Modeling a Multi Depot K- Chinese Postman Problem with Consideration of Priorities for Servicing Arcs

Masoud Rabbani[1a*], Setare Mohammadi[1b]

School of Industrial Engineering, College of Engineering
University of Tehran, Tehran, Iran

[a]*mrabani@ut.ac.ir
[b]mohammadi.9009@ut.ac.ir

Abstract: This study deals with the Chinese Postman Problem in critical situation. Critical situation is defined by moving on arcs. One aim of this investigation is visitation of critical edges as soon as possible. The starting points of the study are the high cost of activities, decreased out of priority in servicing arcs, high numbers of vehicles and routes to service. In this respect, the problem was determined as multi depot k-Chinese postman problem, a kind of arc routing problem. This Problem considers three weighted goals. In this study, the optimal number of postman and the optimal path for each of them is determined. This mathematical model was solved by a heuristic algorithm which determines possible routes for each postman then by comparing paths and eliminating expensive ways sets the best route and the best number of postman. For comparison of the current solution, Artificial intelligence algorithms entitled Genetic and Particle swarm were applied.
Keywords: Chinese postman problem; genetic algorithm; graph theory; particle swarm optimization algorithm

1 INTRODUCTION

In 1962, a Chinese mathematician [1] added aspects of the optimization problem. Edmonds [2] offered a polynomial algorithm to solve the Chinese postman problem. Guan and Fleischner [3] showed that for a planar graph the answer of Maximum Weighted Cycle Packing Problem is equal to CPP, so both of them are equivalent. Edmonds and Johnson [4] found that the DCPP (Directed Chinese Postman Problem) can be solved in appropriate time. They offered a polynomial algorithm to solve it. Existence of Euler tour needs even vertices degree of the corresponding graph. So that the for each vertex entrance edges are equal to outgoing edges. Minoux [5] presented a model which was named Optimal Link Test Pattern Problem but couldn't recognize it's relation to Maximum Weighted Cycle Packing Problem. Chinese Postman Problem allows traveling edges in both directions but DCPP allows passing them in just one determined direction. The first exact algorithm based on branch and bound approach for Mixed Chinese Postman Problem was presented by Christofides, et al. [6]. On some of the Chinese postman Problems the highest goal is to minimize the maximum path. This is written in the following form:

$$w_{\max}(C^*) = \min\{w_{\max}(C) \mid C \text{ is a k-postman tour on } E\}$$

The goal is finding a tour C* with k-postman which minimizes $w_{\max}$. Chinese postman problem with simultaneously k different postman is called K-

ISSN: 2222-7059 (print), ISSN: 2222-7067 (online), http://www.aspbs.com/aiem/
DOI: 10.7508/aiem.2015.02.004

Chinese Postman Problem (k-CPP) [1] .The CPP is a particular case of k-CPP. The CPP aims are to minimize the total cost of the tour, while k-CPP with k ≥ 2. The goal is to minimize the total cost of the tour. Economically, the worst form of k-CPP in this case is just one postman who services all other edges. Problems in the real world require employing multiple postmen instead of considering only the postman. Usually, a company isn't only one person to serve in different parts but also serve several uses. Services such as the delivery of mail, distribution services, snow removal services, milk delivery systems, all require several people to service with minimal cost while time is limited. In a study conducted by Pearn [7], general conditions necessary to solve a k-CPP is k-CPP networks completely undirected, k-CPP is fully directed networks, k-CPP in hybrid networks, even and symmetrical, k-CPP Windy networks, along with periodically symmetric. Generally k-CPP is an NP-hard problem [8] . So the ways of solving this problem are meta-heuristic and approximate methods [9]. Sorge [10] in his found that the k-CPP if the parameter (k) is assumed constant would be solvable and Gutin, et al. [11] in their paper prove this.

2 PROBLEM FORMULATIONS

The problem studied in the research is a multi-objective multi-depot k-Chinese postman problem, which is a class of ARC routing problem. The objective of the Chinese postman problem is minimization of the total cost of the arc routing problem by allowing crossing each arc at least one time. Linear programming model of Chinese postman problem is as follows:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij} \qquad (1)$$

$$\sum_{j=1}^{n} X_{ij} - \sum_{j=1}^{n} X_{ji} = 0 \qquad \forall i \in V \qquad (2)$$

$$X_{ij} + X_{ji} \geq 1 \qquad \forall (i, j) \in A \qquad (3)$$

$$X_{ij} \geq 0 \text{ Integer} \qquad \forall i, j \in V \qquad (4)$$

Equation (1) is objective function that minimizes the number of traversed arcs. The equality of the entry and exit numbers of each vertex is expressed in equation (2). Equation (3) points out that each arc must be at least once visited and last equation means all variables must be positive integer numbers [12].

K vehicles or postmen with specified capacity leave the determined depots and visit the arcs. It should be considered that vehicle's tour should not exceed the amount of each vehicle's capacity.

Now, it can be expressed complete explanation of problem. The aim is to minimize distance, minimize early observation of low priority arcs in the investigation and minimize total number of postman. 2 decimal allocated to each of the arcs. One of them is the distance of each arc and the other one the lack of priority. If the noncritical arcs visit earlier the result would be costs increases. If the arcs are duplicated once again expenses will increase. The optimum number of postmen is not known and should be determined after problem solvation.

Parameters and indexes

n	*Depot number*
h	*Selected depot*
v	*vertexes set*
D	*The maximum acceptable distance for each Eulerian path*
w	*The degree of importance of the objective function*
c_{ij}	*The cost of distance from origin i to destination j*
I_{ij}	*The low degree of priority*
$\{M_1, M_2 ..., M_k\}$	*Euler paths set traveled by the postman*
$\{e_1, e_2 ..., e_n\}$	*Depot set*
T	*D/min(C_{ij})*

Variables

K_h	*The number of postman assigned to the node h.*
$X_{M_f ijt}$	*0, 1 variable of traveling arc (ij) in Euler path M_f at step t*

Objective function and restrictions

$$TotalCost = \min \sum_{e_f = e_1}^{e_n} Z_{eh}$$

$$Z_{e_h} = \min(w1 \sum_{M_f=1}^{M_K} \sum_{t=1}^{T} \sum_{j=1}^{V} \sum_{i=1}^{V} C_{ij} X_{M_f ijt} + w2 CostPriority(t) + w3 \sum_{h=1}^{n} K_h)$$

$$CostPer(t) = \sum_{j=1}^{V} I_{ij} X_{M_f hjt} \quad \forall M_f \in \{M_1, M_2 ..., M_K\} \tag{1}$$

$$CostPriority(t) = 2CostPeriority(t-1) + CostPer(t) \tag{2}$$

$$K_h = \alpha_1 + 2\alpha_2 + 3\alpha_3 + ... + b\alpha_b \tag{3}$$

$$\sum_{i=1}^{b} \alpha_i = 1 \tag{4}$$

$$\sum_{j=1}^{V} X_{M_f jit} = \begin{cases} 1 & If \sum_{i=1}^{V} X_{M_f ij(t-1)} = 1 \\ 0 & Otherwise \end{cases} \tag{5}$$

$$\sum_{t=0}^{T} \sum_{j=1}^{V} \sum_{i=1}^{V} X_{M_f tij} C_{ij} \le D \tag{6}$$

$$\sum_{M_f=M_1}^{M_k} \sum_{t=1}^{T} \sum_{j=1}^{V} X_{M_f jit} = \sum_{M_f=M_1}^{M_k} \sum_{t=1}^{T} \sum_{j=1}^{V} X_{M_f ijt} \tag{7}$$

$$\sum_{M_f=1}^{M_k} \sum_{j=1}^{V} X_{M_f hj0} = K_h \tag{8}$$

Equation (1) is the cost of passing noncritical arc at *t*=0. Equation (2) express the value of a recursive cost function of noncritical arcs, so that if the arc of low priority placed first it's noncritical value (I_{ij}) at each step is added to objective function. The number of postman allocated to each depot determined by Equation (3). α is a 0, 1 variable that only one of them is nonzero which is represented by Equation (4). Equation (5) states if in previous step passed an arc which is ended to vertex *j* now this step one arc started with *j* should be chosen. Equation (6) Implies the distance walked per Eulerian path should not be more than D. The number of outside arcs for each vertex is equal to the number of inside arcs which is determined in equation (7). Equation (8) represents the number of arcs exit from depot at the first step is equal to the number of the postman.

2.1 Graph Form of Problem

Small-scale graph form is examined. Graph is plotted in Fig. 1 is small scale drawing of the problem. Vertices are divided into two categories. The Vertices which represent depots and others represent intersections. Note that the depots are located at the intersections. Various depots are plotted in the form of the vertices of different colors. Vertices that are drawn blue color are not the location of depots. In this problem we have 5 depots. For the black and gray depots the number of postman is equal to 2 while in other places just a postman offers service. For ease of display regarding the severity of the critical pathways arcs are drawn distinct colors. The numbers represent the time it takes passing each of the arcs. Each of the arcs must be visited by at least one of the postman. Then two of the proposed solutions of the problem is presented.

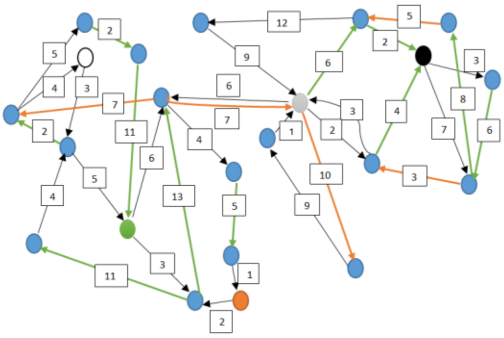

Fig. 1. Small scale of problem

A total of 7 postmen must pass the arcs. Fig. 2 is one of the proposed solutions to the problem. Fig. 3 also offers another solution. All proposed solutions should satisfy the problem constraints. If the capacity of postman is limited they cannot travel any different path lengths so a series of solutions proposed would be unacceptable. To remove such solutions a penalty function is applied in objective function. In this problem the number of the postman at any depot is not fixed. If the arcs are passed several times the cost of the system increases so the postman travels the streets surrounding. Priority routes are those which more critical arcs are passed earlier. The aim is to improve simultaneously both objective functions. A small size of problem along with drawing corresponding graph and the optimal solution represented in Fig. 4 and Table 1.

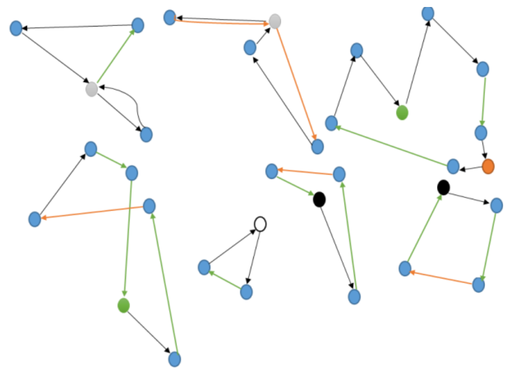

Fig. 2. Proposed solution one

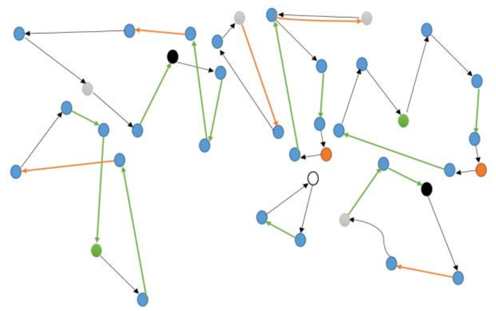

Fig. 3. Proposed solution two

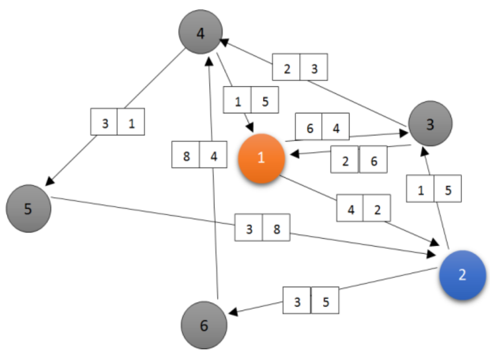

Fig. 4. Graphical form of problem

3 METHODOLOGY

To solve this problem, an optimization algorithm with the approach of Eulerian graphical tours has been developed which is determined in Fig. 5. The problem with the help of Genetic algorithm (GA) and PSO has been conducted so the results have been compared. At the beginning of all solving methods the total number of possible Eulerian paths for each depot counted. The counting has intended Eulerian path length restrictions. At this stage the number of Euler cycles may has determined. In this phase, solving techniques are distinctive which have been described individually. The first solving method results obtained in a longer time seem to be more accurate.

To solve problems in different aspects 10 steps would be followed. *1*. Determine the parameters of the problem. *2*. Specify the number of Eulerian possible cycles for each depot. *3*. Cycles elimination over more than capacity. *4*. Determine the maximum number of postman for each depot demonstrated by K_i with respect to the number of Eulerian possible cycles. *5*. One reduction unit in the total number of the postman. *6*. Collect the problem set solutions. This set contains all possible combinations. If this set is empty, go to Step*9*. *7*. Determine the optimal cycle for each of the postman. *8*. Update the output value of the problem then go to step *5*. *9*. The output value of our problem is reported. Fig. 5 represents the steps taken to achieve optimal solution.

Table 1. Answer of graphical problem

Solution
Depot one: Postman one:
(1,3),(3,4),(4,1)
Depot Two: Postman two:
(2,3),(3,1),(1,2)
Depot Two: Postman three: (2,6),(6,4),(4,5),(5,2)
The total distance walked: 33 Cost1=99 Cost2=131
Total Cost=115

Using Genetic algorithm to solve the problem, we should follow the steps below. *1*. Set parameters of the problem *2*. Determination according to the Eulerian possible cycle paths for each depot. *3*. Omit Eulerian cycles over more than allowable length. *4*. Specification the maximum number of postman for each depot represented by K_i which is equal to the number of Eulerian possible cycles obtained in the previous step. *5*. Allocation of postman set. If $N_1=0$ go to step*6* else the total number of allocated postman is equal to $K_1+K_2+\ldots K_n-N_1$. *6*. Solving optimization problems using PSO algorithms. The solution set at this stage contains combinations emerged with the help of crossover and mutation on population set in addition, the best response in the population. If the answer set was empty at this stage $N_1=N_1-1$ go to Step*5*. *6*. Hold into consideration the best result obtained using Genetic algorithm as the best response. *7*. Omit N_1 Eulerian possible path which

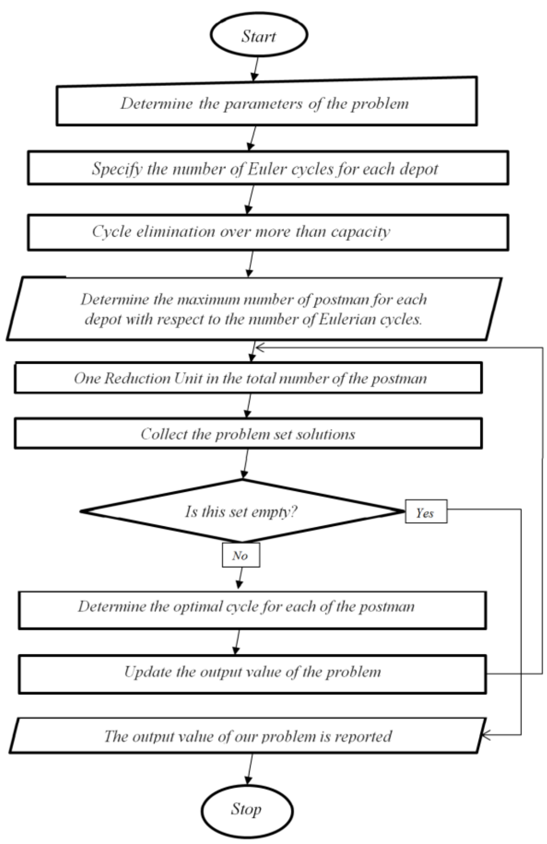

Fig. 5. Represented algorithm

doesn't exist in best result obtained in pervious step from the collection then go to step *4*. *7*. The best Result should be reported.

The first step in using genetic algorithms is creating initial population. The initial population in the form of a chromosome is defined as shown in Fig. 6. Each one represents not selection of a path and each zero represent selection of that path. This chromosome indicates Eulerian paths will be removed from the solution set. The parameters used in genetic algorithm for our problem are number of iterations, population size, crossover probability, and mutation probability. All genetic algorithm codes were written by MATLAB R *2013a* program. Many different ways is possible for processing crossover, at this crossover rate is used. Therefore, as much as the population size, a new solution is created by parent which are selected randomly. Mutation operator is used to avoid local optimum.

1	0	0	1	1	....	0	0	0	1

Fig. 6. An example of GA chromosomes

Fig. 7 and 8 shows Crossover and Mutation operators have been implemented.

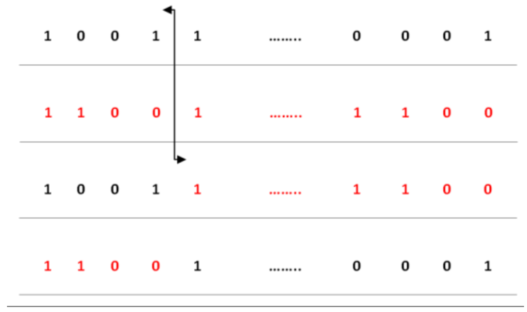

Fig. 7. An example of Crossover operator of GA chromosomes

Fig. 8. An example of Mutation operator of GA chromosomes

Using Particle Swarm Optimization algorithm to solve the problem, we should follow the steps below. Step *1-5* of previous algorithm. *6*. Solving optimization problems using PSO algorithms. If the answer set was empty at this stage $N_1=N_1-1$ go to Step5. 6. Hold into consideration the best result obtained using PSO algorithm as the best response. *7*. Omit N_1 Eulerian possible path which doesn't exist in best result obtained in pervious step from the collection then go to step*4*. *7*. The best Result should be reported.

Various versions of the PSO algorithm are used but at this research the general equation movement of the particles is shown below. All genetic Particle Swarm Optimization codes were written by MATLAB R *2013a* program.

$$v^i[t+1] = wv^i[t] + c_1 r_1 \left(x^{i,best}[t] - x^i[t]\right)$$
$$+ c_2 r_2 \left(x^{gbest}[t] - x^i[t]\right)$$
$$x^i[t+1] = x^i[t] + v^i[t+1]$$

$v^i[t+1]$ is the coefficient of inertia for particle i in next movement. Velocity shows the desire of particle to continue the previous path. $c_1 \,\&\, c_1 \in Unif[1, 2]$, $w \in Unif[0.4, 0.9]$ and $r_1 \,\&\, r_1 \in Unif[0, 1]$. Increase in the amount of w enhances exploration of algorithm and its decreasing, enhances exploitation ability of algorithm.

4 NUMERICAL RESULTS

The problem in different dimensions with the help of the mentioned algorithms has been investigated. Parameter assumed values presented in Table 2.

Arc length and Arcs Priority comes from uniform distribution. The weights considered for each goal is represented in column7. The numerical results are presented in Table 3.The Problem is solved in three dimensions by three solving method and its numerical results are reported in Table 3. As can be observed in Fig. 9 the first the algorithm investigates more solutions so it takes more time and after more iterations it finds the best solution in compression with meta-heuristics so its run time is longer than both of GA and PSO but this algorithm does not fall in local trap so its response can confirm the answers of two other algorithms. Show that the GA and PSO algorithms didn't fall in local trap .In general compression PSO reports better solutions than GA in each step. As can be seen in Fig. 9 after elimination of non-optimal routes all results converge on one hand.

This figure represents the cost results related to a network with 10 vertexes, 2 depots and 20 arcs. The results of genetic and PSO algorithms is near another therefore, the graph in the lower part of the figure is magnified of this area. This figure includes the first two components of cost so the third goal of minimizing total number of postmen is not considered.

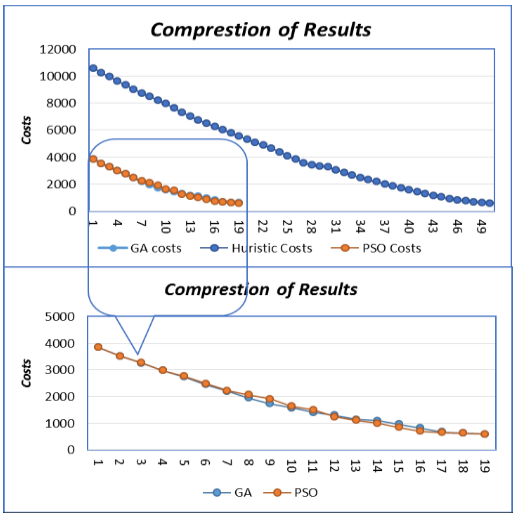

Fig. 9. Comparative results

Numerical results of GA for a small size problem reported by Tables 4, 5 and 6. Table 4 represents the distribution of parameters. Table 5 reports the components of cost and total cost of GA in 12 iterations. Step by step the amount of Total cost decreases till the number of the postman has been minimized. The routs passed by the postmen assigned to depots are observed in Table 6.

Table 2. Parameters amount

Algorithm	*Depots*	*Vertexes*	*Arcs*	*Length*	*Priority*	*Max distance*	*Weights*	*Fuel Cost (C_{ij})*
Heuristic	*2*	*6*	*10*	*unif (1,7)*	*unif(1,10)*	*30*	*0.3,0.3,0.4*	*3*
	2	*20*	*380*	*unif(1,7)*	*unif(1,10)*	*40*	*0.3,0.3,0.4*	*3*
	2	*50*	*1000*	*unif(1,7)*	*unif(1,10)*	*60*	*0.3,0.3,0.4*	*3*
GA	*2*	*6*	*10*	*unif(1,7)*	*unif(1,10)*	*30*	*0.3,0.3,0.4*	*3*
	2	*20*	*380*	*unif(1,7)*	*unif(1,10)*	*40*	*0.3,0.3,0.4*	*3*
	2	*50*	*1000*	*unif(1,7)*	*unif(1,10)*	*60*	*0.3,0.3,0.4*	*3*
PSO	*2*	*6*	*10*	*unif(1,7)*	*unif(1,10)*	*30*	*0.3,0.3,0.4*	*3*
	2	*20*	*380*	*unif(1,7)*	*unif(1,10)*	*40*	*0.3,0.3,0.4*	*3*
	2	*50*	*1000*	*unif(1,7)*	*unif(1,10)*	*60*	*0.3,0.3,0.4*	*3*

Table 3. The results related to Table 2 Parameters

Algorithm	*Cost1*	*Cost2*	*Cost3*	*Total Cost*
Heuristic	*99*	*146*	*3*	*74.7*

Continued in Next Page

	1176	*16194*	*72*	*5239.8*
	3661	*291376*	*113*	*88556.3*
GA	*99*	*146*	*3*	*74.7*
	1232	*16329*	*74*	*5297.9*
	3801	*298819*	*116*	*90832.4*
PSO	*99*	*146*	*3*	*74.7*
	1195	*16276*	*73*	*5270.5*
	3684	*295258*	*115*	*89728.6*

Table 4. Parameters

Parameters	*Number of Depots*	*Number of Vertexes*	*Number of arcs*	*Arc length*	*Arc Priority*	*Maximum distance*	*Cij*
	2	***6***	***10***	***unif(1,7)***	***unif(1,10)***	***17***	***3***

Table 5. Results reported by GA Algorithm

Total Cost GA	***GA Cost1***	***GA Cost2***	***GA Cost3***
650.6	*504*	*1646*	*14*
546.7	*453*	*1352*	*13*
443.1	*405*	*1056*	*12*
339.2	*375*	*741*	*11*
239.8	*327*	*459*	*10*
210.3	*279*	*410*	*9*
174.8	*228*	*344*	*8*
153.1	*204*	*297*	*7*
126.3	*180*	*233*	*6*
109.4	*147*	*211*	*5*
90.1	*114*	*181*	*4*
74.7	*99*	*146*	*3*

Table 6. Routs passed by the postmen

The Postman	***Path Traveled***							
Path Traveled by Postman1	*1*	*3*	*3*	*1*				
Path Traveled by Postman2	*2*	*3*	*3*	*4*	*4*	*5*	*5*	*2*
Path Traveled by Postman3	*2*	*6*	*6*	*4*	*4*	*1*	*1*	*2*

As can be seen even one arc didn't traveled more than one time. Graphical form of Table 5 is represented *in* Fig. 10 and Compressional results of solving methods are represented in Fig. 11.

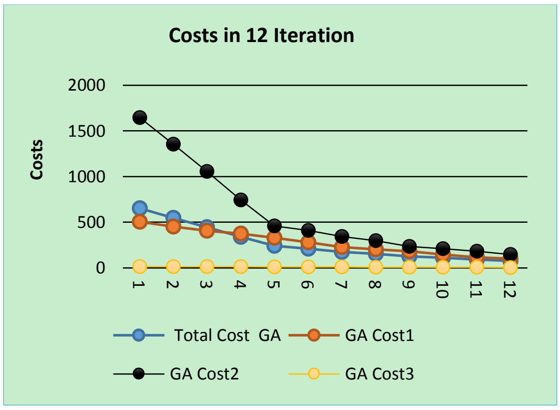

Fig. 10. Components of Cost and Total Cost obtained by GA

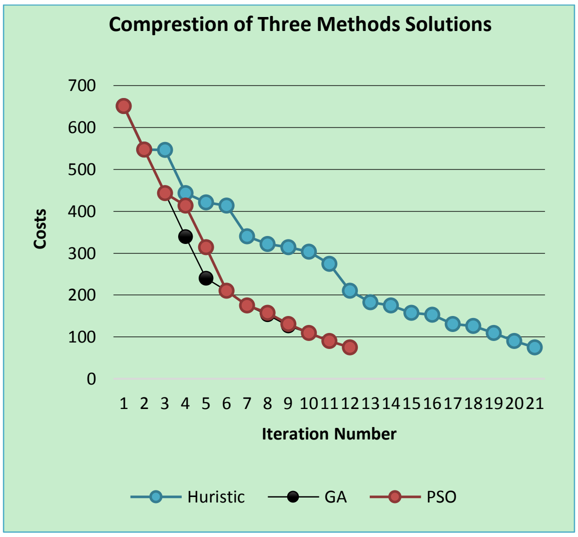

Fig. 11. Components of total cost obtained by GA, PSO and Heuristic Algorithm

5 CONCLUSIONS

This study investigated a multi depot Chinese postman problem which is one of arc routing problems. The first aim is to minimize the mileage by total postmen. The second goal is traveling the critical arcs faster than the rest of the arcs by postmen. The third objective is minimizing the total number of the postmen. The length of path of the survey by the postman is limited. This issue has been evaluated by three separate algorithms to optimize routes were identified. The answers obtained by GA and PSO at each stage are approximately the same but PSO solutions are a little better than GA which is determined in Table 7. If we benchmark the initial algorithm the gap both other algorithms is estimated in Table 7. It is recommended for future research to implement model presented at this study and other relevant models for projects related to emergency management. It is recommended that for power distribution networks by this viewpoint modeling investigation be conducted.

Table 7. Gap of results

	Gap of Compression with Heuristic
GA	*0*
	0.011
	0.025
PSO	*0*
	0.0058
	0.013

References

[1] M. K. Kwan, 1962. Graphic programming using odd or even points. Chinese Math., vol. 1 1, pp. 110.

[2] J. Edmonds, 1965. Paths, trees, and flowers. Canadian Journal of mathematics, vol. 17, no. 3, pp. 449-467.

[3] M. Guan and H. Fleischner, 1983. On the Minimum Weighted Cycle-covering Problem for Planar Graphs. Department of Combinatorics and Optimization, University of Waterloo.

[4] J. Edmonds and E.L. Johnson, 1973. Matching, Euler tours and the Chinese postman. Mathematical programming, vol. 5, no. 1, pp. 88-124.

[5] M. Minoux, 1992. Optimal link test patterns in networks and the Chinese postman problem. Cahiers du Centre d'études de recherche opérationnelle, vol. 34, pp. 139-143.

[6] N. Christofides, et al., 1984. An optimal method for the mixed postman problem. System Modelling and Optimization, vol. 59, pp. 641-649.

[7] W. L. Pearn, 1994. Solvable cases of the k-person Chinese postman problem. Operations Research Letters, vol. 16, no. 4, pp. 241-244.

[8] C. Thomassen, 1997. On the complexity of finding a minimum cycle cover of a graph. SIAM Journal on Computing, vol. 26, no. 3, pp. 675-677.

[9] A. Osterhues and F. Mariak, 2005. On variants of the k-Chinese Postman Problem. Operations Research and Wirtschaftsinformatik, University Dortmund, vol. 30.

[10] M. Sorge, 2013. Some algorithmic challenges in arc routing. Talk at NII Shonan Seminar, no. 18.

[11] G. Gutin, G. Muciaccia and A. Yeo, 2013. Parameterized complexity of k-Chinese postman problem. Theoretical Computer Science,vol 513, pp. 124-128.

[12] İ. Z.Akyurt, T. Keskinturk and Ç. Kalkanci, 2015. Using Genetic Algorithm For Winter Maintenance Operations: Multi Depot K-Chinese Postman Problem. Emerging Markets Journal, vol. 5, no. 1, pp. 50.