International Journal of Applied Operational Research
Vol. 1, No. 3, pp. 33-41, Winter 2012

Journal homepage: www.ijorlu.ir

A New Competitive Approach on Multi-Objective Periodic Vehicle Routing Problem

M. Alinaghian*, M. Ghazanfari, A. Salamatbakhsh, N. Norouzi

Received: May 21, 2011 ; **Accepted:** September 21, 2011

Abstract This paper presents a novel multi-objective mathematical model of a periodic vehicle routing problem (PVRP) in a competitive situation for obtaining more sales. In such a situation, the reaching time to customers affects the sale amount; therefore, distributors intend to service customers earlier than other rivals for obtaining the maximum sale. Moreover, a partial driver's benefit is related to the amount of their sale; thus, the balance of goods based on the vehicles capacity is important. Due to its complexity, it is so difficult to optimally solve this problem in a reasonable computational time. Hence, two algorithms are proposed based on multi-objective particle swarm optimization (MOPSO) and NSGAII algorithm. A comparison of our results with three performance metrics confirms that the proposed MOPSO is an efficient algorithm for solving the competitive PVRP with a reasonable computational time.

Keywords Periodic Vehicle Routing Problem, Competitive Time Windows, Multi Objective Optimization, MOPSO.

1 Introduction

The Vehicle Routing Problem with Time Windows (VRPTW) arises frequently in many distribution systems. Applications often involve additional constraints that may complicate the solution process. The VRP with hard (soft) time window constraints is abbreviated as VRPHTW (VRPSTW). The VRPSTW is a relaxation of the VRPHTW and violation from time windows constraint is allowed if a penalty is paid. Some of applications of the VRPTW include grocery distribution, school bus routing, oil and petroleum delivery, bank deliveries, postal deliveries, industrial refuse collection, and JIT (just in time) manufacturing.

In PVRPTW problem, each customer $v_i \in V$ specifies a set k (i) of combinations, and the visit days are assigned to the customer by selecting one of these combinations, thus the

*** Corresponding Author. (✉)**
E-mail: alinaghian@iust.ac.ir (M. Alinaghian)

M. Alinaghian, M. Ghazanfari
Department of Industrial Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran.

A. Salamatbakhsh, N. Norouzi
Department of Industrial Engineering, Tehran South Branch, Islamic Azad University, Tehran, Iran.

[Downloaded from ijaor.com on 2026-11-08]

vehicles must visit the customer v_i on the days belonging to the selected combination. For example, in a 6-day planning period, if the customer v_i specifies the two visit day combinations {1, 3, 5} and {2, 4, 6}, then the vehicles must visit the customer v_i on the days 1, 3 and 5 if the combination {1, 3, 5} is selected while selecting combination two means the vehicles must visit the customer i on the days 2, 4 and 6 [1].

Alegre et al. [1] applied scatter search for solving the PVRP and solved this problem for automotive Angelelli Company and Speranza (2002) by a tabu search algorithm. Their problem was near to multi depot periodic vehicle routing problem. In this problem when vehicles visited intermediate facilities their capacities replenished. Hemmelmayr et al. [2] presented variable neighborhood search heuristic to solving PTSP problem. Their paper had two contributions, first they used VNS algorithm to solve the PVRP and second their method gain best results in some test problems. In the beginning, authors generated initial solution choice combination of customers randomly then they applied Clarck and Wright's algorithm for daily tours. Then the authors made use of OR interchange instead of local search method. Francis et al. [3] presented an extension type of periodic vehicle routing problem that the number of visit for each customer was one of the dictions in the problem that this problem was called as PVRP with Service Choice.

Due to the importance of service time presented by other companies in real world, distribution companies design the routes of fleets with respect to the condition of other competitors to obtain the maximum sale.

In competitive environment several distributers are in competition and arriving to the customers earlier than other competitors impress to amount of sale therefore each competitor tries to arrive to the customers earlier than the others. The demand of each customer is divided in two parts. The first part (d_{ini}) does not depend on the time and should be sent to the customer completely and the second part (d_{tdi}) is the time-dependent. It will be lost if the rival's arrival time is earlier than vehicle's arrival time to the customer. Therefore, the distributor's reaching time to the customers influences on the amount of sales.

This type of PVRP needs to consider some other parameters such as competition between distributors and the best time to visit each customer for getting optimum cost and purchase that classical PVRP and PVRPTW are unable to achieve the good solutions for this kind of problem. As we know, the competitive approach on the PVRP has not been considered so far; hence, this paper can be considered as the first work on the PVRPCTW. In this paper the proposed problem is solved using MOPSO and the results compared by results obtained by NSGAII algorithm.

The rest of the paper is organized as follows. Section 2 introduces the PVRPCTW and details of the problem solving methodologies are represented in section 3. Section 4 summarizes the result of this study and suggests further direction in this research. Finally, a conclusion is presented in section 5.

[Downloaded from ijaor.com on 2026-11-08]

2 The statement of the problem

In a competitive environment, it's important to attend the time of service presented to the customers in such a manner that if the vehicle presents service to the customers later than its rival, it will miss a partial of its sale. For this reason, distributing companies define routing of their vehicles based upon other rival companies' strategies for serving customers. In other words, distribution of goods is not exclusive in the real world situation and more consideration is needed for the vehicle routing problem in competitive environment. This competition is occurred in which products have short life time and customers need special devices for keeping them. In this situation reaching the time of distributor influences the sales amount. Hence; a problem presented in this paper to route of vehicles in competitive environment such that it can be considered as a new version of PVRP with time windows. This problem is proposed under the condition that a competition is between distributors to obtain the market share.

Before presenting the model, these parameters are introduced to clarity the problem.

The number of days in the planning horizon is denoted by M. A set of m homogenous vehicles is given and the maximum capacity and working time are denoted by Q and T respectively. Let G= (V, A) be a complete graph consisting of vertex and arc sets. $V=\{v_0,v_1,...,v_n\}$ is the vertex set and that consists of n + 1 nodes and , $A=\{(v_i,v_j): v_i,v_j \in V\}$ is the arcs set which each arc (v_i,v_j) is associated with a non-negative cost C_{ij}. The v_0 is associated to the depot and remaining vertices of V represent customers to be serviced.

e_i	Lower bound of rival's arrival time to node i
l_i	Upper bound of the rival's arrival time to node i
tri	Rival's arrival time to node i
tdi	Actual distributer's vehicle arrival time to node i
$fr(x)$	Probability distribution function of the rival's arrival time to node i
$Fr(x)$	Cumulative distribution function of the rival's arrival time to node i
$dtdi$	Time-dependent demand of node i
$dini$	Time-independent demand of node i
Di	Maximum number of the customer's demand in node i
$E(Di)$	Expected value of the customer's demand in node i

Each node has predetermined the demand denoted by D_i, demand of node i is divided in two parts, $Di = d_{tdi} + d_{ini}$, d_{tdi} is time dependent demand of node i and amount of it depend on time of visiting customer. d_{ini} is part of demand that independent to the time of service. A rival presents its services to each vertex with probability distribution function within a specified time window [e_i, l_i] in each day that e_i and l_i are nonnegative. e_i is the earliest rival's arrival time to node i and l_i is the latest rival's arrival time to node i. Each arc has nonnegative associated travel times c_{ij}. The travel time c_{ij} includes a service time at node i. The VRPTW consists of designing M tours on G in each day that satisfied some conditions as: 1) every

[Downloaded from ijaor.com on 2026-11-08]

route starts and ends at the depot; 2) each customer must be served with exactly one vehicle and split service and multiple visits are forbidden; 3) a vehicle is allowed to arrive before the opening of the time window without missing any sale, but arriving after the latest time window causes the distributer lose the first part of demand, d_{tdi}. If arriving time occurs in the time windows, expected value of visit customer before rival reduces according to the probability distribution function of rival's arrival time. The rival's service time distribution to the customers can be determined using stochastic methods. In this paper, it is assumed that the rival's arrival time distribution to customer i follow the uniform distribution in which probability of serving node i before its rival calculated using Eq.1 as follows:

$$p(t_{ri} > t_{di}) = 1 - F_r(t_{di}) \tag{1}$$

In Eq. 1 the P(x) is the probability of x and t_{ri} and t_{di} are arriving time of rival and distributer respectively and $F_r(x)$ is the cumulative distribution function of rival's arrival time. The Probability of reaching the vehicle to customer i earlier than its rival is shown in Fig. 1.

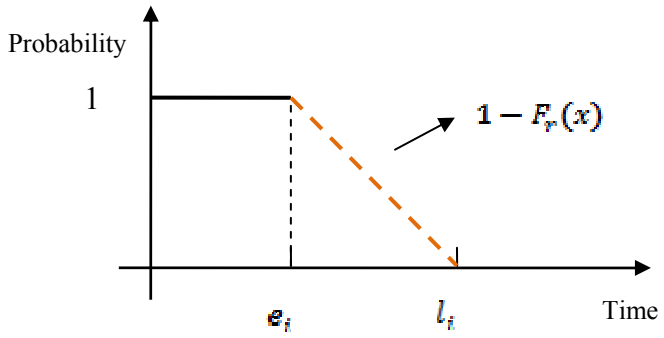


Fig. 1 Probability of receiving to node i earlier than its rival

Now, the probability of reaching to customer i earlier than its rival in time of t_{di} is computed by:

$$p(t_{di} < t_{ri}) = \begin{cases} 1 & if \quad t_{di} \le t_{li} \\ 1 - F_c(t_{di}) & if \quad t_{li} \le t_{di} \le t_{ui} \\ 0 & if \quad t_{di} \ge t_{ui} \end{cases} \tag{2}$$

Due to the above equation, if the vehicle reaches to node i in the bound $[0,t_{li}]$, then it will reach to the customer earlier than its rival definitely. But, if the vehicle reaches to node i in the bound $[t_{li}, t_{ui}]$, then with the probability of $p(t_{di})=1\text{-}F_c(t_{di})$, it will serve node i earlier than its rival. Finally, if the vehicle reaches to node i after t_u, it will reach to node i with the probability of $p(t_{di})= 0$ earlier than its rival. In this case, the customer's time-dependent demand is affected to the service time presented, while if the rival distributor reaches to node i earlier, this part of the demand will be missed.

[Downloaded from ijaor.com on 2026-11-08]

3 Solution procedures

In this section first the framework of propose MOPSO is presented, and then NSGAII algorithm is described.

3.1 MOPSO framework

The MOPSO framework for solving the problem is as follows.

1. Generate an initial solution vector $X(i)$ for i=0: pop size
2. Initialize the velocity of each particle for i=0: pop size, $VEL\ (i)$ =0
3. Initialize *Pbest* of each particle; for i=0: pop size*; Pbest*(i)= $X(i)$
4. Calculate the objective functions of each particle.
5. Store the non-dominated vectors particles' positions based on their objective functions in repository.
6. In objective space, each particle's coordinates are described according to the values of its objective functions. Divide each axis of objective space to n segments and generate hypercubes in this space.
7. Do the following steps till the maximum number of iteration is achieved
8. Update the velocity of each particle as follows:

$$VEL(t+1)= w\ VEL(t) + c_1\ Rand1(pbest- x\ (t)) +c_2\ Rand2(REP- x\ (t)) \tag{3}$$

where t is the current iteration of the algorithm, c_1 and c_2 are the acceleration coefficients that affect on the particles movement, *Rand*1 and *Rand* 2 are uniform random numbers in the interval [0, 1], and w is the inertia weight which controls the convergence of the algorithm. p_{best} is the personal best position of a particle, and *REP* is the global best position that dominates all particles taken from the repository.

9. Compute the new particle's position by the previous particle's position and its velocity vector based on Eq. (3).

$$x\ (t+1)=x\ (t)+VEL\ (t) \tag{4}$$

10. Update the repository and then select *REP* for the next iteration. Each solution is selected as *REP* with the probability based on the following formula. The fewer particles in each hypercube, the more chance to be selected.

$$p_i = \frac{e^{-\beta . n_i}}{\sum_{j=1}^{n} e^{-\beta . n_i}} \tag{5}$$

[Downloaded from ijaor.com on 2026-11-08]

11. Update *Pbest* of each particle. If the new particle's position dominates its *Pbest*, then the new position replaces in *Pbest* and vice versa. If neither of them is dominated by the others, then one of them is selected randomly with the probability 0.5.

3.2 The NSGA-II framework

Now, the NSGA-II steps to implement are as follows.

1. Generate a population of solutions with size N.
2. Sort the population of non-domination solutions into each front. Assign a rank to each individual based on their fitness values. Those solutions are never dominated by any other solution are in rank 1(front 1) and the next best solutions are in rank 2 (front 2) and so on.
3. Calculate the crowding distance for each individual in order to estimate the density of the individuals to each other in each front.
4. Select parents from the population by using a binary tournament selection based on the front rank and crowding distance. An individual is selected if it has been located in a lower rank or has been a greater crowding distance than the other if it belongs to the same rank with other individuals.
5. Generate offspring (Q_t) from the selected population (P_t) by crossover and mutation operators discussed in details in [4].
6. Sort the combined populations containing of the previous population (P_t) and current offspring (Q_t) based on non-domination and only the best N individuals are selected, where N is the population size. An individual is selected based on its rank. In the last front, individuals with greater *CD* are selected. Repeat steps of the algorithm for the pre-determined iterations.

4 Computational experience

Evaluating the multi-objective algorithms performances is more complicated than the single objective algorithms. Many researchers in the last decade have investigated some good criteria to evaluate the multi-objective algorithms performances [5]. To evaluate the quality of the solution sets obtained by the MOPSO and NSGA-II algorithms, three performance metrics are used as follows:

4.1 The quantity metric

The quantity metric measures the number of the non-dominated solutions set of each algorithm.

[Downloaded from ijaor.com on 2026-11-08]

4.2 The *C* metric

The *C* metric was introduced by Zitzler (1998), in which two sets of non-dominated solutions can be compared to each other by this metric.

$$C(A,B) = \frac{\left|\left\{b \in B \mid \exists a \in A : a \geq b\right\}\right|}{\left|B\right|} \tag{6}$$

where A and B are two sets of non-dominated solutions for the first and second algorithms. $|X|$ illustrates a number of members of set X.

4.3 The spacing metric

The spacing metric has been introduced by Schott (1995). It measures the solution sets diversity on the Pareto front distributed in the objective space. This metric is computed by:

$$E = \sqrt{\frac{1}{N-1}\sum_{i=1}^{n}\left(\bar{d} - d_i\right)^2} \tag{7}$$

$$d_i = \min, j \neq i\left(\sum_{m=1}^{M}\left|f_i^m - f_j^m\right|\right) \tag{8}$$

where m is the objective functions counter and d_i is the minimum distance of solution i with other solutions in the Pareto set in the objective space. $\bar{d}$ is the mean of all d_i and n is the size of the non-dominated solution sets.

To evaluate the proposed MOPSO and NSGA-II for the given problems, the standard benchmark instances proposed by Cordeau are used and all of the customers' demands are considered as time dependent demands.

The obtained results from solving test problems are shown in Table 1. In this table, column one is the problem name. Columns 2 to 5 are related to the MOPSO algorithm to demonstrate the run time, quantity metric, *C* metric and spacing metric, respectively. The other columns are related to the performance of the NSGA-II algorithm.

[Downloaded from ijaor.com on 2026-11-08]

Table 1 Comparison of the proposed algorithms performances

Problem	MOPSO				NSGA-II			
	Run.time	Q.Metric	C.Metric	S.Metric	Run.time	Q.Metric	C.Metric	S.Metric
A1	132.24	5	1	0.34	297.55	32	0	2.56
A2	135.47	14	0.95	2.44	298.63	65	0.20	2.78
A3	134.71	22	0.87	2.23	287.15	43	0.28	1.98
A4	148.53	20	1	3.21	376.45	12	0	2.45
A5	136.21	8	0.98	0.87	327.33	8	0.18	1.19
A6	134.05	6	1	1.59	310.15	5	0	1.54
A7	135.41	7	0.83	0.66	297.53	27	0.22	2.22
A8	135.23	6	1	0.86	302.31	7	0	1.66
A9	93.54	2	1	2.13	295.31	19	0	1.98
A10	102.43	13	1	1.52	305.39	56	0	1.12
Average of Problems	128.78	10.3	0.97	1.58	309.78	27.45	0.088	1.948

The run times of MOPSO and NSGA-II are acceptable; however, the run time of MOPSO is significantly better than NSGA-II. The average runtime for MOPSO is 128.78 while this time for NSGA-II is 309.78 seconds. The maximum runtime is for the A4 problem with 148.53 and 376.45 seconds in MOPSO and NSGA-II, respectively.

5 Concluding remarks and future directions

This paper presented a new class of PVRP that arise in a competitive environment which named the periodic vehicle routing problem with competitive time windows (PVRPCTW). This type of PVRP needs to consider some other parameters such as competition between distributors and the best time to visit each customer to get the optimum cost and purchase that classical PVRP and PVRPTW are unable to achieve the good solutions for these kinds of problems. The objectives of the problem were to find the short routes with the minimum travel cost, maximum sales for company and to balance the distributed goods by vehicles regarding their capacities. In this paper, the proposed problem was solved using MOPSO, and the results were compared by results obtained by NSGAII. The results showed that the MOPSO algorithm was better performing according to the performance metrics.

References

1. Alegre, J., Laguna, M., Pacheco, J., (2007). Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts. European Journal of Operational Research, 179, 736–746.
2. Hemmelmayr, C., Doerner, K., Hartl, R. F., (2009). A variable neighborhood search heuristic for periodic routing problems. European Journal of Operational Research, 195 ,791–802.

[Downloaded from ijaor.com on 2026-11-08]

3. Francis, P., Smilowitz, K., Tzur, M., (2006). The period vehicle routing problem with service choice. Transportation Science, 195, 342–354.
4. Angelelli, E., Speranza, M. G., (2002). The periodic vehicle routing problem with intermediate facilities. European Journal of Operational Research, 137(2), 233–247.
5. Jaszkiewicz, A., (2001). Multiple objective metaheuristic algorithms for combinatorial optimization. Habilitation Thesis, Poznan University of Technology, Poznan.
6. Schott, J. R., (1995). Fault tolerant design using single and multicriteria genetic algorithms optimization. Master's Thesis, Department of Aeronautics and Astronautics, Massachusetts.
7. Zitzler, E., Thiele, L., (1998). Multiobjective optimization using evolutionary algorithmsa comparative case study. In: A. E. Eiben, T. Back, M. Schoenauer and H. P. Schwefel (Eds.), Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V), Berlin, Germany, 292-301.

[Downloaded from ijaor.com on 2026-11-08]