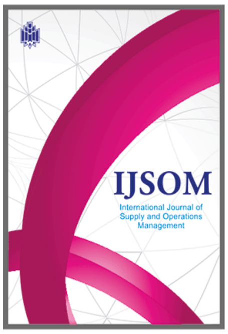
International Journal of Supply and Operations Management

IJSOM

February 2016, Volume 2, Issue 4, pp. 1003-1020
ISSN-Print: 2383-1359
ISSN-Online: 2383-2525
www.ijsom.com



Capacitated Windy Rural Postman Problem with several vehicles: A hybrid multi-objective simulated annealing algorithm

Masoud Rabbani [a,1], Safoura Famil Alamdar[a], Hamed Farrokhi-Asl[b]

[a] *School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran*
[b] *School of Industrial Engineering, Iran University of Science &Technology, Tehran, Iran*

Abstract
This paper presents the capacitated Windy Rural Postman Problem with several vehicles. For this problem, two objectives are considered. One of them is the minimization of the total cost of all vehicle routes expressed by the sum of the total traversing cost and another one is reduction of the maximum cost of vehicle route in order to find a set of equitable tours for the vehicles. Mathematical formulation is provided. The multi-objective simulated annealing (MOSA) algorithm has been modified for solving this bi-objective NP-hard problem. To increase algorithm performance, Taguchi technique is applied to design experiments for tuning parameters of the algorithm. Numerical experiments are proposed to show efficiency of the model. Finally, the results of the MOSA have been compared with MOCS (multi-objective Cuckoo Search algorithm) to validate the performance of the proposed algorithm. The experimental results indicate that the proposed algorithm provides good solutions and performs significantly better than the MOCS.

Keywords: Windy Rural Postman Problem; Multi-objective simulated annealing algorithm (MOSA); Multi-objective Cuckoo Search (MOCS); NP-hard problem.

1. Introduction

The aim of arc Routing Problems (ARP's) are employed to detect a traversal on a graph which satisfy some conditions corresponding to the links of the graph. According to Dror (2000), ARP's have a large number of applications in real-life situations such as collection and delivery of goods, network maintenance (electrical lines or gas mains inspection), snow removal, mail distribution and garbage collection, which have been studied in previous researches.
The famous Chinese Postman Problem (CPP) has been first introduced by Guan (1962) in the

[*] Corresponding author email addresses: mrabani@ut.ac.ir

literature. It includes shortest tour detection which travels all the links of a given graph *G*. Afterwards, this problem has been developed in several cases. Orloff (1974) introduced the Rural Postman Problem (RPP), in which only a given subset of 'required' links should be traversed by the tour. Also the General Routing Problem (GRP), in which the graph has a set of 'required' edges which should be traversed and a given set of 'required' vertices which should be visited proposed by Orloff (1974). All the edges can be traversed in both directions with the equal cost. On the other hand, the CPP can be defined on a directed graph which is known as the Directed Chinese Postman Problem (DCPP), As Edmonds & Johnson (1973) stated, where all the links of a given graph are arcs that have to be traversed in a given direction. Also Christofides et al. (1986), Savall (1990), and Gun (1993) proposed the DRPP and the DGRP which they were defined on a directed graph to obtain the 'directed' versions of the RPP and the GRP, respectively. If a graph *G* has edges and arcs simultaneously, it is called a mixed graph. The CPP defined on a mixed graph is named the Mixed Chinese Postman Problem (MCPP).

Edmonds & Johnson (1973) defined two model including CPP and the DCPP. Similarly, any routing problem defined on a mixed graph is a generalization of the corresponding problems defined on undirected and directed graphs. Finally, Minieka (1979) proposed the Windy Postman Problem (WPP) consists of finding a tour with minimum cost where all the edges of an undirected graph must be traversed, in which the cost of traversing an edge (i,j) in a given direction, c_{ij} , can be different from the cost of traversing it in the other direction, c_{ji} ,. WPP also studied by Guan (1984); Win (1987); Grötschel & Win (1992). Clearly, the WPP covers both the undirected CPP (when $c_{ij} = c_{ji}$) and the DCPP (each arc (i,j) with cost c can be considered as an edge with costs $c_{ij} = c$ and $c_{ji} = 1$) and therefore WPP is a generalization of the MCPP. Except the CPP and the DCPP, which can be solved in polynomial time, all the other problems are NP – hard problems.

Another appealing Arc Routing Problem is the Windy Rural Postman Problem (WRPP) which is the combination of WPP and RPP. The WRPP defined on an undirected graph with two non-negative costs corresponding to the each edge where the cost of an edge depends on the direction of traversal and a subset of edges requiring some service to be done along them (“required’’ edges), we should find a tour with minimum cost, traversing each required edge at least once.

As mentioned, by using an appropriate definition of the costs related to the each edge, the undirected, directed and mixed Problem are special cases of the WPP. Similarly, the Rural Postman Problem defined on an undirected, directed or mixed graph is a specific form of the WRPP. Therefore, the Windy Rural Postman Problem is also NP-hard problem. The study and solution of the WRPP has obtained great interest because it is the most general single-vehicle Arc Routing Problem proposed so far and it has practical applications in several real-life situations. This paper presents the two objective capacitated Windy Rural Postman Problem with several vehicles which has been considered for the first time in the literature.

2. Literature review

In the postman problem, a postman wants to deliver the mail in his town. He starts at his office, delivers the mail along all streets and then goes back to the office. The question is to design a tour of minimum cost for him. In the literature, this problem and similar ones are recognized as postman problems. The undirected postman problem which was proposed by Guan (1962) is known as the Chinese postman problem (CPP). In the Chinese postman problem (CPP) the cost of traversing an edge in one direction is the same as that of traversing it in the opposite direction. But, for some real-world applications, this is not the appropriate case. The problems in which the cost of traversing arcs is dependent on the direction are called the windy postman problem (WPP). The Windy Postman Problem (WPP), is an appealing extension of the classical Chinese Postman Problem (CPP). On windy networks, an edge may be traversed in both directions with different costs.

The WPP was proposed by Minieka (1979). It shows the work that a postman should carry out on a windy day. On a windy day, the cost of traversing a street in the same direction of the wind will be unequal with traversing it against the wind.

According to Edmonds & Johnson (1973), the solution of the CPP may be obtained using a polynomial-time bounded matching algorithm. But, Brucker (1981); Guan (1984) showed that the WPP is NP-complete. In fact, Guan (1984) showed that the WPP is computationally as hard as the Mixed Chinese Postman Problem (MCPP) which is NP-complete (Papadimitriou, 1976).

For tackling the aforementioned problems, several heuristic solution procedures are applied, as presented by Guan (1984). Win (1989) have been presented algorithms to solve the WPP approximately. In Pearn & Li (1994) proposed another heuristic for the WPP that consists of carrying out the stages of Wins algorithm in a different order. An integer linear programming formulation for the WPP was presented by Grötschel & Win (1992). They applied a cutting plane algorithm for the (NP-hard) windy postman problem. This algorithm which is based on a partial linear definition of the windy postman could be used for solving the mixed, directed, and undirected postman problems.

Martínez (2008) by considering a family of polyhedral related to prove that Series–parallel graphs are windy postman perfect. In an undirected graph *G*, polyhedron $O(G)$ induced by a linear programming relaxation of the windy postman problem is considered. He showed that if $O(G)$ is integral, *G* will be windy postman perfect. Based on the ellipsoid method, there exists a polynomial-time algorithm to solve the windy postman problem for the class of windy postman perfect graphs.

Dussault et al. (2013) proposed a practical windy Postman Problem which considers winter street maintenance. In this problem, traversing a street after it has been plowed is faster than traversing it before plowing. With this consideration, the concept of precedence in a route is introduced, where the order of arcs traversed in the route is important. For solving this problem, an algorithm that constructs an initial solution and improves it iteratively by local search and re-initialization procedure is presented.

Corberan et al. (2005) considered the Windy General Routing Problem (WGRP), in which only a subset of edges must be traversed and a subset of vertices must be visited. The description of the

WGRP polyhedron, for some general properties and some large families of facet-inducing inequalities were presented. Since the WGRP consist of several prominent routing problems, they also presented a global view of the associated polyhedral.

Corberan et al. (2007) solved the Windy General Routing Problem with an exact algorithm based on a cutting-plane algorithm that identifies violated inequalities of several classes of facet-inducing inequalities for the corresponding polyhedron. The capability of the algorithm for solving large-size instances of several routing problems such as undirected, mixed, and windy graphs has been shown. Micó & Soler (2011) considered turn penalties and forbidden turns for the capacitated general windy routing problem which are applicable in many real-life conditions such as in downtown areas and for large vehicles. They presented a method to solve this problem optimally and heuristically by conversion it in to a generalized vehicle routing problem. Benavent et al. (2005) described and tested several heuristic algorithms for the windy rural postman problem, in which only a subset of edges must be traversed. They proposed two constructive algorithms for the WRPP and several improvement procedures were discussed. Benavent et al. (2007) presented several heuristic algorithms and a cutting- plane algorithm for the Windy Rural Postman Problem. A formulation of the WRPP, three constructive heuristic algorithms, a cutting-plane algorithm was proposed. Several families of valid inequalities that are used in a cutting-plane procedure for the WRPP were presented.

In Benavent et al. (2009), the Min-Max version of the Windy Rural Postman Problem with several vehicles was presented. In this problem, the objective is to minimize the length of the longest tour. They proposed an ILP formulation and investigated its associated polyhedron. A branch-and-cut algorithm was used based on its partial description and computational results for a large set of examples. Benavent et al. (2010) proposed a metaheuristic for the min–max version of the windy rural postman problem with *K* vehicles. In this problem the objective was to minimize the length of the longest tour in order to find a set of balanced tours for all vehicles. The proposed algorithm was based on the composition of a multi-start procedure with an iterated Local Search and was shown that can produces good feasible solutions in reasonable computing times.

In Benavent et al. (2011), the min–max windy rural postman problem is considered. Several new facet-inducing inequalities obtained from the WRPP were described; also some inequalities that have to be satisfied by any optimal solution were presented. They provided a branch-and-cut algorithm that takes benefit of both these inequalities and good feasible solutions for min–max *K*-WRPP obtained by a metaheuristic. Computational experiments on a large set of examples were also done.

According to aforementioned papers, the contribution of this study is designing a metaheuristics (multi-objective simulated annealing (MOSA) algorithm) for solving capacitated Windy Rural Postman Problem with several vehicles. To the best of our knowledge there is not any paper which considers multi-objective Windy Rural Postman Problem with several vehicles. The Windy Rural Postman Problem (WRPP) is an important Arc Routing Problem which generalizes most of the single-vehicle Arc Routing Problems. In real situation, in addition to the minimization of the total cost of all vehicle routes, the minimization of the length of the longest tour is preferable when the service to every customer as early as possible is important. Also, an assumption of several vehicles with specific capacity to serve customers is more realistic than one vehicle with unlimited capacity. The problem considered in this paper is applicable in the situations in which the cost of traversing on path is dependent on the direction of route. We use the MOCS to validate the performance of the proposed algorithm because this new algorithm has been shown to be very successful in tackling multi objective optimization problems and also characterized by the reduced number of parameters causing robustness of algorithm and this algorithm provides effective results for multimodal functions in comparison with both genetic algorithms and particle swarm optimization (Yang & Deb, 2010). Multi-objective Cuckoo Search is development of Cuckoo Search for multi-objective problems and the results of this method which is compared with other multi-objective algorithms indicate the power of this method in solving these problems.

The rest of the paper is organized as follows: In the next section, Problem description and the mathematical model are presented. In Section 4, the multi-objective simulated annealing (MOSA) algorithm is developed for solving the problem. Computational experiments are performed for evaluation of performance of the suggested meta-heuristic are given in Section 5. In Section 6, the results of the proposed algorithm is compared with MOCS (multi-objective Cuckoo Search algorithm) to validate the performance of it. Finally, Section 7 includes concluding remarks and future researches.

3. Problem description

In this paper, we consider the capacitated *K*-vehicles Windy Rural Postman Problem that can be defined as follows: Given a windy graph, a distinct vertex (the depot), a subset of required edges, and a fixed number of *K* vehicles. The purpose of this problem is determination a set of tours for *K* vehicles in a way that each tour starts and ends at the depot and every required edge is serviced by exactly one of these *K* vehicles. Two objective functions are considered including minimization of the total cost of all vehicle routes and another one is minimization of the maximum cost of vehicle route. Last objective causes to find a set of balanced tours for the vehicles which is preferable when servicing to every customer as early as possible is important (Ahr & Reinelt, 2006).

In windy Rural Postman Problem, an undirected graph *G= (V, E)* is given with two non-negative costs c_{ij} and c_{ji} related to costs of traversing each edge $e=(i,j)$ from i to j and from j to i, respectively and a subset of required edges E_R corresponding to those edges that must be served. We denote node $0 \in V$ representing the depot. A set of K similar vehicles with capacity q is considered. We assume that the demand of each required edge e does not exceed from capacity q to guarantee that the problem is feasible. We must find a tour for every vehicle in a way that each required edge is traversed at least once by one vehicle.

If E_R=E, this means that the WRPP reduces to the Windy Postman Problem (WPP) which all the edges must be traversed. If a suitable definition of the costs c_{ij} and c_{ji} related to each edge $e=(i,j)$ is used, it will be easy to show that the undirected, directed and mixed Problem are specific cases of the WPP. Similarly, the Rural Postman Problem defined in an undirected, directed or mixed graph is a specific case of the WRPP. Therefore, the Windy Rural Postman

Problem is also NP-hard problem.

Because of simplicity, we assumed that all the vertices in V are incident with required edges. This is not a restriction and the generality does not lost because there exists an easy way to convert a case which is not satisfying this assumption into an equivalent one which does (see, e.g., Benavent et al., 2011).

Similar to Benavent et al. (2011), a subset of nodes ($S \subseteq V$) is considered. The edge cut-set related to S is shown by $\delta(S) = \{(i,j) \in E: i \in S, j \in V\backslash S\}$. If $S = \{i\}$, we will denote $\delta(i)$ instead of $\delta(\{i\})$. Furthermore, let $E(S) = \{(i,j) \in E: i,j \in S\}$. Consider two node subsets $S, S' \subseteq V$, and ($S: S'$) show the edge set with one end-point in S and the another in S'. Finally, $\delta_R(S)$, $E_R(S)$, $(S: S')_R$ show the previous sets for the required edges. The following notations are used in the mathematical formulation.

x_{ij}^k	the number of times edge *e* is traversed by vehicle *k* from *i* to *j*
x_{ji}^k	the number of times edge e is traversed by vehicle k from j to i
y_e^k	if required edge *e* is serviced by the vehicle *k*, $y_e^k = 1$; otherwise=0.
d_e	demand of required edge *e*
q	capacity of vehicles
m	number of vehicles

According to the above assumptions and notation, the problem can be formulated as follows:

$$min \sum_{k=1}^{m} \sum_{e \in E} (c_{ij}\, x_{ij}^k + c_{ji} x_{ji}^k) \qquad (1)$$

$$min(\max_{1 \le k \le m} \sum_{e \in E} (c_{ij}\, x_{ij}^k + c_{ji} x_{ji}^k)) \qquad (2)$$

s.t.

$$\sum_{\forall k} y_e^k = 1 \qquad \forall \quad e \in E_R \qquad (3)$$

$$x_{ij}^k + x_{ji}^k \ge y_e^k \qquad \forall \quad e \in E_R, \forall\, k \in \{1,2, \dots, m\} \qquad (4)$$

$$\sum_{(i,j) \in \delta(i)} (x_{ij}^k - x_{ji}^k) = 0 \qquad \forall \quad i \in V, \forall\, k \in \{1,2, \dots, m\} \qquad (5)$$

$$\sum_{(i,j)\in\delta(s)} (x_{ij}^k + x_{ji}^k) \geq 2y_e^k \qquad \forall s \in V\backslash\{1\}, \forall\ e \in E_R(s), \forall\, k \in \{1,2,\dots,m\} \tag{6}$$

$$\sum_{e\in E_R} y_e^k d_e \leq q \qquad \forall\, k \in \{1,2, \dots, m\} \tag{7}$$

$$x_{ij}^k \ , x_{ji}^k \geq 0 \qquad \forall i,j \in V, \forall k \in \{1,2,\dots,m\} \tag{8}$$

$$y_e^k \in \{0,1\} \qquad \forall e \in E_R, \forall k \in \{1,2,\dots,m\} \tag{9}$$

Objective function (1) minimizes the total costs of all vehicle routes and Objective function (2) minimizes the maximum cost of vehicle routes. Equations (3) guarantee that every required edge is serviced by exactly one vehicle and inequalities (4) which are called traversing inequalities, enforce a vehicle traverse the edges it serves. Symmetry equations (5) make that every vehicle route to be symmetric. Connectivity inequalities (6) guarantee each vehicle route connects the edges which it serves and the depot. Constraint (7) ensures that the vehicle capacity is never exceeded. Finally, Constraints (8) and (9) show the binary and non-negativity limitations of decision variables.

4. Proposed Metaheuristic algorithm

The idea of simulated annealing (SA) is taken from the physical process of annealing. Kirkpatrick et al. (1983) used SA for optimization for the first time; and after that it has been used successfully in many problems, as reviewed in Suman & Kumar (2006). SA is a stochastic search technique which uses a probability function to escape local optima. It starts from an initial solution, S, and goes to a new solution, S', in the neighborhood of the initial solution by using a specific neighborhood generation procedure. Then, the amount of change in the objective function value is calculated, $\Delta = f(S') - f(S)$, if $\Delta < 0$, then a move to the new solution is done (for a minimization problem) and If $\Delta > 0$, the move is done with a specific probability, usually is calculated by $exp^{-\Delta/T}$, which T is a control parameter related to temperature in the analogy of a physical annealing schedule. This process is repeated until a predetermined stopping condition is satisfied. Commonly, the temperature T is uniformly decreased during the search which is called the cooling (or annealing) schedule (Suman & Kumar, 2006).

Some authors have used SA in the multi-objective context, but a few of them synthesize the concept of Pareto-dominance (Bandyopadhyayet al., 2008).

Czyzak & Jaszkiewicz (1998) modified simulated annealing algorithm for multi-objective optimization contexts and developed Pareto simulated annealing (PSA). In single-objective simulated annealing algorithm only one candidate for the final solution is used, but in PSA a set of interacting solutions which are called the generating set S is used at each iteration to propagate new solutions. Usually, the initial set of generating solutions is produced randomly. Then, based on the results of the prior iteration, the future sets of generating solutions are produced applying a random swapping procedure. Among solutions in a non-dominated set, any new generated solution that is not dominated by its precedent solution is checked for Pareto dominance. If the latest generated solution is non-dominated, it is added to the non-dominated set. Originally, all solutions in the non-dominated set will be removed from it if they are dominated by the added solution. PSA maintains some solutions based on a probability function P. In following, the specifications of the modified algorithm for solving capacitated Windy Rural Postman Problem with several vehicles will be explained.

4.1. Solutions representation

Most algorithms use integer representation for solving vehicle routing problems. We develop solution representation scheme as follows: In this algorithm, the length of the chromosome (solutions) has been considered equivalent to the number of variables, in which the number of variables is equal to the number of required edges (customers) plus the number of vehicles minus 1. For example, if the number of required edges is 10 and the number of vehicles is 3, so the number of components in the representation matrix will be 12. Representation matrix is a row matrix with the length of 12, in which the numbers less than 11 show the required edges.

For the example, the representation structure shows in Fig.1 indicates that the first vehicle departs from the depot, and visits customers 3, 6 and 1 before returning to the depot. The second vehicle departs from the depot, and visits customers 4, 2, 5 and 7 before returning to the depot and the third vehicle departs from the depot, and visits customers 8, 10 and 9 before returning to the depot.

3	6	1	12	4	2	5	7	11	8	10	9

Figure1. An example of solution representation for the capacitated 3-vehicles Windy Rural Postman Problem

4.2. Determining initial solution

In this paper, initial solution is a random permutation vector of required edges for all vehicle routes.

4.3. Cooling function

An important issue in designing simulated annealing is scheduling the cooling temperature. In this algorithm, the Cooling function is computed from following equation:

$$T = (T_0 - T_f) \text{ /number of iterations} \qquad (10)$$

where T_0 is an initial temperature, T_f is the temperature of the final iteration.

4.4. Creating new solution (neighborhood search)

In this algorithm, an innovative method consists of three steps are designed to create new best solution:

Step1. Here, for creating a solution two methods are considered in which the algorithm will choose one of them randomly in each iteration:

1. Swap: Two components of representation matrix are randomly chosen and their values are exchanged. Most of the time, this method keeps information corresponding to the neighborhood (4 junctions are broken down) and will cause more damage in genes' information.
2. Inversion: Two components of representation matrix are randomly chosen and substring between them is reversed. Most of the time, this method preserves information corresponding to the neighborhood (only 2 junctions are broken down), but it destroys sequence information.

Step2. After creating new solution from step1, crossover is done to create new solutions. Here, single-point crossover is chosen in which one column (gene) is chosen from chromosome. The first sector from the first chromosome and the second sector from the second chromosome create the first offspring. Also, the second offspring is created from the second sector of the first chromosome and the first sector of the second chromosome.

Step3. Then among the produced solutions in previous steps, the best of them is chosen as the new solution (neighborhood).

4.5. Solution acceptance

At first, the difference between the last solution and its neighborhood is calculated . New solution is accepted, if the objective functions of current solution does not dominate the objective functions of new generated solution ; otherwise it is accepted based on a probability $P = exp^{-\Delta f/T}$, where Δf is he difference between the number of solutions which dominates current solution and new generated solution. If P is greater than a random number between 0 and 1, then the new solution will be selected.

4.6. Non-dominated sorting

After Solution acceptance, non-dominated sorting is performed upon the estimated Pareto solution until current iteration. Crowding distance is computed using the Equation 11 in order to create a diverse front. In sorting, solutions are arranged from big to small based on crowding distance and ranged base on Pareto front from smallest rank to biggest rank. Selection criteria of the algorithm are solution rank firstly, and crowding distance corresponding to the solution secondly. The solution with lower rank and greater crowding distance is more desirable.

$$cd_i = \sum_{j \in \{F_{f_i}\}, i \neq j} d_{ij} \tag{11}$$

$$d_{ij} = \sqrt{\sum_{m=1}^{2} \left(\frac{f_m(i) - f_m(j)}{max\{f_m(0)\} - min\{f_m(0)\}}\right)^2}$$

Where cd_i is crowding distance for individual i (chromosome i); $f_m(i)$ is the value of objective of i-th individual; d_{ij} is crowding distance of individual i and j.

4.7. Termination criterion

There are several commonly methods such as algorithm repetition, no improvement in the solutions after several repetitions, right implementation time of the algorithm and combination of these methods. In this paper, due to parameters setting in the design of experiments, predetermined iteration at each temperature has been applied for termination from each temperature and achieving to the predetermined temperature for algorithm.

5. Experimental results

There are large numbers of statistical methods to design experiments in order to set parameters of an algorithm. In this paper, Taguchi method is used to determine the most effective factor for the MOSA and the best composition of parameters (Rabbani et al. 2015). Taguchi (1999) improved a family of matrixes of trivial factorial experiments in a way that he could perform an experiment design in order to reduce number of experiments for one problem. According to the literature and initial tests, the levels for parameters of the algorithm including initial temperature (100, 1000, and 10000); number of neighborhoods (1, 3, and 10); population size (10, 20, and 50) and maximum iteration (20, 50, and 100) are considered for Taguchi experiment.

Based on this method, 9 different scenarios is considered. Six instances dissolved and each instance has been run five times under different scenarios. Taguchi method is designed to maximize the performance criterion which is called the signal-to- interference ratio (SNR or S/N). Key variables that affect on the qualitative characteristics of process can be identified by this criterion. SRN is obtained from the equation (12):

$$\mathrm{SRN} = -10log_{10}\left(\frac{\sum {y_i}^2}{n}\right) \qquad (12)$$

where y_i Shows the response variable of the *i*-th observation and n is the number of experiments. In addition to SNR, the Percent relative deviation (PRD) is used which is calculated as follows:

$$PRD = \frac{OF_i - OF_{min}}{OF_{max} - OF_{min}} \qquad (13)$$

where OF_i is the obtained value in the experiment i, OF_{min} is the minimum value of observations in all experiment and OF_{max} is the maximum value of observations in all experiment. After performing a series of experiments, results obtained from all implementations of algorithms by Taguchi design of experiments is presented in Table1 and Figures 2-3.

Table1. Design of appropriate parameters

		level 1	level 2	level 3
S/N	T_0	-59.42885097	-79.56344361	-98.96135545
	Number of Neighborhoods	-69.85129314	-78.77487248	-89.32748442
	Population size	-85.83312623	-79.90639925	-72.21412456
	Maximum iteration	-87.30280456	-78.53011137	-72.12073412
RPD	T_0	0.458305184	0.486480919	0.517678572
	Number of Neighborhoods	0.506928054	0.471666625	0.483869996
	Population size	0.481744383	0.485243286	0.495477006
	Maximum iteration	0.512502051	0.514341866	0.435620757

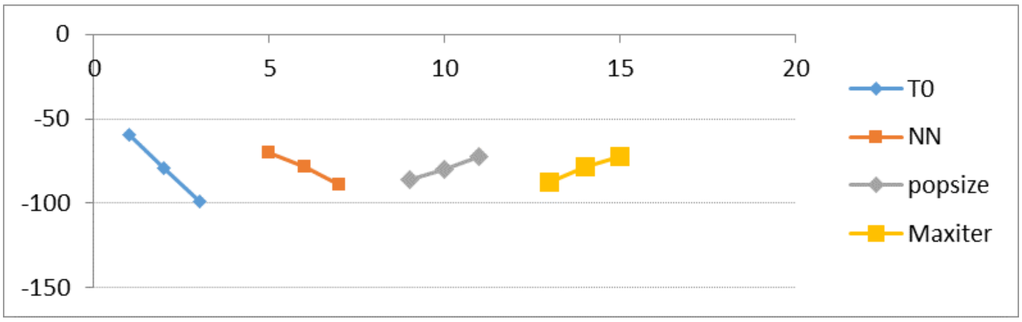


Figure 2. Means plot of S/N for MOSA parameters.

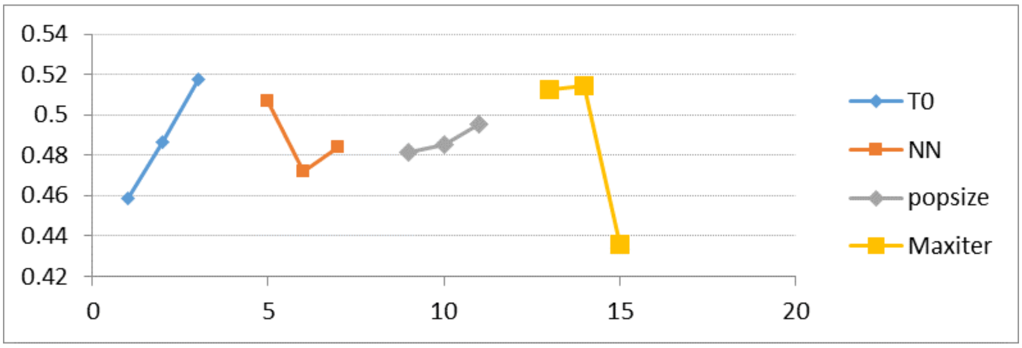


Figure 3. Means plot of RPD for MOSA parameters.

According to the above results, the optimal values for parameters of algorithm is selected as 100 for initial temperature, 3 for number of neighborhoods, 50 for population size and 100 for maximum iteration.

In order to show algorithm efficiency, four problems by different Characteristics have been solved as the example. The experiments have been run on a PC with 2.50 GHz processor and 8.00 GB memory. For coding the MOSA algorithm, Matlab software is used. In order to solve these problems, suitable parameters is chosen using elementary experiments and then the best parameter and combination of parameters and operators are being chosen applying Taguchi method. The results related to solving these four problems have been shown in Tables 2 and 3.

Table2. The first 10s Pareto solutions obtained by MOSA

Problem1			Problem2		
Individual	Objective function values		Individual	Objective function values	
	1	2		1	2
1	406	257	1	440	134
2	435.5	187.5	2	504	117
3	393	285.3	3	501	123
4	422.7	198.1	4	435.5	161.5
5	437.3	153.6	5	450	130
6	381.5	297.5	6	513	104
7	441	135.4	7	507	112
8	410.6	233	8	480.5	126.5
9	452.4	107.9	9	425	182
10	446.2	115.3	10	410	197.5

Table3. The first 10s Pareto solutions obtained by MOSA

Problem3			Problem4		
Individual	Objective function values		Individual	Objective function values	
	1	2		1	2
1	4654	1720.5	1	5764	1662
2	4595	2152.5	2	4743.5	3023
3	5285	1662	3	5254	1871.5
4	4464.5	2312	4	5193	2352
5	4732	1680.5	5	4973.5	2672
6	4420	2380.5	6	5854	1451
7	5320	1441	7	4578	3146
8	5100	1601	8	5960	1270
9	4405	2454.5	9	4436	3283
10	4854.5	1620	10	4852	2820

6. Empirical analysis

In this section the results of the MOSA are compared with MOCS (multi-objective Cuckoo Search). CS is a metaheuristic search algorithm which is recently developed by Yang & Deb (2009). This new algorithm has been shown to be very successful in solving optimization problems. CS is inspired by the obligate brood parasitic behavior of some cuckoo species; combined with Levy flights that describe the foraging patterns taken from many animals and insects.

This algorithm is a good sample of nature inspired metaheuristics. CS algorithm follows the three rules below:
1. The cuckoo lays an egg at any time and put it in a random nest.
2. The best nest with high quality egg creates the next generation.
3. The number of host nests is fixed and the egg laid by cuckoo is discovered by the host bird based on a probability .

CS is also characterized by the reduced number of parameters and provided effective results for multimodal functions in comparison with both genetic algorithms and particle swarm optimization (Yang & Deb, 2010). Multi-objective Cuckoo Search is development of Cuckoo Search for multi-objective problems which has reached to this goal by putting more eggs in a nest rather than one egg. The results of this method compared with other multi-objective algorithms indicate the power of this method in solving these problems.

The best parameters of the algorithm are designated which is shown in Table 4.

Table4. The MOCS algorithm Parameters

Parameter	Value
Population size	100
number of generations	75
Beta	1.5
Discovery rate of alien eggs	0.6
Step size	0.01

6.1. Performance measures

Generally, the performance metrics for multi-objective optimization algorithms concentrate on two important issues:
1. The closeness of solutions to the Pareto-optimal front
2. The diversity of solutions.

This section deals with comparing the capabilities of two provided algorithms according to five criteria which are common in multi-objective optimization algorithms (Rabbani et al. 2016). In this section, five Comparative criteria to measure the capabilities of the multi-objective metaheuristics in terms of NO, SM, DIP, MS and SC are presented. These criteria is explained in detail in the below subsections.

6.1. 1. Number of Pareto solutions (NO)

This criterion shows the number of Pareto solutions obtained from using multi-objective algorithms. The bigger values are preferable.

6.1.2. Spacing metric (SM)

This criterion can help to recognize the monotonous dispersion of solutions in the Pareto curve. This criterion is calculated by using the following equation:

$$S = \sqrt{\frac{1}{|Q|}\sum_{i=1}^{|Q|}(d_i - \bar{d})^2} \quad (14)$$

Where $|Q|$ is the number of elements located in the Pareto curve and d_i and $\bar{d}$ are obtained from Equations 14 and 15:

$$d_i = min_{k\epsilon Q \wedge k\neq i} \sum_{m=1}^{M} \left| f_m^i - f_m^k \right| \quad (15)$$

$$\bar{d} = \sum_{i=1}^{|Q|} \frac{di}{|Q|} \quad (16)$$

At first, d_i has to be computed for each Pareto point. For computing this parameter, the nearest neighbor points are used. Next, value S is calculated, which is the indicator of the spacing metric. Whatever this value is closer to zero, the more spacing metric and the better solution will result.

6.1.3. Distance from the ideal point (DIP)

This criterion is used to measure Proximity to the real Pareto-optimal level; it is calculated from equation 16:

$$DIP = \frac{\sum_{i=1}^{n} c_i}{n} \quad (17)$$

In this equation, n is the number of Pareto solutions and c_i is the euclidean distance of each member of the Pareto front from the ideal point which is obtained from 17:

$$c_i = \sqrt{(f_{1i} - f_1^*)^2 + (f_{2i} - f_2^*)^2 + \cdots + (f_{mi} - f_m^*)^2} \quad (18)$$

In equation 16, f_{mi} is the value of *m*-th objective function in *i*-th solution.

6.1. 4. Most Spread (MS)

This criterion measures the diameter of the spatial cube which is made by end points value of non-dominant solutions in objective space. Therefore, whatever this scale is larger represents a further expansion of the Pareto front solutions.

$$\text{MS} = \sqrt{\sum_{m=1}^{M}\left(max_{i=1:|Q|} f_m^i - min_{i=1:|Q|} f_m^i\right)^2} \tag{19}$$

6.1.5. Set coverage (SC)

Set coverage criterion, $\text{C}(\text{A},\text{B})$, calculates the ratio of solutions of set B which is dominated by solutions of set A.

$$C(A,B) = \frac{|\{b \epsilon B \mid \exists a \epsilon A : a \leq b\}|}{|B|} \tag{20}$$

6.2. Comparing and analysis

This section compares the capabilities of two algorithms relative to each other. Each algorithm was run 20 times (five times for each problem) and the value of each criterion is calculated for each run. Average values of the criteria for each algorithm are shown in Table 5. This results show that the MOSA algorithm relatively well in most criteria and it has been more successful than MOCS algorithm except in set coverage criterion.

Table 5. Comparison of algorithms according to the five criteria

	NO	DIP	SC	SM	MS
MOSA	28	2.56	12.2	0.0800	69.5
MOCS	16	2.82	17.8	0. 3557	48.9

7. Conclusions

In this paper, a two-objective mixed-integer linear programing model (MILP) for a capacitated Windy Rural Postman Problem with several vehicles was designed. Two objectives was considered for this problem: one of them is the minimization of the total cost (of all vehicle routes), expressed by the sum of the total deadheading cost and another one is to minimize the maximum cost of vehicle route in order to find a set of balanced tours for the vehicles. This problem is accounted as NP-hard problems, so multi-objective simulated annealing (MOSA) algorithm was used to solve this problem. Taguchi experiment designing methods was applied for tuning parameters of algorithm. To demonstrate efficiency and effectiveness of the algorithm, several numerical examples was used. To validate the proposed algorithm, we compare the performance of it with the multi-objective Cuckoo search algorithm according to the five criteria which shows significantly better results for the MOSA than those MOCS algorithm.

The following cases can be mentioned as the future researches: Considering the uncertainties in some parameters, such as demand of required edges which increases problem complexity; using other meta-heuristic methods such as Non-dominated Sorting Genetic Algorithm (NSGA-II), multi-objective particle swarm optimization algorithm (MOPSO) and multi-objective Imperialist Competitive Algorithm and comparing results of these algorithms with results obtained from algorithm presented in this paper.

References

Ahr, D., & Reinelt, G. (2006). A Tabu search Algorithm for the Min-Max k-Chinese Postman Problem. *Computers & Operations Research*, 33, 3403-3422.

Bandyopadhyay, S., Saha, S., Maulik, U., & Deb, K. (2008). A simulated annealing based multiobjective optimization algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation*, 12(3), 269–283.

Benavent, E., Corberán, A., Piñana, E., Isaac, P., & Sanchis, J. M. (2005). New heuristic algorithms for the windy rural postman problem. *Computers & Operations Research*, 32, 3111 – 3128.

Benavent, E., Carrotta, A., Corberán, A., Sanchisc, J. M., & Vigo, D. (2007). Lower bounds and heuristics for the Windy Rural Postman Problem. *European Journal of Operational Research*, 176, 855–869.

Benavent, E., Corberán, A., Sanchis, J. M., & Isaac, P. (2009). Min-Max K-vehicles Windy Rural Postman Problem. Wiley Periodicals, Inc. *NETWORKS*, 54(4), 216–226.

Benavent, E., Corberán, A., & Sanchisc, J. M. (2010). A metaheuristic for the min–max windy rural postman problem with K vehicles. *Computational Management Science*, 7(3), 269-287.

Benavent, E., Corberán, A., Isaac, P., & Sanchis, J. M. (2011). New facets and an enhanced branch-and-cut for the min–max *K*-vehicles windy rural postman problem. Wiley Periodicals, Inc. *NETWORKS*, 58(4), 255–272.

Brucker, P. (1980). The Chinese postman problem for mixed graphs. In*Graphtheoretic Concepts in Computer Science* (pp. 354-366). Springer Berlin Heidelberg.

Christofides, N., Campos, V., Corberán, A., & Mota, E. (1986). An algorithm for the rural postman problem on a directed graph. *Mathematical Programming Study*, 26, 155–66.

Corberán, A., Plana, I., & Sanchis, J. M. (2005). The Windy General Routing Polyhedron: A global view of many known Arc Routing Polyhedra. *SIAM Journal on Discrete Mathematics*, 22(2), 606–628.

Corberán, A., Plana, I., & Sanchis, J. M. (2007). A branch & cut algorithm for the windy general routing problem and special cases. Wiley Periodicals, Inc. *NETWORKS*, 49(4), 245–257.

Czyzak, P., & Jaszkiewicz, A. (1998). Pareto simulated annealing – A metaheuristic technique for multipleobjective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7, 34–47.

Dror, M. (Ed.). (2012). *Arc routing: theory, solutions and applications*. Springer Science & Business Media.

Dussault, B., Golden, B., Groer, C., & Wasil, E. (2013). Plowing with precedence: A variant of the windy postman problem. *Computers & Operations Research*, 40, 1047–1059.

Edmonds, J., & Johnson, E. (1973). Matching, Euler Tours and the Chinese Postman Problem. *Mathematical Programming*, 5, 88-124.

Grötschel, M., & Win, Z. (1992). A cutting plane algorithm for the windy postman problem. *Mathematical Programming*, 55, 339–58.

Guan, M. (1962). Graphic programming using odd or even points. *Chinese Mathematics*, 1, 273-277.

Guan, M. (1984). On the windy postman problem. *Discrete Applied Mathematics*, 9(1), 41–6.

Gun, H. (1993). Polyhedral structure and efficient algorithms for certain classes of the directed rural postman problem.

Kirkpatrick, S., Gerlatt, C. D., & Vecchi, N. P. (1983). Optimization by simulated annealing. *Science*, 220, 671–680.

Martínez. F. J. Z. (2008). Series–parallel graphs are windy postman perfect. *Discrete Mathematics*, 308, 1366 – 1374.

Micó, J. C., & Soler, D. (2011). The capacitated general windy routing problem with turn penalties. *Operations Research*, 39, 265–271.

Minieka, E. (1979). The Chinese postman problem for mixed networks. *Management Science*, 25, 643–8.

Orloff, C. (1974). A fundamental problem in vehicle routing. Wiley Periodicals, Inc. *NETWORKS*, 27, 95-108.

Papadimitriou, C. H. (1976). on the complexity of edge traversing. *Journal of the ACM (JACM)*, 23(3), 544-554.

Pearn, W. L., & Li, M. L. (1994). Algorithms for the Windy Postman Problem. *Computers &*

Operations Research, 21, 641–651.

Rabbani, M., Farrokhi-Asl, H., & Ameli, M. (2016). Solving a fuzzy multi-objective products and time planning using hybrid meta-heuristic algorithm: Gas refinery case study. *Uncertain Supply Chain Management*, 4(2), 93-106.

Rabbani, M., Ramezankhani, M. J., Farshbaf-Geranmayeh, A., & Farrokhi-Asl, H. (2015). Vehicle Routing with Time Windows and Customer Selection for Perishable Goods. *International Journal of Supply and Operations Management*, *2*(2), 700-719.

Savall, J. V. (1990). *Polyhedral results and approximate algorithms for the directed rural postman problem* (Doctoral dissertation, PhD dissertation).

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

Mori, T. (1991). *The new experimental design: Taguchi's approach to quality engineering*. Dearborn, Michigan: ASI press.

Win, Z. (1987). *Contributions to routing problems*. PhD dissertation, University of Augsburg, Germany.

Win, Z. (1989). On the windy postman problem on Eulerian graphs. *Mathematical Programming*, 44, 97–112.

Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In*Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on* (pp. 210-214). IEEE.

Yang, X. S., & Deb, S. (2010). Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation*, 1, 330–343.