An Improved Particle Swarm Optimization for a Class of Capacitated Vehicle Routing Problems

Hamed Alinezhad[1], Saeed Yaghubi[2], Seyyed-Mehdi Hoseini-Motlagh[3], Somayeh Allahyari[4], Mojtaba Saghafi Nia[5]

Received: 07. 12. 2016 Accepted: 16. 04. 2017

Abstract

Vehicle Routing Problem (VRP) is addressed to a class of problems for determining a set of vehicle routes, in which each vehicle departs from a given depot, serves a given set of customers, and returns back to the same depot. On the other hand, simultaneous delivery and pickup problems have drawn much attention in the past few years due to its high usage in real world cases. This study, therefore, considered a Vehicle Routing Problem with Time Windows and Simultaneous Delivery and Pickup (VRPTWSDP) and formulated it into a mixed binary integer programming. Due to the NP-hard nature of this problem, we proposed a variant of Particle Swarm Optimization (PSO) to solve VRPTWSDP. Moreover, in this paper we improve the basic PSO approach to solve the several variants of VRP including Vehicle Routing Problem with Time Windows and Simultaneous Delivery and Pickup (VRPTWSDP), Vehicle Routing Problem with Time Windows (VRPTW), Capacitated Vehicle Routing Problem (CVRP) as well as Open Vehicle Routing Problem (OVRP). In proposed algorithm, called Improved Particle Swarm Optimization (IPSO), we use some removal and insertion techniques and also combine PSO with Simulated Annealing (SA) to improve the searching ability of PSO and maintain the diversity of solutions. It is worth mentioning that these algorithms help to achieve a trade-off between exploration and exploitation abilities and converge to the global solution. Finally, for evaluating and analyzing the proposed solution algorithm, extensive computational tests on a class of popular benchmark instances, clearly show the high effectiveness of the proposed solution algorithm.

Keywords: Improved particle swarm optimization; simulated annealing; vehicle routing problem; simultaneous delivery and pickup; time windows.

Corresponding author E-mail: yaghoubi@iust.ac.ir

1 MSc. Student, School of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran.
2 Assistant Professor, School of Industrial Engineering, Iran University if Science and Technology, Tehran, Iran.
3 Assistant Professor, School of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran.
4 PhD. Student, School of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran.
5 Instructor, School of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran.

1. Introduction

The Vehicle Routing Problem (VRP) was proposed for the first time by Dantzig and Ramser [Dantzig and Ramser, 1959]. VRP is addressed to a class of problems for determining a set of vehicle routes, in which each vehicle departs from a given depot, serves a given set of customers, and returns back to the same depot. Collection of household waste, gasoline delivery, goods distribution and mail delivery are the most common applications of VRPs. Many variants of VRP may appear since there are many cases in real-world conditions, such as the number of depots, type of vehicles, and customers' requirements. Comprehensive details on VRP, its variants, formulations, and solution methods has been provided by Thoth and Vigo [Thoth and Vigo, 2002].

For more realistic applications, this paper further investigates a more general situation, called the Vehicle Routing Problem with Time Windows and Simultaneous Delivery and Pickup (VRPTWSDP). In this variant, customers require not only the delivery of goods but also the simultaneous pickup of goods from them, in which each customer should be served within a specific time period. Such an application is frequently encountered for example in the distribution system of grocery store chains, each grocery store may have a demand for both delivery (i.e. fresh food or soft drinks) and pickup (i.e. outdated items or empty bottles), and is serviced with a single stop by the supplier [Chun-Hua, Hong and Jian, 2009].

It is known that the VRP is NP-hard [Solomon, 1987]. Moreover, VRPTWSDP is one extension of VRP which contains time windows and pickup and delivery. Hence, the VRPTWSDP is also NP-hard; in this manner, it is believed that one may never find a computational technique guaranteeing an optimal solution to larger instances for such problems. Since exact algorithms are not efficient for solving NP-hard problems, several heuristics and metaheuristics such as Tabu Search (TS) [Montane and Galvao, 2006], Genetic Algorithm (GA) [Wang and Chen, 2012] as well as Particle Swarm Optimization (PSO) [Ai and Kachitvichyanukul, 2009] have been proposed for solving the VRP and its variants.

In this paper, we propose an Improved Particle Swarm Optimization (IPSO) approach for solving a class of vehicle routing problems including VRPTWSDP, VRPTW, CVRP as well as OVRP. PSO is a population-based search method proposed by Kennedy and Eberhart [Kennedy and Eberhart,1995], which is motivated by the group organism behavior such as bee swarm, fish school, and birds flock. PSO imitate the physical movement of the individuals in the swarm as a searching method. A brief and complete survey on the PSO mechanism, technique, as well as application, has been provided by Kennedy and Eberhart [Kennedy and Eberhart, 2001] and also Clerc [Clerc, 2006]. Rest of this section will review articles which have used PSO to solve the VRP variants.

[Xiao, Huang, Li and Wang, 2005] proposed a modified particle swarm optimization for solving the VRP. [Xiao, Li and Wang, 2005] also applied a method based on PSO to solve the discrete VRP. They changed the VRP into a quasi-continuous problem by designing a new real coding and solved it using PSO. [Wang, Wu, Zhao and Feng, 2006] also solved the OVRP using PSO. The authors applied several heuristic methods into the post-optimization procedure after decoding, such as Nearest Insertion algorithm, GENI algorithm, and 2-Opt to optimize the inner or outer routes and modify illegal solutions. [Zhang, Pang, Xiao and Wu, 2006] used a combination of PSO and Simulated Annealing (SA) for solving the VRP which can avoid being trapped in a local optimum using probability search. A new hybrid approximation algorithm to solve CVRP was introduced by [Chen, yang and Wu, 2006], in which discrete particle swarm optimization combines global search and local search to search the optimal results and SA uses certain probability to avoid being trapped in a local optimum. [Zhu, Qian, Li and Zhu, 2006] have combined local search methods with global search methods, attempting to balance both exploration and exploitation as well as have proposed an improved PSO algorithm for solving the VRPTW. [Xu and Huang, 2007] have solved the VRP with multiple objectives

by combining PSO with mutation operator and with the help of roulette-wheel. [Marinakis and Marinaki, 2008] have proposed a new hybrid algorithmic nature inspired approach based on PSO, for solving the location routing problem.
Two solution representations and the corresponding decoding methods for solving the CVRP using PSO also have presented by [Ai and Kachitvichyanukul, 2009], in which the first representation constructs vehicle routes based on the customer priority list and vehicle priority matrix, and the second representation constructs vehicle routes based on vehicle orientation points and the vehicle coverage radius. [Ai and Kachitvichyanukul, 2009] proposed a formulation for the VRPSPD and developed a PSO algorithm with multiple social structures for solving it. The introduced decoding method starts by transforming the particle to a priority list of customers to enter the route and a priority matrix of vehicles to serve each customer. The vehicle routes are constructed based on the customer priority list and vehicle priority matrix. A heuristic based on PSO algorithm for solving VRPTW which is an extension of PSO application for the CVRP was also introduced by [Ai and Kachitvichyanukul, 2009].
[Castro, Landa-Silva and Pérez, 2009] investigated the ability of a discrete PSO algorithm to evolve solutions from infeasibility to feasibility for the VRPTW. The proposed algorithm incorporates some principles of multi-objective optimization to allow particles to conduct a dynamic trade-off between objectives in order to reach feasibility. [Yannis and Magdalene, 2010] employed a hybrid genetic – PSO for the VRP in which the evolution of each individual of the total population, which consists of the parents and the offspring, is realized with the use of a particle swarm optimizer. During evolution process, each individual of the total population has to improve its physical movement following the basic principles of PSO until it will obtain the requirements to be selected as a parent. The authors [Yannis and Magdalene, 2010] also proposed a hybrid PSO which combines a PSO algorithm, the multiple phase neighborhood search algorithm, greedy randomized adaptive search procedure algorithm, the expanding neighborhood search strategy and a path relinking strategy together for solving the VRP.
[Gong, Zhang, Liu, Huang and Chung, 2012] solved the VRPTW using a set-based PSO algorithm. [Tian, Ma, Wang Y. L. and Wang, K. L. 2011] modeled an emergency supplier with fuzzy demands, dynamic transportation network, and prioritized supplying and then designed a PSO algorithm according to the characteristics of the model for solving it. [MirHassani and Abolghasemi, 2011] implemented PSO for solving the OVRP in which a vehicle does not return to the depot after servicing the last customer on a route. [Moghaddam, Ruiz and Sadjadi, 2012] introduced an advanced PSO algorithm for solving an uncertain VRP in which the customers' demand is supposed to be uncertain with unknown distribution. [Kim and Son, 2012] proposed a probability matrix based hybrid PSO algorithm for the CVRP in which the developed PSO approach uses a probability matrix as the main device for particle encoding and decoding. The proposed algorithm assigns customers to routes and determines a sequence of customers simultaneously. [Marinakis, Iordanidou and Marinaki, 2013] also introduced a new hybrid based on PSO for solving the VRP with stochastic demands. [Goksal, Karaoglan and Altiparmak, 2013] solved VRPSPD using PSO algorithm in which a local search is performed by variable neighborhood descent algorithm, moreover they implemented an annealing-like strategy to preserve the swarm diversity. [Belmecheri, Christian, Farouk and Lionel, 2013] also proposed a PSO with a local search for solving a complex VRP called "particle swarm optimization algorithm for a vehicle routing problem with a heterogeneous fleet, mixed backhauls and time windows".
[Mokhtarimousavi, Rahami, Saffarzadeh and Piri, 2014] modeled aircraft landing scheduling problem (ASLP) and solved it by a multi-objective genetic algorithm (NSGA-II) and multi-objective particle swarm optimization algorithm (MOPSO). [Norouzi, Sadegh-Amalnick and Alinaghiyan, 2015] presented a new mathematical model for measuring and evaluating the efficiency of periodic vehicle

routing problem (PVRP) and solved it by using IPSO and Original PSO. Their computational results show that the improved PSO algorithm performs well in terms of accuracy but the original PSO performs better in computational time. [Marinakis, 2015] also presented a new version of the particle swarm optimization (PSO) algorithm suitable for discrete optimization problems and applied it for the solution of the capacitated location routing problem and for the solution of a new formulation of the location routing problem with stochastic demands. [Cheng, Chen Y. Y., Chen T. L. and Yoo, 2015] implemented an efficient hybrid algorithm for solving the joint batch picking and picker routing problem which the core of the hybrid algorithm is composed of the PSO and the ant colony optimization (ACO) algorithms. A VRP that simultaneously considers production and pollution routing problems with time window (PPRP-TW) is considered by [Kumar, Kondaraneni, Dixit, Goswami and Thakur, 2016] they used a hybrid Self-Learning Particle Swarm Optimization (SLPSO) algorithm in the multi-objective framework to solve the problem. [Chen, Hsiao, Reddy and Tiwari, 2016] attempted to address the VRP of distribution centers with multiple cross-docks for processing multiple products. Due to the high complexity of the model, they solved it by using a variant of Particle Swarm Optimization (PSO) with a Self-Learning strategy, namely SLPSO. With respect to both the literature reviewed above as well as the best of our knowledge, researchers do not take PSO algorithm into account to solve the VRPTWSDP.

One of the main contributions of this paper is to show that we can combine PSO and other algorithms to obtain solutions for VRP problems with remarkable results from both of quality and computational efficiency point of view. In other words, a new combination of PSO with some removal and insertion and SA algorithm is given to improve the quality of PSO results and avoid being trapped in local optima. The second contribution of this paper is the usage of presented algorithm for solving four different kinds of VRP problems and solving instances utilized in the literature. The rest of this paper will be organized as follows. Since VRPTW, CVRP and OVRP are special cases of VRPTWSDP, section 2 reviews the VRPTWSDP definition and mathematical formulation. Section 3 also illustrates the developed IPSO algorithm for solving VRPTWSDP. Section 4 discusses the computational experiment of the proposed IPSO on the benchmark instances of a class of VRP variants. Finally, Section 5 concludes the result of the research.

2. Problem Description and Mathematical Formulation

Vehicle routing issue can be described as follows: a homogeneous fleet of vehicles has to visit a number of customers with deterministic demands which are located in various cities. Furthermore, each customer can be visited at a specific time interval, called time window. The problem concerned in this paper, called VRPTWSDP, is how to send out a fleet of capacitated vehicles from a distribution center to meet the customers' request (simultaneous delivery and pickup) with the minimum distance traveled in such a way that:

a) All routes start at the depot and end at the same depot;
b) Each customer is visited exactly once within its time window;
c) The total of customers' demand for each route cannot exceed the vehicle capacity.

In VRPTWSDP we are given a directed graph $G = (V, A)$ in which $V = \{0, 1, \dots, n, n+1\}$ is the set of vertices, and $A = \{(i,j) | i, j \in V\}$ is the set of arcs. More precisely $V' = \{1,2,\dots,n\}$ represents the set of customer vertices where each $i \in V'$ has a pre-specified delivery and pickup demand respectively shown by $r_i > 0$ and $p_i > 0$, to be met by exactly one vehicle. Moreover, indices 0 and $n+1$, are related to the depots which are consequently the start and end node of each route. Each arc $(i,j) \in A$ is associated with a non-negative routing distance $d_{i,j}$ and travel time $t_{i,j}$. We assume that a limited set $K = \{1,2,\dots,r\}$ of capacitated vehicles is available, where each vehicle $k \in K$ has capacity C_k. We also assume that $[e_i, l_i]$ is time window interval in which each customer $i \in V'$

should be visited and s_i is the service time of customer $i \in V'$. Based on the above problem description, decision variables have been introduced below.

$U_{i,k}$	Pickup load of vehicle $k \in K$ after serving customer $i \in V'$.
$V_{i,k}$	Delivery load of vehicle $k \in K$ before serving customer $i \in V'$.
$X_{i,j,k}$	If vehicle $k \in K$ travels directly from node $i \in V$ to node $j \in V$, then $X_{i,j,k} = 1$; otherwise $X_{i,j,k} = 0$.
$ST_{i,k}$	Arrival time of vehicle $k \in K$ to node $i \in V$.

Finally, the mathematical formulation for VRPTWSDP is given here, in which M is an arbitrary large constant.

The objective function (1) minimizes the total traveling distance subject to vehicle capacity, travel time and arrival time as well as the other feasibility constraints, explained as follows. Each customer can be visited at most once (2) and served by one and only one vehicle (3). Vehicle capacity and sub-tour elimination constrains which due to polynomial complexity has a considerable effect on reducing the computational time are (4)-(10). These constraints are adaptions of those applied by [Karaoglan, Altiparmak, Kara and Dengiz, 2012]. Constraints (11) and (12) make sure that depot is the start and end point of all routes which means each vehicle should leave node 0 and enter node n+1. Constraints (13), (14) and (15) ensure feasibility of the time schedule. Constraints (16) and (17) denote that no vehicle enters to node 0 and leaves from node $n+1$, consequently. Finally, constraints (18) and (19) determine binary variables and the sign of variables.

$$\text{Min} \sum_{i\in V}\sum_{j\in V}\sum_{k\in K} d_{i,j}X_{i,j,k} \tag{1}$$

subject to:

$$\sum_{j\in V}\sum_{k\in K} X_{i,j,k} = 1; \qquad \forall i \in V' \tag{2}$$

$$\sum_{j\in V} X_{j,i,k} - \sum_{j\in v} X_{i,j,k} = 0; \qquad \forall i \in V', k \in K \tag{3}$$

$$U_{j,k} - U_{i,k} + C_k X_{i,j,k} + (C_k - r_i - r_j)X_{j,i,k} \le C_k - r_i; \quad \forall i,j \in V', i \ne j, k \in K \tag{4}$$

$$V_{i,k} - V_{j,k} + C_k X_{i,j,k} + (C_k - p_i - p_j)X_{j,i,k} \le C_k - p_j; \quad \forall i,j \in V', i \ne j, k \in K \tag{5}$$

$$U_{i,k} + V_{i,k} - r_i \le C_k; \qquad \forall i \in V', k \in K \tag{6}$$

$$U_{i,k} \ge r_i + \sum_{\substack{j\in V' \\ (j\ne i)}} r_j X_{i,j,k}; \qquad \forall i \in V', k \in K \tag{7}$$

$$V_{i,k} \ge p_i + \sum_{\substack{j\in V' \\ (j\ne i)}} p_j X_{j,i,k}; \qquad \forall i \in V', k \in K \tag{8}$$

$$U_{i,k} \le C_k - (C_k - r_i)X_{i,n+1,k}; \qquad \forall i \in V', k \in K \tag{9}$$

$$V_{i,k} \le C_k - (C_k - p_i)X_{0,i,k}; \qquad \forall i \in V', k \in K \tag{10}$$

$$\sum_{i\in V} X_{0,i,k} = 1; \qquad \forall k \in K \tag{11}$$

$$\sum_{i\in V} X_{i,n+1,k} = 1; \qquad \forall k \in K \tag{12}$$

$$ST_{j,k} \geq ST_{i,k} + s_i + t_{i,j} - M(1 - X_{i,j,k}); \qquad \forall i, j \in V, i \neq j \tag{13}$$

$$ST_{i,k} \geq e_{i;} \qquad \forall i \in V, k \in K \tag{14}$$

$$ST_{i,k} \leq l_i; \qquad \forall i \in V, k \in K \tag{15}$$

$$\sum_{i \in V} \sum_{k \in K} X_{i,0,k} = 0 \tag{16}$$

$$\sum_{i \in V} \sum_{k \in K} X_{n+1,i,k} = 0 \tag{17}$$

$$X_{i,j,k} \in \{0,1\}; \qquad \forall i, j \in V, k \in K \tag{18}$$

$$U_{i,k}, V_{i,k}, ST_{i,k}, Z \geq 0; \qquad \forall i \in V, k \in K \tag{19}$$

3. An Improved Particle Swarm Optimization for VRPTWSDP

3.1 Basic PSO

PSO has been widely used mainly due to its simple concept, effectiveness and its ability to find a reasonable solution fast. Since it is easy to be trapped into local optima while optimizing complex global optimization problems, it is not always efficient [Liang, Qin, Suganthan and Baskar, 2006]. The PSO is designed for global optimization by emulating the behavior of animals' societies that do not have any leader in their group or swarm, such as bird flocking and fish schooling [Chen and Ye, 2004]. The process of PSO algorithm in finding optimal values follows the work of this animal society.

PSO is a population-based stochastic algorithm that starts with an initial population of randomly generated particles. In the PSO, each solution to a particular problem is called particle and the population of solutions is called swarm [Figureueiredo, Ludermir and Bastos-Filho, 2016]. Each particle has two properties of position and velocity for a search problem in a $|D|$-dimensional space where $D = \{1, 2, \dots, d\}$ is the set of dimensions. A particle $i \in V'$ in PSO represents a solution $X_i = [x_i^1, x_i^2, \dots, x_i^d]$ which is associated with a velocity vector $V_i = [v_i^1, v_i^2, \dots, v_i^d]$, where $d \in D$. (20) is used to calculate the particle's new velocity according to particle's immediate previous velocity, the distances of its current position from its own best experience (position) and the group's best experience. Then the particle flies to a new position according to (21).

where $pbest_i = (pbest_i^1, pbest_i^2, \dots, pbest_i^D)$ is the best previous position yielding the best fitness value for the ith particle in $|D|$-dimensional space and $gbest = (gbest_1, gbest_2, \dots, gbest_D)$ is the global best particle found by all particles so far. Both $rand1_i^d$ and $rand2_i^d$ are two uniform random numbers generated independently within the range of $[0, 1]$ and stochastic exploration nature of PSO is due to these random numbers, c_1 and c_2 are two learning factors which represent the particle confidence in itself and swarm. The parameter w, called inertia weight, which is used to balance the global and local search abilities of particles and is also to balance the exploration and the exploitation abilities, a linearly decreasing w over the search process is a good choice [Shi and Eberhart, 1998]. In order to reduce this weight over the iterations, it is updated according to the following equation (22); where w_{max} and w_{min} are the maximum and minimum values that the inertia weight can take, t_{max} is the maximum number of iterations and t is the current iteration. Some other strategies such as using an inertia weight with a random component can also be used [Poli, Kennedy and Blackwell, 2007].

The velocities of particles are limited in interval $[V_{min}, V_{max}]$. The equations (23) and (24) are used to initialize the max and min velocity in the dth dimension to the solution, i.e., $V_{max,d}$ and $V_{min,d}$ respectively: Where $X_{max,d}$ and $X_{min,d}$ are the minimum and maximum positions of the particle in the dth dimension and ψ is a constant factor taken from [0,1]. If the resulting value for velocity is smaller than V_{min}, velocity vector is set to V_{min} and if the resulting value is greater than V_{max}, it is set to V_{max}. After updating velocity, the performance of each particle is measured according to the fitness function. The PSO algorithm will be terminated after a maximum number of iterations or when it achieves a maximum CPU time [Han, Zhang, Hu and Lu, 2016].

$$v_i^d(t+1) = w.v_i^d(t) + c_1.\text{rand}1_i^d.\left(\text{pbest}_i^d(t) - x_i^d(t)\right) + c_2.\text{rand}2_i^d.\left(\text{gbest}_d(t) - x_i^d(t)\right) \quad (20)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (21)$$

$$w_t = w_{max} - ((w_{max} - w_{min})/t_{max})t \quad (22)$$

$$V_{max,d} = \psi(X_{max,d} - X_{min,d}) \quad (23)$$

$$V_{min,d} = \psi(X_{min,d} - X_{max,d}) \quad (24)$$

3.2 Particle Definition and Solution Expression

Regarding the mentioned description of the particle behavior in PSO searching a continuous space as well as the discrete nature of the VRP, we define a suitable mapping between VRPTWSDP solution and particles in PSO. Each particle is recorded via the path representation of each route which is the specific sequence of nodes. In this paper, we use a special particle coding for VRPTWSDP problem which helps to convert discrete combinational problem to continuous problem so that the PSO algorithm can be directly applied. We assume a VRPTWSDP problem with $|V'|$ customers and $|K|$ available vehicles. A route sequence for the whole problem can be defined as a $|V'| + |K| - 1$ dimensional permutation. For example, in a problem with three vehicles and nine customers, a possible route sequence can be shown as Figureure1. The numbers which are greater than $|V'|$ (i.e. delimiters) separate the individual routes in the sequence.

5	6	9	10	1	8	3	11	4	2	7

Figure 1. Sequence of routes for the VRPTWSDP

There is one situation that should be pointed out: if two numbers larger than $|V'|$ are neighbors, it means that there exists a vehicle which has not been assigned any delivery job. The above-mentioned representation should be transformed appropriately, therefore we turn each element of the solution into a floating point between $[0, 1]$. Thus, we divide each element of the solution by the vector's largest element. More precisely, the previous example becomes as follows:

0.45	0.55	0.82	0.91	0.09	0.73	0.27	1	0.36	0.18	0.64

Figure 2. Transformed solution into continuous vector

After calculating the velocity of all particles using (20), the elements of the velocity vector are transformed back into the integer domain using relative position indexing (Lichtblau, 2002). In this way, the smallest floating value is assigned to the smallest integer, the next highest floating value to the next integer and so on.

With this definition, each particle can represent a route for the VRPTWSDP. Furthermore, in this paper, some of the constraints are automatically satisfied. Because of a direct relation between the number of constraints and problem solving duration, the vehicle capacity and the time window constraints are added to the objective function. To do this, suppose that the relation between g_0 and g is $g \leq g_0$. The amount of violation is calculated as $\text{Violation} = \begin{cases} 0 & ; g \leq g_0 \\ \frac{g-g_0}{g_0} & ; g \geq g_0 \end{cases}$.

So, general formula for Violation amount will be as $\text{Violation} = \text{Max}\left\{0, \frac{g}{g_0} - 1\right\}$.

The vehicle capacity and the time window constraints are added to the objective function as follows using two large positive numbers R and S as the penalty coefficients to deal with these constraints. During PSO algorithm iterations, the infeasible solutions would attain very large fitness values and the particles will move to the feasible solutions.

The detail of the basic PSO algorithm for the VRPTWSDP problem is explained below:

Step 1, Initialization: The initial solution is defined by a permutation with $|V'| + |K| - 1$ elements, then the routes are determined by delimiters.

Step 2, Evaluation: each element of the solution is transformed into a floating point between $[0, 1]$ and every particle is evaluated according to (25) as well as both gbest and pbest_i (for each particle $i \in V'$) are saved.

Step 3, Velocity and Position Update: Each particle's position is updated according to (21) and then is transformed back into the integer domain.

Step 4, Judgment: If the termination condition is not met, the algorithm continues with step 2.

$$\text{Min} \sum_{i \epsilon V} \sum_{j \epsilon V} \sum_{k \epsilon K} d_{i,j} X_{i,j} + R \sum_{k \epsilon K} \text{Max}\left\{\left(\left(\sum_{i,j \epsilon V'} r_i X_{i,j,k}\right)/C_k\right) - 1, 0\right\} + S \sum_{i \epsilon V'} \text{Max}\left\{1 - \left(\frac{ST_{i,k}}{e_i}\right), 0, \left(\frac{ST_{i,k}}{l_i}\right) - 1\right\} \quad (25)$$

3.3 Improvement of PSO

Generally, disadvantages of stochastic search algorithms include premature convergence due to the quick losing of diversity. The evolutionary mechanism of PSO can bring in a much faster convergence speed once the local optimal position has been found, because of regarding the global best particle as the optimal particle and all the other particles learn from it [Zhang, 2012]. If the problems are complex with many local optima, the traditional algorithms can hardly escape from them. In order to avoid the situations described above, in this paper, we propose IPSO, in which we use some algorithms to improve the searching ability of PSO and maintain the diversity of solutions. These algorithms help to achieve a trade-off between exploration and exploitation abilities and converge to the global. Proposed techniques are expected to get good results for solving complex problems.

3.3.1 Improvement by Removal and Insertion

Sometimes, the sub-optima are near to the global optimum and the neighborhoods of trapped individuals may contain the global optimum. In such a situation, searching the neighborhoods of individuals is helpful to find better solutions [Wang, Sun, Li, Rahnamayan and Pan, 2013]. Based on this idea, we introduce several local searches that have been applied to some nature-inspired algorithms to escape from local optima and maintain the diversity of solutions. Techniques used here consist of a removal and an insertion. In the other words, in each iteration, some customers are removed from their positions and then inserted at new positions. The local searches applied in this research are divided into two groups. One of both groups is chosen randomly and then one of the selected groups' operations is executed at random. This process is iterated as long as we reach the maximum number of iterations. The procedures of two groups are described in the following:

The first group based on randomization, consists of four stochastic local search operators, namely swap operator, twice tour swap operator, reversion operator and insertion operator. A brief illustration of these operators is given below for completeness and also shown in Figure 3.

- **Swap operator:** swap operator selects two customers (nodes) randomly in a route sequence and changes the location of visiting them, see [Wang, Huang, Zhou and Pang, 2003].

- **Twice tour swap operator:** it is the same as swap operator but selects two range of customers instead of selecting two customers. Each range should contain at least one customer and at most $|V'| - 2$ customers, see [Wang, Huang, Zhou and Pang, 2003].
- **Reversion operator:** Reversion operator also selects two random customers among route sequence and changes their location. Furthermore, it also reverses the order of customers between the two selected nodes, see [Wu, Liang, Lee and Lu, 2004].
- **Insertion operator:** Insertion operator also selects two random nodes through route sequence, regarding the order' importance of selected numbers here. This operator shifts the first selected number to the position after the second selected number, see [Mester, Bräyay and Dullaert, 2007].

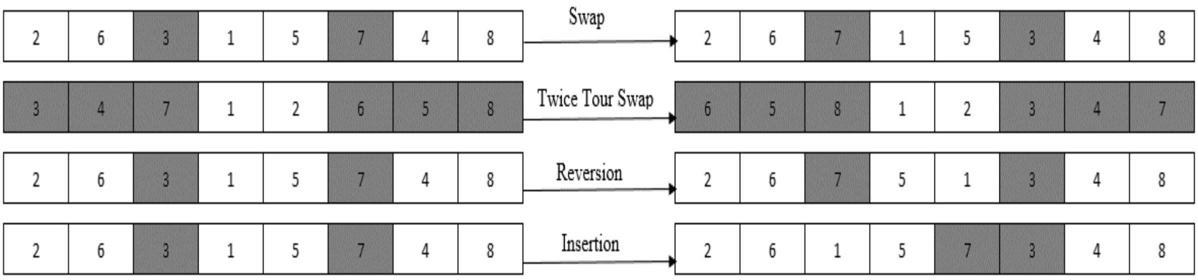

Figure 3. Performance of stochastic neighborhood searching operators

In the second group, we introduce some intelligent removal and insertion techniques for generating neighborhood solutions. Applying these methods demonstrated below, the quality of solutions will be better due to intelligent removal and insertion.

- **Worse-Distance Removal (WDR):** WDR removes the customer with the largest cost (dr^*) in each iteration [Cho, Cheung, Edwards and Fung, 2003], where the cost of each customer is defined as distance from before ($i \in V'$) and next ($h \in V'$) customer/node at the same route. In the other words, the removed customer is selected according to (26).

$$dr^* = \arg\max_{j \in V'} \{|d_{ij} + d_{jh}|\} \quad (26)$$

- **Worse-Time Removal (WTR):** This removal operator removes the customer with the largest deviation (tr^*) from the vehicle arrival time to customer position (ST_j) and start of customer time window (e_j) [Planeta, 2007]. More precisely, the customer for this removal strategy is chosen according to (27).

$$tr^* = \arg\max_{j \in V'} \{|ST_j - e_j|\} \quad (27)$$

- **Greedy Insertion (GI):** GI inserts the removed customer ($j \in V'$) at the best feasible position, see [Zhang, Jue and Mukherjee, 2000]. For inserting j between all combinations of two customers $i \in V'$ and $h \in V'$, this operator calculates equation (28), and then selects position with gi^* cost.

$$gi^* = \arg\min_{j \in V'} \{d_{ij} + d_{jh} - d_{ih}\} \quad (28)$$

- **Best Time Insertion (BTI):** Because of the time window constraints, applying this insertion operator is valuable and helps the algorithm to find the feasible solutions. BTI computes (29) for inserting customer $j \in V'$, and finally selects the position with minimum quantity (ti^*) [Diana and Dessouky, 2004].

$$ti^* = \arg\min_{j \in V'} \{|ST_j - e_j|\} \quad (29)$$

3.3.2 Hybrid with SA

SA is an approximate local search meta-heuristic, described by Kirkpatrick et al. (1984), which adapts Metropolis-Hastings algorithm [Metropolice, Rosenbluth A. W., Rosenbluth M. N. and Teller, 1953]. SA searches the neighborhoods of current solution and creates a new solution by random [Van LAarhoven and Aarts, 1987]. The key feature of SA is that it does not search for the best solution in the neighborhood of the current solution. The new solution is accepted or rejected depending on its relative cost. An improved or unchanged solution is always accepted and a fraction of inferior solutions may also be accepted in the hope of escaping from local optima [Naderi, Zandieh, Balagh and Roshanaei, 2009]. Temperature is used to imitate the cooling process and is a parameter for controlling the performance of the algorithm. The probability of accepting inferior solutions depends on the current temperature. If a new solution is inferior, it is accepted with the probability of $e^{-\Delta E/T}$ in which ΔE is the difference between the value of new objective function and best-found solution and T is the temperature [Romero, Gallego and Monticelli, 1996]. To help PSO escape from local optimums, we determine an initial temperature which decreases in each iteration by a specific ratio and let inferior solutions to be accepted with the mentioned probability. A brief review of IPSO steps is shown in Table 1.

4. Computational Results

To evaluate the performance of proposed algorithm, we report extensive computational tests on the benchmark instances available in the literature for the VRPTWSDP, VRPTW, CVRP and OVRP. Section 4.1 provides parameter tuning of the proposed algorithm. Section 4.2 compares the performance of IPSO with several heuristics on VRPTWSDP. In section 4.3, the results of the developed algorithm are compared with some heuristics on VRPTW. Finally, the results of CVRP and OVRP benchmark instances are also given in section 4.4 and 4.5, respectively. The IPSO has been implemented in MATLAB 2013 environment and all experiments were executed on an Intel Core i5 with 2.5 gigahertz and 6 gigabytes of RAM running under Windows 8-64 bits system.

The efficiency of the proposed algorithm is discussed with respect to the quality of results. The quality is due to the deviation from best-known solutions. In the following tables, Input denotes the test problem and Gap is calculated by $100 \times (\mathrm{Cost}_{\mathrm{IPSO}} - \mathrm{C}_{\mathrm{best}})/\mathrm{C}_{\mathrm{best}}$, where $\mathrm{Cost}_{\mathrm{IPSO}}$ is the cost of the solution found by IPSO and $\mathrm{C}_{\mathrm{best}}$ denotes the cost of best-found solution in the literature. The other abbreviations used in this section are $|V'|$, $|K|$ and Cost, which are the number of customers, the number of vehicles and total distance for each instance, consequently. In the following, the obtained results are explained in details.

4.1 Parameters Tuning

The result of IPSO algorithm is influenced by a number of control parameters, the number of particles (swarm size, i.e. $|N|$), the acceleration coefficients (c_1 and c_2), the inertia weight (w), the number of iterations (t_{max}), the initial temperature (T_0) and the temperature reduction factor (α). According to the considerable effect of parameter adjusting on the results of the proposed algorithm, we have used Taguchi design for tuning the algorithm parameters by considering five levels for each parameter value. The selected parameters are given in Table 2.

4.2 Comparison of IPSO with Other Heuristics on VRPTWSDP Instances

We have selected a set of fifteen instances of VRPTWSDP that have been utilized by [Wang and Chen, 2012] and have divided them into two groups, small-size and large-size instances. For small-size instances (less than 25 nodes), we also have reported exact solutions. For each small size instance, CPLEX is run with a time limit of 7200 seconds. The results are given in Table 3 and Table 4.

Computational results indicate that our algorithm has found the best-found solutions for all small-size instances and for large-size instances, our method has reached good results and the gaps are acceptable.

Table 1. The frame work of the IPSO

Initialization
Generate the initial population of the particles
Evaluate the fitness Function of each particle
Keep Optimum solution of each particle
Keep Optimum particle of the whole swarm
Main Loop
Do until the maximum number of iterations has not been reached:
    Determine the velocity of each particle
    Apply velocity limits
    Determine the position of each particle
    Apply position limits
    Evaluate the new fitness function of each particle
    Update the optimum solution of each particle by accepting a fraction of inferior solutions with probability of $e^{-\Delta E/T}$
    Update the optimum particle
    Apply removal and insertion operators
    Evaluate the new fitness function of each particle
    Update the optimum solution of each particle by accepting a fraction of inferior solutions with probability of $e^{-\Delta E/T}$
    Update the optimum particle
    Update the inertia weight and temperature
End do
Return the best particle (the best solution)

Table 2. Parameters setting

Parameter	Value
$\|N\|$	45
c_1	1.99
c_2	2
w_{max}	0.93
w_{min}	0.35
t_{max}	1800
T_0	1250
α	0.99

Table 3. Cost comparison of IPSO results with small-size benchmark instances on VRPTWSDP

Input	$\|V'\|$	$\|K\|$	C_{best}	Cplex	Wang & Chen (2012)	IPSO	Gap (%)
RCdp1001	10	3	**348.98**	**348.98**	**348.98**	**348.98**	0.00
RCdp1004	10	2	**216.69**	**216.69**	**216.69**	**216.69**	0.00
RCdp1007	10	2	**310.81**	**310.81**	**310.81**	**310.81**	0.00
RCdp2501	25	5	**551.05**	862.14	**551.05**	**551.05**	0.00
RCdp2504	25	4	**473.46**	746.23	**473.46**	**473.46**	0.00
RCdp2507	25	5	**540.87**	680.20	**540.87**	**540.87**	0.00
Average			406.98	527.51	406.98	406.98	0.00

Table 4. Cost comparison of IPSO results with large-size benchmark instances on VRPTWSDP

Input	$\|V'\|$	$\|K\|$	C_{best}	Wang & Chen (2012)	IPSO	Gap (%)
RCdp5001	50	9	**994.18**	**994.18**	998.56	0.44
RCdp5004	50	6	**725.59**	**725.59**	**725.59**	0.00

RCdp5007	50	7	**809.72**	**809.72**	**809.72**	0.00
Rdp201	100	4	**1280.44**	**1280.44**	**1280.44**	0.00
Rdp204	100	3	**775.23**	**775.23**	**775.23**	0.00
Cdp201	100	3	**591.56**	**591.56**	594.63	0.52
Cdp204	100	3	**590.60**	**590.60**	590.90	0.05
RCdp201	100	4	**1587.92**	**1587.92**	1589.99	0.13
RCdp204	100	3	**822.02**	**822.02**	**822.02**	0.00
Average			908.58	908.58	909.67	0.13

4.3 Comparison of IPSO with Other Heuristics on VRPTW Instances

In this section, we test the performance of proposed algorithm over fifteen small-size and large-size instances. Table 5 reports the summary of solutions obtained by the IPSO in VRPTW benchmark instances provided by [Solomon and Desrosiers, 1988]. IPSO was successful to equal the best cost in twelve of fifteen instances and the average gap between the solutions found by IPSO and the best-found solution is 0.06 percent. According to obtained results, we can conclude that the proposed method can result in good solutions. Moreover, Table 5 reports the summary of solutions obtained by the IPSO in VRPTW benchmark instances provided by [Küçükoğlu and Öztürk, 2014].

4.4 Comparison of IPSO with Other Heuristics on CVRP Instances

To test the performance of our method, we have selected seven instances proposed by [Christofides, Mingozzi and Toth, 1979] for the CVRP. The detailed results are provided in Table 6. The algorithm has found best-known solutions for five instances and for other two instances, the gap is 0.19 percent and 0.49 percent. With due attention to the average performance of the proposed algorithm, it is obvious that the proposed algorithm is effective with the average gap 0.1 percent from the best-known solutions.

4.5 Comparison of IPSO with Other Heuristics on OVRP Instances

Table 6 lists the running results of IPSO over 7 instances taken from [Christofides, Mingozzi and Toth, 1979] in OVRP. In this table, we compare the result of IPSO with three state-of-the-art heuristics available in the literature. The reported results in Table 7, show that our algorithm has found all best-known solutions except in one case with the gap 0.16 percent. Due to average gap 0.02 percent, we can conclude the efficiency of proposed algorithm.

5. Conclusion

Simultaneous delivery and pickup problems have drawn much attention in the past few years due to its high usage in real world cases. Furthermore, customers request specific service time. This study, therefore, considered a vehicle routing problem with simultaneous delivery and pickup with time windows and formulated it into a mixed binary integer programming model denoted by VRPTWSDP.

Due to the NP-hard nature of the problem, in this paper, we proposed a variant of PSO to solve VRPTWSDP. To avoid being trapped into local optima and maintain diversity, we used some removal and insertion algorithms and also combined it with SA to make the quality of results better. Finally, the proposed IPSO tested on VRPTWSDP, VRPTW, CVRP and OVRP benchmark instances in different sizes and produced very satisfactory results. For future research, we can apply this algorithm in other variants of the classic vehicle routing problem or solve the problem by other algorithms and compare the solutions obtained from different algorithms. We also can apply the proposed algorithm for stochastic vehicle routing problem and consider some parameters in a stochastic environment.

Table 5. The cost comparison of IPSO with other heuristics on VRPTW benchmark instances

Input	$\lvert V' \rvert$	$\lvert K \rvert$	C_{best}	Küçükoğlu and Öztürk (2014)	Ursani et al. (2011)	IPSO	Gap (%)
R101.25	25	8	**617.10**	**617.10**	**617.10**	**617.10**	0.00
R101.50	50	12	**1044.00**	**1044.00**	**1044.00**	**1044.00**	0.00
R101.100	100	20	**1637.70**	**1637.70**	**1637.70**	**1637.70**	0.00
R102.25	25	7	**547.10**	**547.10**	**547.10**	**547.10**	0.00
R102.50	50	11	**909.00**	**909.00**	**909.00**	**909.00**	0.00
R102.100	100	18	**1466.60**	1469.20	**1466.60**	**1468.30**	0.12
R103.25	25	5	**454.60**	**454.60**	**454.60**	**454.60**	0.00
R103.50	50	9	**772.90**	773.90	**772.90**	**772.90**	0.00
R103.100	100	14	**1208.70**	1225.00	**1208.70**	**1208.70**	0.00
R104.25	25	4	**416.90**	**416.90**	**416.90**	**416.90**	0.00
R104.50	50	6	**625.40**	629.00	**625.40**	**625.40**	0.00
R104.100	100	11	**971.50**	997.60	**971.50**	976.40	0.50
R105.25	25	6	**530.50**	**530.50**	**530.50**	**530.50**	0.00
R105.50	50	9	**899.30**	**899.30**	**899.30**	**899.30**	0.00
R105.100	100	15	**1355.00**	1375.80	**13550**	1359.40	0.32
Average			897.09	901.78	897.09	897.82	0.06

Table 6. The cost comparison of IPSO with other heuristics on CVRP instances

Input	$\lvert V' \rvert$	$\lvert K \rvert$	C_{best}	Pisinger and Ropke (2007)	Vidal et al. (2012)	Mester and Braysy (2007)	IPSO	Gap (%)
C1	50	5	**524.61**	**524.61**	**524.61**	**524.61**	**524.61**	0.00
C2	75	10	**835.26**	**835.26**	**835.26**	**835.26**	**835.26**	0.00
C3	100	8	**826.14**	**826.14**	**826.14**	**826.14**	**826.14**	0.00
C4	150	12	**1028.42**	1029.56	**1028.42**	**1028.42**	1030.37	0.19
C5	199	17	**1291.29**	1297.12	1291.45	**1291.29**	1297.63	0.49
C11	120	7	**1042.11**	**1042.11**	**1042.11**	**1042.11**	**1042.11**	0.00
C12	100	10	**819.56**	**819.56**	**819.56**	**819.56**	**819.56**	0.00
Average			909.63	910.62	909.65	909.63	910.81	0.10

Table 7. The cost comparison of IPSO with other heuristics on OVRP instances

Input	$\lvert V' \rvert$	$\lvert K \rvert$	C_{best}	Pisinger and Ropke (2007)	Allahyari et al. (2015)	Salari et al. (2010)	IPSO	Gap (%)
C1	50	5	**416.06**	**416.06**	**416.06**	**416.06**	**416.06**	0.00
C2	75	10	**567.14**	**567.14**	**567.14**	**567.14**	**567.14**	0.00
C3	100	8	**639.74**	641.76	**639.74**	**639.74**	**639.74**	0.00
C4	150	12	**733.13**	**733.13**	**733.13**	**733.13**	**733.13**	0.00
C5	199	17	**867.89**	896.08[a]	**867.89**	869.24	869.24	0.16
C11	120	7	**682.12**	**682.12**	**682.12**	**682.12**	**682.12**	0.00
C12	100	10	**534.24**	**534.24**	**534.24**	**534.24**	**534.24**	0.00
Average			634.33	638.65	634.33	634.52	634.52	0.02

6. References

-Ai, T. J. and Kachitvichyanukul, V. (2009) "Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem", Computers & Industrial Engineering,Vol. 56, No. 1, pp. 380-387.

-Allahyari, S., Salari, M., and Vigo, D. (2015) "A hybrid metaheuristic algorithm for the multi-depot covering tour vehicle routing problem", European Journal of Operational Research, Vol. 242, No. 3, pp. 756-768.

-Belmecheri, F., Christian, P., Farouk, Y.and Lionel, A. (2013) "Particle swarm optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows", Journal of intelligent manufacturing, Vol. 24, No. 4, pp. 775-789.

-Castro, J. P., Landa-Silva, D., and Pérez, J. A. M., (2009) "Exploring feasible and infeasible regions in the vehicle routing problem with time windows using a multi-objective particle swarm optimization approach", Nature inspired cooperative strategies for optimization, Springer, Vol. 236, pp. 103-114.

-Chen, A., Yang, G., and Wu, Z., (2006) "Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem", Journal of Zhejiang University Science A, Vol. 7, No. 4, pp. 607-614.

-Chen, C. Y. and Ye, F. (2012) "Particle swarm optimization algorithm and its application to clustering analysis", Networking, Sensing and Control, IEEE International Conference.

-Cho, P., Cheung, S. W., Edwards, M. H., Fung, J., (2003) "An assessment of consecutively presenting orthokeratology patients in a Hong Kong based private practice", Clinical and Experimental Optometry, Vol. 86, No. 5, pp. 331-338.

-Christofides, B., Mingozzi, A., and Toth, P. (1979) "The Vehicle Routing Problem, in Combinatorial optimization", B. Christofides, Editors. 1979, Chichester: Wiley, pp. 313-338.

-Chun-Hua, L., Hong, Z., and Jian, Z. (2009) "Vehicle routing problem with time windows and simultaneous pickups and deliveries", Industrial Engineering and Engineering Management, IE&EM 16th International Conference.

-Clerk, M. (2006) "Particle Swarm Optimization", London: ISTE.

-Cheng, C. Y., Chen, Y. Y., Chen, T. L., Yoo, J. J. (2015) "Using a hybrid approach based on the particle swarm optimization and ant colony optimization to solve a joint order batching and picker routing problem", International Journal of Production Economics, Vol. 170, No. 3, pp. 805-814.

-Chen, M. C., Hsiao, Y. H., Reddy, R. H., Tiwari, M. K. (2016) "The Self-Learning Particle Swarm Optimization approach for routing pickup and delivery of multiple products with material handling in multiple cross-docks", Transportation Research Part E: Logistics and Transportation Review, Vol. 91, pp. 208-226.

-Dantzig, G. B. and Ramser, J. H. (1959) "The truck dispatching problem", Management Science, Vol. 6, No. 1, pp. 80-91.

-Diana, M. and Dessouky, M. M. (2004) "A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows", Transportation Research Part B: Methodological, Vol. 38, No. 6, pp. 539-557.

-Figureueiredo, E., Ludermir, T. B., and Bastos-Filho, C. (2016) "Many Objective Particle Swarm Optimization", Information Sciences, Vol. 374, pp. 115-134.

-Goksal, F. P., Karaoglan, I. and Altiparmak, F. (2013) "A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery" Computers & Industrial Engineering, Vol. 65, No. 1, pp. 39-53.

-Gong, Y. J., Zhang, J., Liu, O., Huang, R. Z., Chung, H. S. (2012) "Optimizing the vehicle routing problem with time windows: a discrete particle swarm optimization approach", Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions, Vol. 42, No. 2, pp. 254-267.

-Han, J., Zhang, G., Hu, Y., Lu, J. (2016) "A solution to bi/tri-level programming problems using particle swarm optimization", Information Sciences, Vol. 370-371, pp. 519-537.

-Karaoglan, I., Altiparmak, F., Kara, I., Dengiz, B. (2012) "The location-routing problem with simultaneous pickup and delivery: Formulations and a heuristic approach", Omega, Vol. 40, No. 4, pp. 465-477.

-Kennedy, J. and Eberhart, R. (1955) "particle swarm optimization", proceedings of IEEE international conference on neural networks, Vol. 4, pp. 1942-1948.

-Kennedy, J. and Eberhart, R. (2001) "Swarm Intelligence", San Francisco: Morgan Kaufmann Publishers.

-Kim, B. I. and Son, S. J. (2012) "A probability matrix based particle swarm optimization for the capacitated vehicle routing problem", Journal of Intelligent Manufacturing, Vol. 23, No. 4, pp. 1119-1126.

-Kirkpatrick, S. C., Gelatt, D and Vecchi, M. P. (1984) "Optimization by simulated annealing: Quantitative studies", Journal of statistical physics, Vol. 34, No. 5-6, pp. 975-986.

-Küçükoğlu, İ. and Öztürk, N. (2015) "An advanced hybrid meta-heuristic algorithm for the vehicle routing problem with backhauls and time windows", Computers & Industrial Engineering, Vol. 86, pp. 60-68.

-Kumar, R. S., Kondaraneni, K., Dixit, V., Goswami, A. and Thakur, L. S. (2016) "Multi-objective modeling of production and pollution routing problem with time window: A self-learning particle swarm optimization approach", Computers & Industrial Engineering, Vol. 99, pp. 29-40.

-Lichtblau, T. (2002) "Discrete Optimization using Mathematica, in World Multi Conference on Systemic and Informatics", Callaos, N. Editors, International Institute of Informatics and Systemics, pp. 169-174.

-Liang, J. J., Qin, I. K., Suganthan, P. N., Baskar, S. (2006) "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions", IEEE transactions on evolutionary computation, Vol. 10, No. 3, pp. 281-295.

-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, Vol. 7, No. 1, pp. 59-78.

-Marinakis, Y., Iordanidou, G. R. and Marinaki, M. (2013) "Particle swarm optimization for the vehicle routing problem with stochastic demands", Applied Soft Computing, Vol. 13, No. 4, pp. 1693-1704.

-Mester, D. and Bräysy, O. (2007) "Active-guided evolution strategies for large-scale capacitated vehicle routing problems", Computers & operations research, Vol. 34, No. 10, pp. 2964-2975.

-Mester, D., Bräysy, O. and Dullaert, W. (2007) "A multi-parametric evolution strategies algorithm for vehicle routing problems", Expert Systems with Applications, Vol. 32, No. 2, pp. 508-517.

-Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N. and Teller A. H. (1953) "Equation of state calculations by fast computing machines", The journal of chemical physics, Vol. 21, No. 6, pp. 1087-1092.

-MirHassani, S. and Abolghasemi, N. (2011) "A particle swarm optimization algorithm for open vehicle routing problem", Expert Systems with Applications, Vol. 38, No. 9, pp. 11547-11551.

-Moghaddam, B. F., Ruiz, R. and Sadjadi, S. J. (2012) "Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm", Computers & Industrial Engineering, Vol. 62, No. 1, pp. 306-317.

-Montané, F. A. T. and Galvao, R. D. (2006) "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service", Computers and Operations Research, Vol. 33, No. 3, pp. 595–619.

-Mokhtarimousavi, S., Rahami, H., Saffarzadeh, M. and Piri, S. (2014) "Determination of the aircraft landing sequence by two meta-heuristic algorithms", International Journal of Transportation Engineering, Vol. 1, No. 4, pp. 271-284.

-Marinakis, Y., (2015) "An improved particle swarm optimization algorithm for the capacitated location routing problem and for the location routing problem with stochastic demands", Applied Soft Computing, Vol. 37, pp. 680-701.

-Naderi, B., Zandieh, M., Balagh, A. K. G. and Roshanaei, V. (2009) "An improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness", Expert systems with Applications, Vol. 36, No. 6, pp. 9625-9633
.
-Norouzi, N., Sadegh-Amalnick, M. and Alinaghiyan, M. (2015) "Evaluating of the particle swarm optimization in a periodic vehicle routing problem", Measurement, Vol. 62, pp. 162-169.

-Pisinger, D. and Ropke, S. (2007) "A general heuristic for vehicle routing problems", Computers & operations research, Vol. 34, No. 8, pp. 2403-2435.

-Planeta, D. S. (2007) "Priority Queue Based on Multilevel Prefix Tree", eprint arXiv:0708.2936

-Poli, R., Kennedy, J. and Blackwell, T. (2007) "Particle swarm optimization", Swarm intelligence, Vol. 1, No. 1, pp. 33-57.

-Romero, R., Gallego, R. and Monticelli, A. (1995) "Transmission system expansion planning by simulated annealing", IEEE Transactions on Power Systems, Vol. 11, No. 1, pp. 364-369.

-Salari, M., Toth, P. and Tramontani, A. (2010) "An ILP improvement procedure for the Open Vehicle Routing Problem", Computers & Operations Research, Vol. 37, No. 12, pp. 2106-2120.

-Shi, Y. and Eberhart, R. C. (1998) "A modified particle swarm optimizer", IEEE international conference on evolutionary computation Proceedings, pp. 69–73.

-Solomon, M. M. and Desrosiers, J. (1988) "Survey paper-time window constrained routing and scheduling problems", Transportation science, Vol. 22, No. 1, pp. 1-13.
-Solomon, M. M. (1987) "Algorithms for the vehicle routing and scheduling problems with time window constraints", Operations research, Vol. 35, No. 2, pp. 254-265.

-Thoth, P. and Vigo, D. (2002) "The vehicle routing problem", Philadelphia, PA: SIAM.

-Tian, J., Ma, W. Z., Wang, Y. L. and Wang, K. L. (2011) "Emergency supplies distributing and vehicle routes programming based on particle swarm optimization", Systems Engineering-Theory & Practice, Vol. 5, pp. 0-16.

-Ursani, Z., Essam, D., Cornforth, D. and Stocker, R. (2011) "Localized genetic algorithm for vehicle routing problem with time windows", Applied Soft Computing, Vol. 11, No. 8, pp. 5375-5390.

-Van Laarhoven, P. J. and Aarts, E. H. (1987) "Simulated annealing: theory and applications", Springer Science & Business Media, Vol. 37, pp. 7-15.

-Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N. and Rei, W. (2012) "A hybrid genetic algorithm for multidepot and periodic vehicle routing problems", Operations Research, Vol. 60, No. 3, pp. 611-624.

-Wang, H. and Chen, Y. A (2012) "A genetic algorithm for the simultaneous delivery and pickup problems with time window", Computers and Industrial Engineering, Vol. 62, No. 1, pp. 84-95.

-Wang, H., Sun, H., Li, C., Rahnamayan, S. and Pan, J. S. (2013) "Diversity enhanced particle swarm optimization with neighborhood search", Information Sciences, Vol. 223, pp. 119-135.

-Wang, K. P., Huang, L., Zhou, C. G. and Pang, W. (2003) "Particle swarm optimization for traveling salesman problem", International Conference on Machine Learning and Cybernetics, IEEE, Vol. 3, pp. 1583-1585.

-Wang, W., Wu, B., Zhao, Y. and Feng, D. (2006) "Particle swarm optimization for open vehicle routing problem", Computational Intelligence, Springer, Volume 4114 of the book series, pp. 999-1007.

-Wu, C., Liang, Y., Lee, H. P. and Lu, C. (2004) "Generalized chromosome genetic algorithm for generalized traveling salesman problems and its applications for machining", Physical Review E, Vol. 70, No. 1, pp. 016701.

-Xiao, J. M., Huang, Y. F., Li, J. J. and Wang, X. H. (2005) "Vehicle Routing Problem Based on Discrete Particle Swarm Optimization", Systems Engineering, Vol. 4, pp. 97-100.

-Xiao, J. M., Li, J. J. and Wang, X. H. (2005) "particle swarm optimization algorithm for vehicle routing problem1", Computer Integrated Manufacturing Systems, Vol. 11, No. 4, pp. 577-581.

-Xu, J. and Huang, D. (2007) "Hybrid particle swarm optimization for vehicle routing problem with multiple objectives", Computer Integrated Manufacturing Systems-Beijing, Vol. 13, No. 3, pp. 573.

-Yannis, M. and Magdalene, M. A. (2010) "Hybrid genetic - particle swarm optimization algorithm for the vehicle routing problem", Expert Systems with Applications, Vol. 37, No. 2, pp. 1446-1455.

-Zang, H., Jue, J.P. and Mukherjee, B. (2000) "Capacity allocation and contention resolution in a photonic slot routing all-optical WDM mesh network", Journal of lightwave technology, Vol. 18, No. 12, pp. 1728.

-Zhang, J. (2012) "Particle Swarm Optimization Using Crossover Operator", Journal of Convergence Information Technology, Vol. 7, No. 4, pp. 287-295.

-Zhang, L., Pang, X. H., Xia, W. and Wu, Z. (2006) "A hybrid particle swarm optimization algorithm for vehicle routing problem with time windows", Journal of Shanghai Jiaotong University, Vol. 40, No. 11, pp. 1890.

-Zhu, Q., Qian, L., Li, Y. and Zhu, S. (2006) "An improved particle swarm optimization algorithm for vehicle routing problem with time windows", IEEE Congress on Evolutionary Computation, pp. 1386-1390.