *Omega*, **21**, 497-506.

Srivastava, R. and Benton, W. C. (1990), The location-routing problem: Considerations in physical distribution system design, *Computers and Operations Research*, **17**, 427-435.

Suman, B. and Kumar, P. (2006), A survey of simulated annealing as a tool for single and multiobjective optimization, *Journal of the Operational Research Society*, **57**(10), 1143-1160.

Tuzun, D. and Burke, L. I. (1999), Two-phase tabu search approach to the location routing problem, *European Journal of Operational Research*, **116**, 87-99.

Wu, T. H., Low, C., and Bai, J. W. (2002), Heuristic solutions to multi-depot location-routing problems, *Computers and Operations Research*, **29**, 1393-1415.

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