



Received: 31 January 2018
Accepted: 06 April 2018
First Published: 17 April 2018

*Corresponding author: Ji Ung Sun, Department of Industrial and Management Engineering, Hankuk University of Foreign Studies, Global campus 81, Oedae-ro, Mohyeon-eup, Cheoin-gu, Yongin-si, Gyeonggi-do, 17035 Korea
E-mails: jusun@hufs.ac.kr, jeddy4u@gmail.com

Reviewing editor:
Wei Meng, Wuhan University of Technology, China

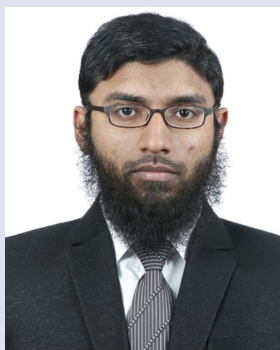
Additional information is available at the end of the article

PRODUCTION & MANUFACTURING | RESEARCH ARTICLE

A novel approach to combine the hierarchical and iterative techniques for solving capacitated location-routing problem

A.K.M. Foysal Ahmed¹ and Ji Ung Sun^{1*}

Abstract: This study focuses on the capacitated location-routing problem (CLRP), which is a combination of capacitated facility location problem and capacitated vehicle routing problem. We propose a novel approach to find solutions for CLRP, a common version of location routing problem, founded in the literature. We have proposed an enhanced version of particle swarm optimization (PSO), a swarm inspired metaheuristic to handle CLRP. The proposed approach consists of unique assignment techniques of the customers to the opened depots and a tri-fold PSO based searching strategy which combines the influence of both hierarchical and iterative techniques in order to find near optimal solutions to a CLRP. Two folds of PSO are for maintaining the global view of clustering the nodes and the remaining PSO fold keeps the local nature of prioritizing route construction cost while making a complete solution of CLRP, thus proposed approach preserves the influence of both hierarchical and iterative methods. Experimental performance evaluation of the proposed approach is compared to other particle swarm optimization based methods to solve benchmark instances available in literature which show better performance of the proposed method.



A.K.M. Foysal Ahmed

ABOUT THE AUTHORS

Ji Ung Sun is currently working as a professor in the department of Industrial and Management Engineering, Hankuk University of Foreign Studies, Republic of Korea and is the corresponding author of this article. I, A.K.M. Foysal Ahmed has pursued Ph.D. under the supervision of Professor Ji Ung Sun on Industrial and Information System Engineering. Our present interest is about handling the combinatorial optimization problems which are well-known and relevant problems in the various fields of industrial engineering such as operation research, supply chain management and so on. Our motivation is to contribute with new, innovative and advanced idea to enhance and improve the existing metaheuristic techniques as well as to introduce new metaheuristic algorithms to deal large size, practical, realistic and complex problems effectively and efficiently. In future, we are really keen to explore the characteristics of nature based metaheuristics and tackle more complex problems.

PUBLIC INTEREST STATEMENT

This article is about a combined problem of facility location problem, FLP and vehicle routing problem, VRP namely location routing problem, CLRP. FLP is to search the best depot positions among the candidate depots and VRP is to find routes to serve the demand associated customers on a graph or, map. Hubs and vehicles are assumed to have restriction on their capacity thus the combined problem is called CLRP. There are three kinds of techniques to solve the CLRP i.e. clustering technique which is greatly biased on clustering quality, hierarchical technique where FLP is solved first then VRP causing no real influence of VRP on FLP solution and lastly iterative technique where both FLP and VRP are solved simultaneously therefore FLP gets no priority which may cause VRP to bias the FLP solution critically. A unique technique maintaining both effects of hierarchical and iterative approaches eliminating their drawbacks is proposed.

Subjects: Industrial Engineering & Manufacturing; Supply Chain Management; Transport & Vehicle Engineering

Keywords: capacitated location-routing problem; tri-fold particle swarm optimization; combined hierarchical-iterative approach; hub based clustering; nearer neighbor based decoding technique

1. Introduction

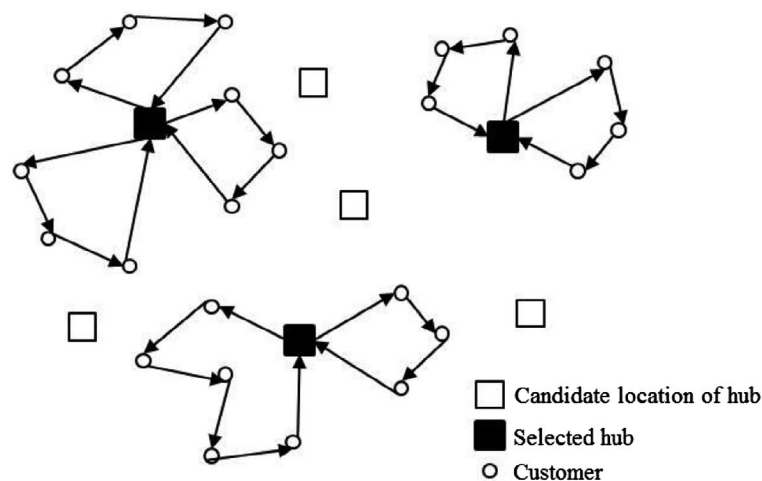
Capacitated location-routing problem (CLRP) is a conjunction of two combinatorial optimization problems namely facility location problem (FLP) and vehicle routing problem (VRP). The increased amount of applications force researchers to consider this combined optimization problem together. This is very useful in many transportation network such as food and drink distribution (Watson-Gandy & Dohrn, 1973), airline network design (Chan & Hearn, 1977), parcel delivery (Bruns, Klose, & Stähly, 2000), telecommunication network design (Billionnet, Elloumi, & Djerbi, 2005), locating blood banks (Or & Pierskalla, 1979), rubber plant location (Nambiar, Gelders, & Van Wassenhove, 1981) and so on. Huge attraction and applicability of CLRP drags remarkable attention of the researchers from the field of logistics, supply chain management and so on.

CLRP is obviously a NP-Hard problem as mentioned by Nagy and Salhi (2007). Researchers usually approach to solve the CLRP as two separate design problems (Laporte, 1992; O’Kelly, 1986). FLP is traditionally considered as long-term planning issue. VRP, on the other hand, is a short-term planning which may require to be solved on daily basis. However, when the time horizon for FLP is not too long and the associated cost is comparable to VRP, both problems can also be dealt together as a combined problem. A combination of these two problems, commonly referred to as location—routing problem (LRP) (Nagy & Salhi, 2007; Salhi & Nagy, 2009), is more realistic as route cost is a vital issue to decide hub locations and routes are obviously dependent on the positions of the hubs.

In this study, a common CLRP is considered where a solution need to choose a set of capacitated hubs among a list of candidate hubs, assign every customer to exactly one single hub, and design some routes of capacitated vehicles from every hub to serve the demand of the customers assigned to it by visiting them exactly once. A vehicle should return to the hub after satisfying the customers assigned to it. The growing trend of online shopping has added more weight on contributing on such field of Location-Routing Problems (LRP) which is similar to multi hub and candidate node network as portrayed in Figure 1.

Hierarchical technique is found to be a very common approach to handle CLRP (Albareda-Sambola, Díaz, & Fernández, 2005; Melechovský, Prins, & Calvo, 2005). In hierarchical techniques, FLP is considered as the main problem to be solved first and VRP is considered as the subordinate problem to

Figure 1. Typical portrayal of multi hub network for LRP.



be solved later for every hub and associated customer nodes. Approximated costs of routes are considered while solving the FLP. If VRP is very negligible compared to the FLP then it may not affect the FLP solution much. However, the present nature of delivery service is seen to be over crowded with online ordered shipment delivery. This raises an urge to take the VRP part of the problem more seriously even during solving the FLP part. Iterative approaches try to handle this problem by simultaneously considering FLP and VRP as two sub-problems. Here the candidate solutions along with other necessary information are passed back and forth between the two sub-problems aiming at reaching a better optimal. The very big drawback of iterative approaches is that they offer equal significance to the FLP and VRP where VRP may bias the FLP greatly. We have proposed a technique to consider the global aspects of CLRP first and guide this CLRP with the solution of VRP by solving the VRP separately to track better routes of transports in any iteration. We believe this will impact the proper effect of VRP in solving CLRP with the influence of both hierarchical and iterative solution approaches hence overcome the drawbacks of the traditional iterative and hierarchical approaches.

As CLRP is a NP-Hard problem, it is difficult to find exact algorithms to handle a big size problem whereas in literature there are evidence of exact algorithms dealt small to medium size problems. In many approaches it is very common to take an advantage of metaheuristics to solve CLRP which is closer to real life problems (Nagy & Salhi, 2007). In this article, a tri-fold particle swarm optimization (TPSO) is proposed to solve the CLRP. According to our best knowledge any algorithm is never been applied in such combined hierarchical-iterative fashion.

We have adopted a concept of pool to explore the solutions with local search operators which was first introduced for handling capacitated vehicle routing problem (Ahmed & Sun, 2018). According to our best knowledge, this is for the first time in any literature of handling capacitated location routing problem where a pool is utilized to explore neighborhood solution space with local search operators. The pool is made up with the best solution of each iteration though there is a refining technique is utilized to avoid continuous increment of the size of the pool. The refining pseudocode is described in our previous article (Ahmed & Sun, 2018). We believe the solution quality increases due to utilizing the pool which explore better solutions in the neighborhood of the local best of each iteration if any.

Structure of this study from here on is a literature review in Section 2 including recent advancement of metaheuristics solving CLRP and recent development of PSO related to CLRP. Section 3 presents the algorithm implementation. Later, in Section 4 computational evaluation is presented with the comparison among proposed TPSO and PSO-based approaches found in the literatures. This paper concluded in Section 5 along with future research directions.

2. Related literature review

A problem similar to CLRP was first introduced in 1964 (Maranzana, 1964). The mathematical modeling and methods for solving CLRP got much attractions from the researchers in around 1980s (Chan & Hearn, 1977; Drezner, 1982; Drezner & Wesolowsky, 1982; Kolen, 1985). Afterwards, the literature has become rich with many different variations and extensions of CLRP as well as many exact and heuristic algorithms to solve them (Alumur & Kara, 2008; Drexl & Schneider, 2015; Nagy & Salhi, 2007; Prodhon & Prins, 2014). The available techniques to solve CLRP can broadly be categorized into three approaches: clustering approach, iterative approach and hierarchical approach.

The clustering techniques first partition the customer nodes into clusters and a depot is made responsible for a cluster, and then a vehicle route searching method is executed in each cluster independently (Billionnet et al., 2005; Branco & Coelho, 1990; Min, 1996; Nambiar et al., 1981). This is very effective when clustering algorithm can make very good clusters along with the depot location (Sun, 2013). However, the clustering part of the problem does not consider the actual effective routes to be followed by the vehicles, and hence the clusters may not lead to optimized costs of serving the customers. The iterative and hierarchical techniques take advantage of overcoming such drawbacks compared to clustering method to show better results as reported in the literatures

(Prodhon, 2010) that optimal solutions found till today for the benchmark problems are often obtained by iterative approach (Melechovský et al., 2005) or hierarchical approach (Wu, Low, & Bai, 2002).

The hierarchical approach is a strategical approach to handle the FLP as a main problem and deal with the VRP as a sub-problem (Perl & Daskin, 1985; Salhi & Fraser, 1996; Schwardt & Dethloff, 2005; Wu et al., 2002). To obtain reasonable results these two problems cannot be considered independently since the possible routes also need to be considered during solving the FLP. If each customer demand is equal to full-truck-load (TL) then usual technique is to consider a vehicle routing cost as the cost of going directly to and coming back (Karaoglan, Altiparmak, Kara, & Dengiz, 2012). However, when the customer demands are usually less than a truck-load (LTL) then more than one customer can be served by a vehicle. In such cases, the routes of the vehicles in the clusters have a great impact on the overall cost of a solution, but often not easy to consider optimal routes in the FLP since it incurs a huge computation to find the routes in the solution space. So, usual trend is to use some approximate route cost in the objective function to solve FLP. After deciding the locations of the facilities, the algorithm considers the vehicle routing sub-parts of the problem to find optimal routes. Thus, hierarchical fashion may not always be realistic since the solution to the FLP may not be optimal due to the approximation of the route costs (Salhi & Rand, 1989).

The iterative approach simultaneously considers FLP and VRP as sub-problems (Albareda-Sambola et al., 2005; Melechovský et al., 2005). This solution approach deals the CLRP by passing the information of one sub-problem to other for interactive decision making strategy. The drawback of hierarchical approach may be overcome by iterative approach apparently as the node clustering, hub location and vehicle routing decisions iterate more than one time to find the solution. For an example, the simulated annealing-based method presented in (Vincent, Lin, Lee, & Ting, 2010) assigns the customers to their nearest hubs until the hub capacity be violated. Then a route is constructed by a travelling salesman problem (TSP) solving technique (Lin & Kernighan, 1973). Afterwards, the obtained route is broken into parts considering the vehicle capacity. This whole process iterates until each customer is successfully assigned to a hub as well as to a route. The shortcoming in such approaches is that the breaking of the TSP solution may not preserve the optimality of the solution achieved. Moreover, though the process is an iterative one, the basic flow of the clustering and route finding is done sequentially, similar to the hierarchical approach, and thus the aforementioned shortcomings may still prevail.

Population-based or multi-agent metaheuristics have also gained much popularity among the researchers to solve different NP hard problems including CLRP. Genetic algorithms design chromosomes capturing the hub locations and the routing solutions along with ad hoc crossover and mutation operators (Derbel, Jarbouj, Hanafi, & Chabchoub, 2012). Prins, Prodhon, and Calvo (2006a) presented the memetic algorithm/population management (MA|PM) based approach for the CLRP. Here a chromosome represents depot status and customer status. Upon getting the hub selection routes are constructed based on the procedure described in Prins, Prodhon, and Calvo (2004). Ting and Chen (2013) apply three different ant colony optimization (ACO) algorithms sequentially to select hub locations, assign customers to hubs and find routes for the customers to be served by the respective hubs.

Many hybrid methods that combine different approaches are also been proposed. Wu et al. (2002) combined tabu search (TS) and simulated annealing (SA) while Lin, Chow, and Chen (2002) used threshold accepting and SA to decide facility locations and vehicle routing. Bouhaf, Hajjam, and Koukam (2006) combined SA with ACO in order to solve the CLRP. Another approach proposed a two phase algorithm (Prins, Prodhon, & Calvo, 2006b) which applies greedy randomized adaptive search procedures (GRASP) and path relinking. A GRASP with evolutionary location search approach was presented by Duhamel, Lacomme, Prins, and Prodhon (2010). Several hierarchical and nonhierarchical clustering techniques are integrated in a sequential heuristic algorithm (Barreto, Ferreira, Paixao, & Santos, 2007). Marinakis and Marinaki (2008) proposed another hybrid algorithm where they

combined the PSO, multiple phase neighborhood search-GRASP, the expanding neighborhood search and path relinking to solve the LRP.

In CLRP, the quality of the selection of hub location and the formation of the routes depend of each other. The influence of the hub selection on the route construction is usually considered in the traditional approaches where the route formation is done after the hub selection while the opposite is usually not (or very less) considered. However, costs of vehicle routes indeed affect the worthiness of the hub location as well as customer-allocation decisions. Consequently, dealing with the interdependence between these two decisions are very important issues to be taken care of in solving CLRP, which we attempt in this work. We propose a combined hierarchical-iterative approach utilizing tri-fold PSO algorithm to handle CLRP. At first the algorithm looks at the global view of the solution with a bi-fold PSO while in the other layer focuses on the route construction in the local PSO. The information of the constructed routes in any iteration is fed back to the FLP layer so that it can look for better candidate solutions in the next iteration.

This global and local view will preserve the hub location and customer assignment information as well as the near optimal routes found in any iteration in the runtime of the algorithm. We believe that our algorithm can solve more complex or bigger problems than those traditional approaches. Although the computational complexity will be more than the traditional inefficient methods but this doesn't exceed few minutes to optimize a problem of hundred or more customers and few hubs.

3. The proposed combined hierarchical-iterative approach

Traditional hierarchical approaches handle CLRP as two different problems namely FLP and VRP, and try to solve them one after another: deciding the hub locations first followed by solving the VRP for each hub. Such methods do not have any mechanism to update the hub selections later, based on the experience of solving the VRPs in the customers selected for different hubs, and hence it may limit the search horizon. On the other hand, traditional iterative methods are based on generating and updating population of candidate solutions. Here, the problem is approached as a whole, and hence local neighborhood of a candidate solution, e.g. a better set of routes for the VRP for the hubs at hand, may not be well explored.

In our proposal, we make an Iterative-Hierarchical approach to hybridize the hierarchical and the iterative approaches to influence the search procedure by the desirable properties of both the approaches. We adopt the PSO, a swarm-based optimization technique inspired by bird flocking or fish schooling behavior of swarms, to search the solution in our work. In the proposed approach, a bi-fold global PSO searches for solution of FLP, and another fold of local PSO is dedicated to optimize the tracks. Thus, in the proposed tri-fold PSO (TPSO) technique, routes are optimized at least partially for a set of selected hubs and an allocation of customer nodes to them, as done in the second stage (i.e. VRP) of hierarchical approaches. Additionally, the global procedure keeps on generating more candidate solutions consisting of different sets of selected hubs and customer node allocations. Thus the proposed approach broadens the search space, which is expected to lead to better solutions, as compared to the traditional approaches.

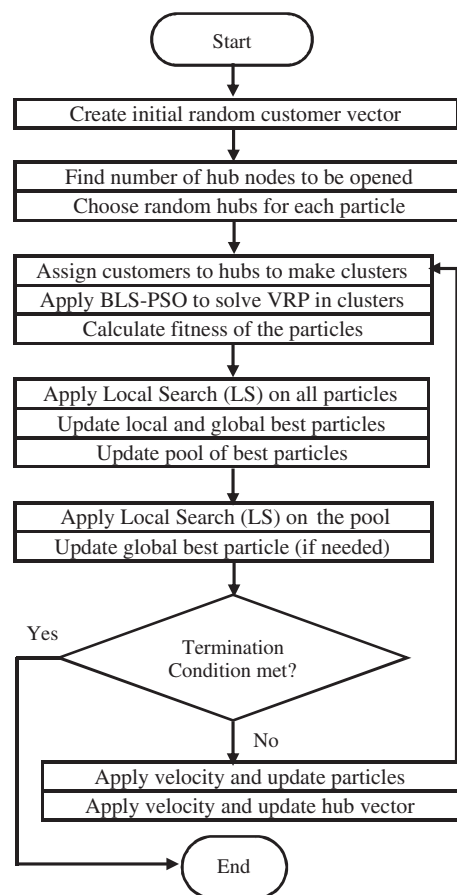
3.1. Tri-fold PSO algorithm for CLRP

The eligibility of the PSO algorithm has already been proven in a good number of application areas (Banks, Vincent, & Anyakoha, 2007, 2008). Though PSO was basically developed to handle continuous problems (Eberhart & Kennedy, 1995), several approaches are available in the literature to discretize the PSO (Kennedy & Eberhart, 1997; Tan, Gao, & Zeng, 2004; Wang, Huang, Zhou, & Pang, 2003) to make PSO applicable to the discrete problems such as VRP (Ai & Kachitvichyanukul, 2009), LRP (Marinakis & Marinaki, 2008) and so on. There are several notations required to explain the model proposed which are introduced in Table 1.

Table 1. Nomenclature of the notations used in this article

Notations	Description
N	Total number of customers
P	Number of candidate hubs in a problem statement
ρ	Minimum hubs required to satisfy the total demand of customers
p	Number of hubs to be opened for a solution
t	Current iteration number
T	Maximum number of iteration; Termination criteria
C	Customer Vector, a permutation of all customer nodes
H	Hub Vector, a binary vector representing opened hubs
CC	Customer cluster vector
g	Indices of particles
j	Indices of clusters
V	Velocity for any particle to move from current to new position
Z	Population size

Figure 2. Sketch of the Tri-fold PSO in the proposed Hierarchical-Iterative approach.



In TPSO, a set of particles is randomly generated in the solution space where each particle, $C = \{c_n | n = 1, 2 \dots N\}$, represents an encoded candidate solution of the CLRP. The particles keep on changing their positions (moving to a different candidate solution) through different iterations of the algorithm to find better solutions according to their velocity. The initial set of particles, called the population and associated velocity is generated at random. Figure 2 shows a sketch of how we apply the

global bi-fold PSO, which also includes the local PSO stage in it. The global bi-fold PSO generates different selections of hubs along with the allocation of different sets of customers to them in an encoded vector forms in the particles. After decoding a particle, we get the comprehensive representation of allocations of customers for different selected hubs. Upon getting the allocation of nodes to hubs, a local PSO is applied to each of the selected hubs to solve the VRP for that hub. The solutions provided by the local PSO is combined with the respective particle of the global PSO to interpret a complete candidate solution.

Now, we focus on how we apply PSO to follow the proposed search approach in this work. Let us first focus on the customer vectors in the particles. At this point, we adopt a simple discretization approach of PSO, namely the swap sequence-based PSO (SSPSO) (Wang et al., 2003), in our work. A particle, specifically the customer vector, changes its position (i.e. the values in the particle) following its velocity, represented as a swap sequence (SS), at that time to reach a new position in the solution space using Equation (1) as proposed in Eberhart and Kennedy (1995).

$$C_{g(t)} = C_{g(t-1)} \oplus V_{g(t)}^C \tag{1}$$

where $C_{g(t)}$ is the position (customer vector) and $V_{g(t)}^C$ is the customer velocity of the g th particle at time (or, iteration) t , and \oplus is the swap sequence operator.

The initial customer velocity $V_{g(0)}^C$ of every particle g is also assigned randomly. Every particle keeps track of its personal best solution, known as the local best solution, L_g , throughout the algorithm run time, and updates it when the particle represents a better solution. The swarm also keeps track of the global best solution Γ achieved by the particles over the completed iterations. At every iteration, a particle's customer velocity is updated based on its customer vector L_g^C of local best solution and the customer vector Γ^C of the global best solution by Equation (2) as proposed in Shi and Eberhart (1998).

$$V_{g(t)}^C = \theta_1 V_{g(t-1)}^C \otimes \theta_2 * \text{rand} * (L_g^C \ominus C_{g(t-1)}) \otimes \theta_3 * \text{rand} * (\Gamma^C \ominus C_{g(t-1)}) \tag{2}$$

where, \ominus is the basic swap sequence (BSS) where $(W \ominus X)$ represents the BSS needed to convert a vector X to W . rand is a uniform random value between 0 and 1. θ_i is a random probabilistic value which determines the contribution of the respective swap sequence in the updated SS. \otimes represents the selection operator that select the swap operators from the SSs based on the θ_i values. An example to understand the meaning of the \otimes and \ominus can be found in literature (Ahmed & Sun, 2016). θ_1 is calculated as $\theta_1 = (0.1 - (1 - \sqrt{(1 - 2 * t/T)}) * 1/2)$ as proposed in Akhand, Akter, Rashid, and Yaakob (2015) where t is current iteration and T is maximum iteration specified. θ_2, θ_3 are uniform random numbers between 0 and 1 (with an interval of 0.1).

We now describe the representation and modification of hub selections as the algorithm iterates. The p elements (hub IDs) of a hub vector H_g of particle g are also initially selected from the available hubs and arranged at random. A corresponding velocity vector, $V_{g(t)}^H$, is also initialized randomly. The selection of the hubs is changed based on the velocity using Equation (3) as proposed in Eberhart and Kennedy (1995).

$$H_{g(t)} = H_{g(t-1)} + V_{g(t)}^H \tag{3}$$

Equation (3) follows standard vector algebraic addition formula, where the values are rounded to the nearest integer. After applying Equation (3), a post processing step searches for duplicated entries and invalid hub IDs, and replaces them using random hub IDs. The velocity is updated, following a somewhat similar strategy of updating customer velocity in Equation (4), as proposed in Shi and Eberhart (1998).

$$V_{g(t)}^H = \theta_1 V_{g(t-1)}^H \times \theta_2 * \text{rand} * (L_g^H - H_{g(t-1)}) \times \theta_3 * \text{rand} * (\Gamma^H - H_{g(t-1)}) \quad (4)$$

where θ_i are the same random values used in Equation (2). L_g^H is the hub vector personal best position of particle g , and Γ^H is hub vector the global best particle among all particles till this iteration.

Before explaining the framework of the above described PSO folds, we now look into the VRP local fold of PSO here. After getting the customer vector and hub vector on each iteration, we are able to find the customer allocation according to the clustering technique described later in Algorithm 2. In this stage we have designed a PSO population for each cluster of the customers. Customer cluster gets updated with the similar process described for Equation (1), as proposed in (Eberhart & Kennedy, 1995).

$$CC_{jg(t)} = CC_{jg(t-1)} \oplus V_{jg(t)}^{CC} \quad (5)$$

where $CC_{jg(t)}$ is g th particle of j th cluster on t th iteration. With Equation (5) the customer cluster vector can move from one generation to other generation according the velocity V_{jg}^{CC} , velocity of g th particle of j th cluster. \oplus is the same swap sequence operator explained above. The velocity is also got updated in every iteration according the Equation (6) which is similar to the Equation (2) which is firstly proposed in Shi and Eberhart (1998).

$$V_{jg(t)}^{CC} = \theta_1 V_{jg(t-1)}^{CC} \otimes \theta_2 * \text{rand} * (L_{jg}^{CC} \ominus CC_{jg(t-1)}) \otimes \theta_3 * \text{rand} * (\Gamma_j^{CC} \ominus CC_{jg(t-1)}) \quad (6)$$

where θ_i are the same random values used in Equation (2). L_{jg}^{CC} is the personal best position of particle g in j th cluster and Γ_j^{CC} is the global best particle in j th cluster among all particles till this iteration. Operators \otimes and \ominus are the same operators used in Equation (2). All these PSO folds give encoded solution which are needed to be related with some sort of decoding technique to interpret as a real solution of CLRP. These techniques are explained in the following section.

3.2. TPSO for combined hierarchical-iterative framework

Obtaining the customer vector and the hub vector, we actually have the clusters of nodes based on the selected hubs in encoded form. To make the cluster comprehensive, we apply a decoding algorithm to rearrange the customer vector and represent the cluster boundaries. We assign the customer nodes to their nearest opened hub as showed in Algorithm 2.

The next step following to the assignment of customer nodes in every hub is to set required routes to serve the customer nodes. This problem is handled by the local PSO in our proposed combined hierarchical-iterative approach. This whole process is demonstrated in Algorithm 1.

Algorithm 1. Pseudocode of the combined hierarchical-iterative framework.

```
/* Initialize the Swarm */
1. Find  $p$ , minimum number of hubs required to serve the total demand of customers
2. For  $i = 1$  to  $Z$  do /*  $Z$  is the size of population */
    A. Set  $C_{i(0)}$  to a random permutation of customers
    B. Set  $p$  to a random number in range  $[p, |P|]$ 
    C. Set  $H_{i(0)}$  to a random permutation of  $p$  hubs selected from  $P$ 
    D. Set  $V_{i(0)}^C$  to a random SS
    E. Set  $V_{i(0)}^H$  to a random velocity
3. End for
/* Iteration phase */
4. Set  $t = 0$ 
5. Repeat
    A. For  $i = 1$  to  $Z$  do
        i.  $[C'_{i(t)}, S_{i(t)}] = \text{decode}(C_{i(t)}, H_{i(t)})$ 
        ii. For each cluster  $CC$  defined by  $C'_{i(t)}, H_{i(t)}$  and  $S_{i(t)}$  do
             $[CC_{j(t)}, R_{j(t)}^{CC}] = \text{Solve VRP for cluster } CC$ 
        iii. End for
        iv. Calculate fitness of  $i$ th particle
        v. Apply Local Search (LS) on particle  $i$ 
        vi. Update local best solution  $L_i$  if necessary
    B. End for
    C. Update global best solution  $\Gamma$  if necessary
    D. Add the best of the local best solutions to the pool
    E. Refine and Apply local search on the pool
    F. Update global best solution  $\Gamma$  if necessary
    G.  $t = t + 1$ 
    H. if  $t > \text{max\_iter}$  then return  $\Gamma$ 
    I. Update  $V_{i(t)}^C$  and  $V_{i(t)}^H$  using Equations (2) and (4).
    J. Update  $C_{i(t)}$  and  $H_{i(t)}$  using Equations (1) and (3).
6. Until false
```

The initialization part of Algorithm 1 randomly initializes the population of Z particles. Since the candidate hubs may have different capacities, step 1 of the algorithm finds the minimum number of hubs required to serve all the customers' demands. We do it considering the hubs with the highest capacities. Afterwards, for every particle, we select a random integer in range $[p, |P|]$ as the number of hubs used in a particle. Thus different particle represent solutions with different numbers of hubs. Customer vectors, hub vectors and corresponding velocities of the particles are also set at random.

The iteration steps mainly follow our concept of the global bi-fold PSO by generating new particles through updating the customer vectors and hub vectors of different particles using steps 5.I and 5.J of Algorithm 1. Thus the global PSO iterates to look for better clusters. However, unlike traditional hierarchical approaches, we do not evaluate a cluster formation generated by the global PSO. Rather, we look for the optimized vehicle routes by applying local PSO to solve VRPs for the clusters, at step 5.A(ii), after which the complete solution of CLRP by each particle may obtain.

A customer vector and the corresponding hub vector, generated at steps 5.I and 5.J of Algorithm 1, of a particle jointly contain the clustering formation in an encoded form. This is decoded at step 5A(i), after which we get the customers arranged according to the clusters formed and the hub marker vector indicating the cluster boundaries. The decoding algorithm is depicted in Algorithm 2.

The working principle of this algorithm is simply assigning the customers to their nearest opened hubs if capacity of respective hub permits.

Algorithm 2. Decoding a customer vector to make comprehensive representation of clusters

Inputs

C: customer vector

p: number of opened hubs, i.e. required clusters

Outputs

ClustersMade: a Boolean value Indicating whether clusters could successfully be made

cluster: the clusters made

Procedure

1. Put the first p nodes of C to the p clusters

2. $i = p$;

3. while $i \leq |C|$ do

4. $i = i + 1$; // index of the next node in C to insert in any cluster

5. sortedClusters = cluster indices of sorted in non-increasing order based on distances from the hubs to C(i)

6. assigned = 0;

7. for $j = 1$ to |sortedClusters| do

8. $r = \text{sortedClusters}(j)$;

9. if addition of C(i) to cluster(r) satisfies hub capacity constraint then

10. add C(i) to cluster(r);

11. assigned = 1;

12. go to 15;

13. end if

14. end for

15. if assigned == 0 then // C(i) was not assigned to any cluster in loop from line 7 to 14

16. maxClusterSize = maximum number of nodes assigned to a cluster

17. for $k = \text{maxClusterSize}$ down to 1 do

18. sortedClusters = indices of clusters sorted in non-increasing order based on the increases of the total distances if the kth nodes of the clusters are replaced by C(i)

19. for $j = 1$ to |sortedClusters| do

20. $r = \text{sortedClusters}(j)$;

21. if replacing kth node of cluster(r) by C(i) satisfies hub capacity constraint then

22. add kth node of cluster(r) to the end of C; // to be assigned again

23. replace kth node of cluster(r) by C(i);

24. assigned = 1;

25. go to 30;

26. end if

27. end for

28. end for

29. end if

30. if assigned == 0 then

31. ClustersMade = 0;

32. return;

33. end if

34. end while

35. ClustersMade = 1;

36. return;

3.3. PSO fold for vehicle routing stage

Upon getting clusters for opened hubs, we utilize bilayer local search based PSO (BLS-PSO) to optimize the routes within a cluster (Ahmed & Sun, 2018). In this stage, we generate random particles of customers assigned to a hub to implement BLS-PSO. We then decode the clustered customer vector by utilizing Algorithm 3 to make valid routes for each cluster. Later we have applied two stages of local search and update the local and global best for PSO then the best particle of each generation given another chance putting in the pool. In the pool we have applied the same local searches to exploit more to find better fitness value and update global best value when applicable.

Algorithm 3. Making routes for a cluster

Inputs

CC : customers cluster vector or customers in a hub

// decode for each hub

Outputs

routesMade: a Boolean value Indicating whether routes could successfully been made

routes: the routes made

Procedure

```
1. m = estimateRequiredRoutes(); // Number of vehicle required if we assign customers sequentially one after another to vehicles as they appear in the vector
2. for r = 1 to m do
3.   routes(r) = CC(r); // put the first m nodes in c as the starting nodes of the m routes
4. end for
5. i = m;
6. while i <= |CC| do
7.   i = i+1; // index of the next node in CC to insert in any route
8.   sortedRoutes = indices of routes sorted in non-increasing order based on distances of their last nodes to CC(i)
9.   assigned = 0;
10.  for j = 1 to |sortedRoutes| do
11.    r = sortedRoutes(j);
12.    if addition of CC(i) to the end of routes(r) satisfies capacity constraints then
13.      add CC(i) to the end of routes(r);
14.      assigned = 1;
15.      go to 18;
16.    end if
17.  end for
18.  if assigned == 0 // CC(i) was not assigned to any route in the loop from line 10 to 17
19.    maxRouteSize = maximum number of nodes assigned to a route
20.    for k = maxRouteSize down to 1 do
21.      sortedRoutes = indices of routes sorted in non-increasing order based on the increases of the distances if the kth nodes of the routes are replaced by CC(i)
22.      for j = 1 to |sortedRoutes| do
23.        r = sortedRoutes(j);
24.        if replacing kth node of routes(r) by CC(i) satisfies capacity constraints then
25.          replace kth node of routes(r) by CC(i);
26.          add kth node to the end of CC; // to be assigned again
27.          assigned = 1;
28.        go to 33;
29.      end if
```

Algorithm 3. Making routes for a cluster

```
30. end for
31. end for
32. end if
33.   if assigned == 0 then
34.     routesMade = 0;
35.     return;
36.   end if
37. end while
38.   routesMade = 1;
39. return;
```

3.4. Double layered local search framework

There are two layers of local search incorporated with PSO for finding better solutions in neighborhood in FLP stage as well as in the VRP stage. In FLP stage, the first layer of local search is consists of two stages of local search techniques namely Intra-hub Local Search (Intra-HLS) and Inter-Hub Local Search (Inter-HLS) which are implemented on all the clustered customer vectors. Each of those are composed of two types of local search operators namely insertion method and swapping method. The insertion method picks a node and tries to find a better position to insert it to get a overall better result of CLRP. In this process if any node is found to obtain a better result the move is immediately allowed to be executed rather searching the whole customer list to avoid excessive time consumption. The swapping method picks two neighboring nodes at a time and tries swapping their position if any better solution is found then the move is immediately executed otherwise they are kept in the same position as they were.

Second layer of local search is done with similar strategies like the first layer of local search but this layer has a pool of best particles from different generations. Best particles remain in the pool until and unless it is the best particle among all or the fitness has improved in previous iteration.

The local search technique in VRP is similar to the local search used in FLP stage. We have utilized a bilayer local search here as well. Each layer has Intra-track Local Search (Intra-TLS) and Inter-Track Local Search (Inter-TLS) and both of them have insertion and swapping strategies. First layer of local search is applied on all the particles in any generation whereas second layer of local search is only applied on the pool having the best particles of different iterations. This procedure is similar to the bilayer technique proposed in Ahmed and Sun (2018).

4. Computational experiments

In population-based metaheuristics, a good way of representation of the population and easy application of different operators for evolving them in order to effectively searching the solution space is the key. The representation proposed in this article opens up a new direction to implement PSO-based metaheuristics to easily handle capacitated location-routing problem, and the proposed bilayer PSO provides with an effective way to perform the search for expected solutions. To analyze the performance, we have implemented the proposed method with MathWorks® MATLAB R2017a in an Intel® Pentium® T4300 Dual-Core 2.10 GHz 2.10 GHz (only single core is used) machine having 2 GB of RAM for well-known benchmark set of CLRP instances, namely Barreto instances (Barreto et al., 2007), used in other works found in literature. Marinakis and Marinaki (2008) dealt the set to show the performance of proposed HybPSO-LRP as well. Comparison of performance analysis for solving Barreto instances with our proposed algorithm and the HybPSO-LRP is presented in this section.

We have considered 13 Barreto (Barreto et al., 2007) instances which can be downloaded from http://prodhonc.free.fr/Instances/instances_us.htm. The characteristics of the instances are mentioned in Table 2.

In Table 3, we have tabulated Cost and gap values obtained by the HybPSO-LRP and the proposed TPSO algorithm under consideration on the Barreto instances (Barreto et al., 2007). Here, gap can be calculated

Table 2. Properties of the Barreto instances (Barreto et al., 2007)

Number of hubs	Number of customers	Demand of customers	Capacity of hubs	Capacity of transports	Customer location
{5, 8, 10}	{21, 22, 27, 29, 32, 36, 50, 75, 88, 100, 134, 150}	Distribution mentioned in dataset Prodhon (2010)	Distribution mentioned in dataset Prodhon (2010)	{160, 200, 250, 850, 2,500, 4,500, 6,000, 8,000, 11,000, 8,000,000, 9,000,000}	Randomly generated

Table 3. Performances of different algorithms on Barreto instances

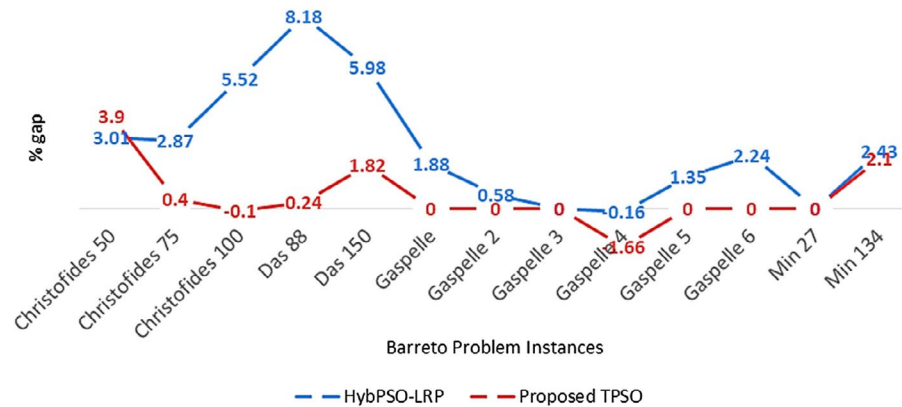
Algorithm		HybPSO-LRP ^a			Proposed tri-fold PSO		
Problems	BKS	Cost	CPU	Gap (%)	Cost	CPU	Gap (%)
Christofides 69-50x5	565.65	582.7	0.05	3.01	587.81	0.66	3.9
Christofides 69-75x10	861.6	886.3	0.37	2.87	865.28	1.2	0.4
Christofides 69-100x10	842.9	889.4	0.58	5.52	841.97*	1.15	-0.1
Daskin 95-88x8	355.8	384.9	1.11	8.18	356.66	1.75	0.24
Daskin 95-150x10	44,011.7	46,642.7	2.31	5.98	44,814.1	4.8	1.82
Gaskell 67-21x5	424.9	432.9	0.01	1.88	424.9	0.2	0.00
Gaskell 67-22x5	585.1	588.5	0.07	0.58	585.1	1.1	0.00
Gaskell 67-29x5	512.1	512.1	0.12	0.00	512.1	1.65	0.00
Gaskell 67-32x5	571.7	570.8	0.15	-0.16	562.22*	1.3	-1.66
Gaskell 67-32x5	504.3	511.1	0.16	1.35	504.3	1.3	0.00
Gaskell 67-36x5	460.4	470.7	0.15	2.24	460.4	1.4	0.00
Min 92-27x5	3,062.0	3,062.0	3.18	0.00	3,062.0	2.1	0.00
Min 92-134x8	5,809.0	5,950.1	3.10	2.43	5,931.22	3.6	2.1
Average			0.81	2.42		1.59	0.48

Notes: Boldface Cost values are better fitness value than HybPSO-LRP.

*Marked fitness values are better than the best known results in the literature.

^aResults reported in Marinakis and Marinaki (2008).

Figure 3. Performance comparison between HybPSO-LRP and proposed TPSO.



as $gap = \frac{Cost_{Algorithm} - BKS}{BKS} \times 100\%$ where, *Algorithm* can be HybPSO-LRP or TPSO. Proposed TPSO has obtained results with average gap of only 0.48% from *BKS* whereas, HybPSO-LRP has comparably higher gap of 2.42% from *BKS*. Among 13 instances, our approach has reached to the best known results for 6 instances and attained better solutions for 2 instances while it demonstrate the better performance than HybPSO-LRP on 12 instances. Comparison of difference in percentage gap between HybPSO-LRP and TPSO is portrayed in the Figure 3. Negative gap percentage represents the instances for those the proposed TPSO obtained a better result than the best known solutions reported in the literature.

For this set of instances, our proposed algorithm comparably required a higher average *CPU* time, but the average *gap* shows considerable improvement as compared to HybPSO-LRP approach.

5. Conclusions

In this article we have introduced a novel solution technique to handle the CLRP. The solution technique takes an advantage of utilizing the effects of both hierarchical and iterative approaches. While hierarchical technique or iterative technique alone has their own drawbacks in handling capacitated vehicle routing problem instance. Noteworthy to mention that they obviously have their pros which effects are tried to be hybridized in this proposed algorithm. Our proposed method overcomes the limitations of making a use of hierarchical and iterative approaches separately. Thus our proposed technique opens a new dimension for search strategy by combining the influence of hierarchical and iterative techniques in the same approach to solve CLRP instances. Such manner helps to keep the global view of the CLRP and local view to preserve optimal tracks from iterations to iterations.

For an example, we have implemented PSO to handle the CLRP instances. However, we believe that our approach can be adopted for any other population based metaheuristic algorithms to handle CLRP instances. Comparison of computational analysis has shown a promising performance in terms of obtaining solutions with lower cost for different CLRP benchmark instances.

Proposed technique can be upgraded to utilize for handling *p*-hub location-routing problem, a *p*-median type problem, hub and network design problem and so on. We look forward to explore these in our future endeavors.

Funding

This work was supported by Hankuk University of Foreign Studies Research Fund.

Author details

A.K.M. Foyzal Ahmed¹
 E-mail: foysal2011@live.com
 ORCID ID: <http://orcid.org/0000-0002-4185-4518>
 Ji Ung Sun¹
 E-mails: jusun@hufs.ac.kr, jeddy4u@gmail.com

¹ Department of Industrial and Management Engineering, Hankuk University of Foreign Studies, Global campus 81, Oedae-ro, Mohyeon-eup, Cheoin-gu, Yongin-si, Gyeonggi-do, 17035 Korea.

Citation information

Cite this article as: A novel approach to combine the hierarchical and iterative techniques for solving capacitated location-routing problem, A.K.M. Foyzal Ahmed & Ji Ung Sun, *Cogent Engineering* (2018), 5: 1463596.

References

- Ahmed, A. K. M. F., & Sun, J. U. (2016). An improved particle swarm optimization algorithm for the travelling salesman problem. *Advanced Science Letters*, 22(11), 3318–3322. <https://doi.org/10.1166/asl.2016.7864>
- Ahmed, A. K. M. F., & Sun, J. U. (2018). Bilayer local search enhanced particle swarm optimization incorporated with a distinct decoding method for solving capacitated vehicle routing problem. *Algorithms*, 11(3), 31. <https://doi.org/10.3390/a11030031>
- Ai, T. J., & Kachitvichyanukul, V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 36(5), 1693–1702. <https://doi.org/10.1016/j.cor.2008.04.003>
- Akhand, M. A. H., Akter, S., Rashid, M. A., & Yaakob, S. B. (2015). Velocity tentative PSO: An optimal velocity implementation based particle swarm optimization to solve traveling salesman problem. *Production Planning and Control*, 42(3), 221–232.
- Albareda-Sambola, M., Díaz, J. A., & Fernández, E. (2005). A compact model and tight bounds for a combined location-routing problem. *Computers & Operations Research*, 32(3), 407–428. [https://doi.org/10.1016/S0305-0548\(03\)00245-4](https://doi.org/10.1016/S0305-0548(03)00245-4)
- Alumur, S., & Kara, B. Y. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1), 1–21. <https://doi.org/10.1016/j.ejor.2007.06.008>
- Banks, A., Vincent, J., & Anyakoha, C. (2007). A review of particle swarm optimization. Part I: Background and development. *Natural Computing*, 6(4), 467–484. <https://doi.org/10.1007/s11047-007-9049-5>
- Banks, A., Vincent, J., & Anyakoha, C. (2008). A review of particle swarm optimization. Part II: Hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. *Natural Computing*, 7(1), 109–124. <https://doi.org/10.1007/s11047-007-9050-z>
- Barreto, S., Ferreira, C., Paixao, J., & Santos, B. S. (2007). Using clustering analysis in a capacitated location-routing problem. *European Journal of Operational Research*, 179(3), 968–977. <https://doi.org/10.1016/j.ejor.2005.06.074>
- Billionnet, A., Elloumi, S., & Djerbi, L. G. (2005). Designing radio-mobile access networks based on synchronous digital hierarchy rings. *Computers & Operations Research*, 32(2), 379–394. [https://doi.org/10.1016/S0305-0548\(03\)00242-9](https://doi.org/10.1016/S0305-0548(03)00242-9)
- Bouhafs, L., Hajjam, A. and Koukam, A. (2006). A combination of simulated annealing and ant colony system for the capacitated location-routing problem. In *Knowledge-Based Intelligent Information and Engineering Systems Springer Berlin/Heidelberg*, pp. 409–416.
- Branco, I. M., & Coelho, J. D. (1990). The hamiltonian p -median problem. *European Journal of Operational Research*, 47(1), 86–95. [https://doi.org/10.1016/0377-2217\(90\)90092-P](https://doi.org/10.1016/0377-2217(90)90092-P)
- Bruns, A., Klose, A., & Stähly, P. (2000). Restructuring of Swiss parcel delivery services. *OR Spectrum*, 22(2), 285–302. <https://doi.org/10.1007/s002910050106>
- Chan, A. W., & Hearn, D. W. (1977). A rectilinear distance round-trip location problem. *Transportation Science*, 11(2), 107–123. <https://doi.org/10.1287/trsc.11.2.107>
- Derbel, H., Jarboui, B., Hanafi, S., & Chabchoub, H. (2012). Genetic algorithm with iterated local search for solving a location-routing problem. *Expert Systems with Applications*, 39(3), 2865–2871. <https://doi.org/10.1016/j.eswa.2011.08.146>
- Drexl, M., & Schneider, M. (2015). A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research*, 241(2), 283–308. <https://doi.org/10.1016/j.ejor.2014.08.030>
- Drezner, Z. (1982). Fast algorithms for the round trip location problem. *Proceedings of the 6th IIE Transactions*, 14(4), 243–248.
- Drezner, Z., & Wesolowsky, G. O. (1982). A trajectory approach to the round-trip location problem. *Transportation Science*, 16(1), 56–66. <https://doi.org/10.1287/trsc.16.1.56>
- Duhamel, C., Lacomme, P., Prins, C., & Prodhon, C. (2010). A GRASP × ELS approach for the capacitated location-routing problem. *Computers & Operations Research*, 37(11), 1912–1923. <https://doi.org/10.1016/j.cor.2009.07.004>
- Eberhart, R. C., & Kennedy, J. (1995). Particle swarm optimization, proceeding of IEEE International Conference on Neural Network. In *Proceeding of IEEE International Conference on Neural Network. Perth, Australia*, pp. 1942–1948.
- Karaoglan, I., Altıparmak, F., Kara, I., & Dengiz, B. (2012). The location-routing problem with simultaneous pickup and delivery: Formulations and a heuristic approach. *Omega*, 40(4), 465–477. <https://doi.org/10.1016/j.omega.2011.09.002>
- Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *IEEE International Conference on Computational Cybernetics and Simulation: Systems, Man, and Cybernetics*, pp. 4104–4108.
- Kolen, A. (1985). the round-trip p -center and covering problem on a tree. *Transportation Science*, 19(3), 222–234. <https://doi.org/10.1287/trsc.19.3.222>
- Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3), 345–358. [https://doi.org/10.1016/0377-2217\(92\)90192-C](https://doi.org/10.1016/0377-2217(92)90192-C)
- Lin, C. K. Y., Chow, C. K., & Chen, A. (2002). A location-routing-loading problem for bill delivery services. *Journal of the Korean Institute of Management Science*, 43(1), 5–25.
- Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2), 498–516. <https://doi.org/10.1287/opre.21.2.498>
- Maranzana, F. E. (1964). On the location of supply points to minimize transport costs. *OR*, 261–270. <https://doi.org/10.2307/3007214>
- Marinakis, Y., & Marinaki, M. (2008). A particle swarm optimization algorithm with path relinking for the location routing problem. *Journal of Mathematical Modelling and Algorithms*, 7(1), 59–78. <https://doi.org/10.1007/s10852-007-9073-6>
- Melechovský, J., Prins, C., & Calvo, R. W. (2005). A metaheuristic to solve a location-routing problem with non-linear costs. *Journal of Heuristics*, 11(5–6), 375–391. <https://doi.org/10.1007/s10732-005-3601-1>
- Min, H. (1996). Consolidation terminal location-allocation and consolidated routing problems. *Journal of Business Logistics*, 17(2), 235–263.
- Nagy, G., & Salhi, S. (2007). Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177(2), 649–672. <https://doi.org/10.1016/j.ejor.2006.04.004>
- Nambiar, J. M., Gelders, L. F., & Van Wassenhove, L. N. (1981). A large scale location-allocation problem in the natural rubber industry. *European Journal of Operational Research*, 6(2), 183–189. [https://doi.org/10.1016/0377-2217\(81\)90205-8](https://doi.org/10.1016/0377-2217(81)90205-8)
- O’Kelly, M. E. 1986. The location of interacting hub facilities. *Transportation Science*, 20(2), 92–106. <https://doi.org/10.1287/trsc.20.2.92>
- Or, I., & Pierskalla, W. P. (1979). A transportation location-allocation model for regional blood banking. *AIIE Transactions*, 11(2), 86–95. <https://doi.org/10.1080/0569557908974447>

- Perl, J., & Daskin, M. S. (1985). A warehouse location-routing problem. *Transportation Research Part B: Methodological*, 19(5), 381–396.
[https://doi.org/10.1016/0191-2615\(85\)90052-9](https://doi.org/10.1016/0191-2615(85)90052-9)
- Prins, C., Prodhon, C., & Calvo, R. W. (2004). Nouveaux algorithmes pour le problème de localisation et routage sous contraintes de capacité. In *MOSIM*, pp. 1115–1122.
- Prins, C., Prodhon, C., & Calvo, R. W. (2006a). A memetic algorithm with population management (MA|PM) for the capacitated location-routing problem. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, Springer, Berlin Heidelberg, pp. 183–194.
- Prins, C., Prodhon, C., & Calvo, R. W. (2006b). Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. *4OR: A Quarterly Journal of Operations Research*, 4(3), 221–238. <https://doi.org/10.1007/s10288-006-0001-9>
- Prodhon, C. (2010, February 25). Classical instances for LRP. Retrieved December 15, 2017, from http://prodhon.free.fr/Instances/instances_us.htm
- Prodhon, C., & Prins, C. (2014). A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1), 1–17.
<https://doi.org/10.1016/j.ejor.2014.01.005>
- Salhi, S., & Fraser, M. (1996). An integrated heuristic approach for the combined location vehicle fleet mix problem. *Studies in Locational Analysis*, 8, 3–22.
- Salhi, S., & Nagy, G. (2009). Local improvement in planar facility location using vehicle routing. *Annals of Operations Research*, 167(1), 287–296.
<https://doi.org/10.1007/s10479-007-0223-z>
- Salhi, S., & Rand, G. K. (1989). The effect of ignoring routes when locating depots. *International Journal of Systems Science*, 39(2), 150–156.
- Schwardt, M., & Dethloff, J. (2005). Solving a continuous location-routing problem by use of a self-organizing map. *International Journal of Physical Distribution & Logistics Management*, 35(6), 390–408.
<https://doi.org/10.1108/09600030510611639>
- Shi, Y., & Eberhart, R. C. (1998). A modified particle swarm optimizer. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, Anchorage, AK, USA, 4–9 May, pp. 69–73.
- Sun, J. U. (2013). A two-phase meta heuristic approach to the location-routing problem for transportation network planning. *Applied Mechanics and Materials*, (Trans Tech Publications), 361, 1900–1905.
<https://doi.org/10.4028/www.scientific.net/AMM.361-363>
- Tan, Y., Gao, H. M., & Zeng, J. C. (2004). Particle swarm optimization for integer programming [J]. *Systems Engineering-theory & Practice*, 5, 126–129.
- Ting, C. J., & Chen, C. H. (2013). A multiple ant colony optimization algorithm for the capacitated location routing problem. *International Journal of Production Economics*, 141(1), 34–44.
<https://doi.org/10.1016/j.jipe.2012.06.011>
- Vincent, F. Y., Lin, S. W., Lee, W., & Ting, C. J. (2010). A simulated annealing heuristic for the capacitated location routing problem. *Computers & Industrial Engineering*, 58(2), 288–299.
- Wang, K. P., Huang, L., Zhou, C. G., & Pang, W. (2003). Particle swarm optimization for traveling salesman problem. In *IEEE International Conference on Machine Learning and Cybernetics*, pp. 1583–1585.
- Watson-Gandy, C. D. T., & Dohrn, P. J. (1973). Depot location with van salesmen—A practical approach. *Omega*, 1(3), 321–329.
[https://doi.org/10.1016/0305-0483\(73\)90108-4](https://doi.org/10.1016/0305-0483(73)90108-4)
- Wu, T. H., Low, C., & Bai, J. W. (2002). Heuristic solutions to multi-depot location-routing problems. *Computers & Operations Research*, 29(10), 1393–1415.
[https://doi.org/10.1016/S0305-0548\(01\)00038-7](https://doi.org/10.1016/S0305-0548(01)00038-7)



© 2018 The Author(s). This open access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.

You are free to:

Share — copy and redistribute the material in any medium or format
Adapt — remix, transform, and build upon the material for any purpose, even commercially.
The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made.
You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
No additional restrictions

You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.



Cogent Engineering (ISSN: 2331-1916) is published by Cogent OA, part of Taylor & Francis Group.

Publishing with Cogent OA ensures:

- Immediate, universal access to your article on publication
- High visibility and discoverability via the Cogent OA website as well as Taylor & Francis Online
- Download and citation statistics for your article
- Rapid online publication
- Input from, and dialog with, expert editors and editorial boards
- Retention of full copyright of your article
- Guaranteed legacy preservation of your article
- Discounts and waivers for authors in developing regions

Submit your manuscript to a Cogent OA journal at www.CogentOA.com

