



## Heterogeneous Vehicle routing problem by Considering Multi-Path between Customer and Delivery Time Windows with Genetic Algorithm

Mehdi FİROUZİ<sup>1,\*</sup>, Behrooz ABBASI<sup>1</sup>, Abolfazl ROKOOEI<sup>1</sup>

<sup>1</sup>Department of Industrial Engineering, Naragh Branch, Islamic Azad University, Naragh, Iran

Received: 22.03.2015; Accepted: 29.05.2015

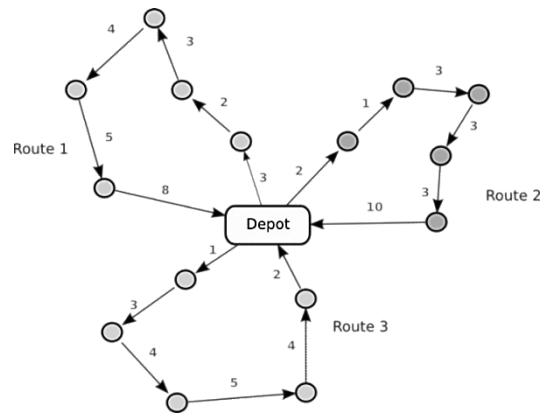
**Abstract.** In this paper vehicle routing problem with heterogeneous vehicle regarding the delivery time windows is examined. In the present case, each customer has a hard time window and a soft time window. Delivery outside the hardware time window is not possible. Delivery outside the soft time window earliness or delay costs imposed on the model. The mathematical model of the problem, innovative genetic algorithm for a given problem is developed. The results obtained from the sample solution by genetic algorithms, the exact solution of the model in the software Lingo is obtained and suggests that genetic provided acceptable performance and low error gives an answer in a reasonable time.

**Keywords:** heterogeneous vehicle routing problem, time window, genetic algorithms

### 1. INTRODUCTION

The basic problem of vehicle routing problem is with capacity constraints. So if the word without mentioning the vehicle routing problem specification is used, it means the routing of vehicles has limited capacity. This graph can be used on  $G = (V, E)$  and defined where  $V$  is a set of  $n + 1$  nodes which indicates the storage node and other nodes represent customers. Regardless of the storage nodes and the set of edges  $E$   $(i, j)$  is the condition  $pp$ . The cost of moving on edge  $(i, j)$  is given by  $c_{ij}$  [1]  $K$  is identical vehicles, each with a weight capacity  $Q$  is depending on the requirements of part or all of it can be used. Each client must be a vehicle service. In other words, discrete delivery is not possible. My client's demand of certain items with a total weight  $D_i$  made a bet  $D_i < Q$ . When the vehicle  $k$  path includes  $S(k) \subset V'$  a set of clients assigned,  $\sum_{i \in S(k)} D_i$  vehicle weight capacity ( $Q$ ) is exceeded. Vehicle routing problem with capacity constraints in the allocation of customers trying to find a subset that does not exceed the number of subsets of  $K$ , for each sub-path from depot to depot beginning and end so that the weight limits are adhered to the total cost of moving the selected edges all the way to the minimum [2] Figure 1 is an example of a solution for vehicle routing problem can be seen by 15 customers in this example; the customer service is performed using three vehicles. Numbers on the arcs can express the distance, cost, or any other similar unit.

\* Corresponding author. Email address: mfirmouzi68@yahoo.com



**Figure 1.** An example of a VRP.

In the case of a heterogeneous fleet, the capacity of the vehicle will be different. As well as delivery time windows is added to the model, it may be any of the devices, in addition to the volume capacity, as well as their capacity constraints. This means that the time taken for a vehicle must not exceed the capacity of the device. In these matters the heterogeneous fleet, the capacity of the vehicles will also differ from each other. Article provided by Dantzig et al [3] The first article is registered in the VRP literature; they studied a large-scale TSP and provide a solution for it. This study was of interest to a lot of other articles. Clarke and Wright [4] for the first time more than one vehicle were considered in formulating the problem, as a result of this study can be considered the first study known VRP literature. The first article of the word "vehicle routing" in their title had been used attributable to Golden et al [5] It should be noted that other versions of the VRP emerged in the early 80's. The 90s, this is the most stable version was developed. The computational complexity and the lack of small computers, accidental releases, dynamic fuzzy and did not receive much attention. Studies VRP, more rapidly expanded during the 1990s. In this period because of compatibility and availability of small computers, sophisticated search methods were innovative methods for solving combinatorial optimization problems such as routing is used. For further study in the field of vehicle routing problem can be presented [6] and [2] Vehicle routing problem in real-world distributed systems is a common problem, where the operating costs of vehicles and crew important component of total costs comprise: So even a small percentage of the total cost savings can be savings substantial. Due to various applications, including the assumptions that have been added to the CVRP can be mentioned the following:

- CVRP with distance limitation. In this case, each arc or edge of an upper limit for the total travels time, and a vehicle application.
- CVRP with time windows: in this case, for each customer is defined as a period of service, the customer must begin in this time period the model presented in this paper are placed in this category.
- CVRP downloads return to this issue, in addition to a set of customers who need to get their goods (linehaul customers), another set of the customer that the goods are sent (backhaul customers).

In every way, all customers who take delivery of goods to customers in advance of the goods deliver the service.

CVRP with large loading area in the back part of CVRP is with simultaneous loading and unloading. These areas are diverse: it is possible for customers to get the goods to be unloaded; it is possible to load goods from one client to another client or return to a central warehouse delivery.

# Heterogeneous Vehicle routing problem by Considering Multi-Path between Customer and Delivery Time Windows with Genetic Algorithm

## 2. Statement of problem

Assumptions follow in demand nodes (cities). All nodes have non-zero final demand and demand must be met. The demand for a product is in other words a single product model. Depot node is zero. Complete graph taken from each node to another is a straight path, the path of reaching and their travel costs and the time and cost that is independent of the path through the machine. Produced in a limited number of available vehicles and each are with a volume capacity and the capacity of a different time. For each client (node) a hard time window and a soft delivery time window are considered [2] It is impossible to satisfy customer's demand outside the hardware and software time window in violation of a late fee or earliness window proportional to the amount of time rape, the model can be imposed. If you have a used car, and it will be immediately removed from the depot to each node after node request promptly met and from the node to the next destination on the way out and back to the depot. All applications provide a car during your trip should not exceed the capacity of the machine and also the time when a car leaving the depot and return to the depot does not exceed the capacity of the machine. Penalties for violation of the software time window for all the cars and customers are considered as equal. The indices  $i$  and  $j$  refer to the node and the index  $k$  is used to refer to the car [1]

Parameters and variables of the model are as follows:

- $n$  The number of nodes (clients)
- $q_k$  Volumetric capacity of vehicle  $k$
- $T_k$  : Vehicle time capacity  $k$
- $K$  The number of machines available
- $c_{ij}$  The cost of traveling from node  $i$  to node  $j$
- $d_i$  The demand node  $i$
- $[e_i, l_i]$ : The top and bottom hard time.  $e_i < l_i$
- $t_{ij}$  Travel time from node  $i$  to node  $j$
- $ta_{ik}$  Time to machine  $k$  to node  $i$
- $[LB_i, UB_i]$ : The top and bottom of the window when soft.  $e_i \leq LB_i$  ,  $UB_i \leq l_i$
- $p_e$  Earliness penalties
- $p_l$  The late penalty
- $ye_{ki}$  The amount of time earliness customer service  $i$   $k$  by car
- $yl_{ki}$  The amount of time of the customer service  $i$   $k$  by car
- $x_{ijk}$  Variable zero and one, if the node  $i$  to node  $j$  directly by machine  $k$ , the direction, value is 1.

The mathematical model is as follows:

$$\min Z = \sum_{i=0}^n \sum_{j=1}^n \sum_{k=1}^K c_{ij} x_{ijk} + \sum_{k=1}^K \sum_{i=1}^n (p_e \cdot ye_{ki} + p_l \cdot yl_{ki}) \quad (1)$$

$$\sum_{i=1}^n \sum_{k=1}^K x_{ijk} = 1, j=1, \dots, n \quad (2)$$

$$\sum_{\substack{i=1 \\ i \neq j}}^n \sum_{k=1}^K x_{jik} = 1, j=1, \dots, n \quad (3)$$

$$\sum_{i=1}^n x_{0ik} \leq 1, k = 1, \dots, K \quad (4)$$

$$\sum_{i=1}^n x_{i0k} \leq 1, k = 1, \dots, K \quad (5)$$

$$\sum_{i=0}^n x_{ijk} - \sum_{i=0}^n x_{jik} = 0, j=0, \dots, n \quad k=1, \dots, K \quad (6)$$

$$\sum_{i=0}^n \sum_{\substack{j=0 \\ i \neq j}}^n d_i x_{ijk} \leq q_k, k = 1, \dots, K \quad (7)$$

$$\sum_{i=0}^n \sum_{\substack{j=0 \\ i \neq j}}^n t_{ij} x_{ijk} \leq T_k, k = 1, \dots, K \quad (8)$$

$$\sum_{h=0}^n x_{0hk} \geq x_{ijk} \quad i, j = 1, \dots, n \quad k = 1, \dots, K \quad (9)$$

$$ta_{0k} = 0, k = 1, \dots, K \quad (10)$$

$$ta_{ik} + t_{ij} - ta_{jk} \leq M(1 - x_{ijk}) \quad (11)$$

$$i = 0, \dots, n \quad j = 1, \dots, n \quad k = 1, \dots, K$$

$$ta_{ik} + t_{ij} - ta_{jk} \geq -M(1 - x_{ijk}) \quad (12)$$

$$LB_i \leq ta_{ik} \quad i = 1, \dots, n \quad k = 1, \dots, K \quad (13)$$

$$UB_i \geq ta_{ik} \quad i = 1, \dots, n \quad k = 1, \dots, K \quad (14)$$

$$ye_{ik} \geq e_i - ta_{ik} \quad i = 1, \dots, n \quad k = 1, \dots, K \quad (15)$$

$$yl_{ik} \geq ta_{ik} - l_i \quad i = 1, \dots, n \quad k = 1, \dots, K \quad (16)$$

$$ta_{ik} \geq 0 \quad x_{ijk} \in \{0, 1\} \quad (17)$$

The first part of the objective function is the total cost of the trip, and the second part of the total cost of earliness and tardiness. Constraints (2) and (3) require that each node and time you visited once and only demand is supplied by a car. Constraints (4) and (5) indicates that the machine if the machine is used, the depot is only one of the nodes can move. And then return to the depot, only one node can be returned to the depot. Constraints (6) maintain the continuity of the flow. Constraints (7) and (8) respectively volume capacity and capacity constraints are time machines. Constraints (9) tours that include a depot not remove it. Constraints (10) require if you used the machine, immediately began to move the depot. The constraints (11) and (12) show that the arrival time of each client machine is equal to the arrival time plus the duration of the trip between the client machines to a previous customer. Four restrictions on the windows, and the last time constraints related to the definition of variables to the model.

1 Presents genetic algorithms to solve the given problem and methods and algorithms of optimization algorithms into two categories: exact and approximate algorithms are classified.

## Heterogeneous Vehicle routing problem by Considering Multi-Path between Customer and Delivery Time Windows with Genetic Algorithm

Exact algorithms to find optimal solutions to difficult optimization problems are accurate but not performance the time resolution of the problem increases exponentially. Approximate algorithms find good solutions (near optimal) in a short time for hard optimization problems. Approximation algorithms usually are classified into two heuristic algorithms (Heuristic) and innovative (Metaheuristic) [7] The approach is innovative algorithm for finding near optimal solutions based on acceptable cost, feasibility or optimality without guarantee is used [1] Identify innovative ways to conduct a limited search space and generally good quality solutions in reasonable computational times produce in addition, most of which can easily be varied in terms of the real world, there are limitations that is why these methods are widely used in commercial packages [8] These procedures are impossible or difficult to make it easier to solve optimization problems and responses can be used to determine the approximate location close to optimal. The main problem with heuristic algorithms, placing them at local, and their inability use the various issues. Metaheuristic algorithms are presented for solving these problems. Metaheuristic in fact, is one of the approximate optimization algorithms that are out of local optimum solutions are applicable to a wide range of issues [7] In recent years, the word Metaheuristic to all modern and high-level algorithms, such as genetic algorithms in which the (GA), gradual freezing algorithm (SA), tabu search (TS), Ant colony optimization algorithms (ACO), particle swarm optimization (PSO), Bees Algorithm (BA), Firefly Algorithm (FA) algorithm and harmony search (HS) are pointed out [10] The basic idea of using genetic algorithms was introduced first by Holland [9] Genetic algorithms, evolutionary algorithms to handle the basics process of inheritance are for solving large combinatorial optimization problems have been successfully employed. In this way, the answer to this question as the initial population and the parents are selected for the next generation. The next generation of crossover and mutation operators can be done. Replaces the previous generation and the new generation as it continues reach a certain number of generations. To solve a problem using genetic algorithms, several issues must be considered:

- 1) Use a suitable way to express each answer in the answer space.
- 2) Use of a fitness function to express the quality of each answer studied
- 3) Defines the proper operation of the structure of matter
- 4) Using a suitable method to replace the current population and new, different ways of using this algorithm has been used in the literature.  
For example, new people replace the current population, or one of the children, parents and other alternatives to replace one of the worst of the crowds.
- 5) Definition of stopping criteria: reach a certain number of repetitions or reach a certain number of consecutive iterations without improvement or converging population having defined the general concepts, the proposed algorithm based on genetic algorithms is presented in this section are as follows:

### Display results:

The answer is contained within a chromosome. In each chromosome,  $v_n$  represents the number of machines used. Array  $v$  indicates that the net results of which machine are used. Tour  $\{i\}$  indicates that the net  $i$ , in order to meet customer. Node 1 is intended as a depot. For example, it assumes that a given problem has 7 nodes (including depot). Sol solution is as follows:

```
sol =  
  vn: 2  
  v: [3 1]  
  tour: {[1 4 3 7] [1 6 2 5 1]}
```

This means that the solution sol of the machine is used. The first tour of the car 3 and the second tour of the machine 1 are used. In the first tour, the depot to the client machine, 3, 4 and then 3 and then the customer to customer demands to provide 7 moves and then returned to the depot.

The second car also hit the second tour, from depot to customer 6 and then moved to 2 and then 5 and then returned to the depot.

The initial population is generated:

Before the specified algorithm (genetic algorithms parameter is the number of initial population Taguchi test was considered the best value for the 100), the initial response is generated. In making any answer the volume capacity constraints and time machines is followed. First, choose a car accident and the depot to the client node moves to the accident. The volumetric capacity of the nodes and time allows a node is randomly selected, and the car goes to the node (in the capacity of a node when the time machine back to the depot also is considered);

If you do not meet the existing demand, a car accident were selected of the other remaining machines and used in the above process is repeated for the car. It is noteworthy that in making the initial response to the condition each node (except depot) should meet only once. If all machines are used for supply and demand however, the volumetric capacity of the selected machine that does not apply to the supply (in this case it is impossible) or if all the machines are used however, when the capacity of the selected machine that does not apply to the supply (in this case the answer may be achieved by changing touring cars) Volumetric capacity constraints, and when the last car is overlooked however, these violations are penalized in calculating the objective function value is added to the objective function. At all stages of the genetic algorithm, the compliance condition of each city can be passed only once. First Answer, is made if it is possible to limit the volume capacity of the machine is not rape.

### **Calculation of fitness**

After making initial population, the fitness of each of the solutions is calculated. The objective function is calculated for each answer. The calculated value of the objective function, except for travel costs, costs exceed the capacity of the volume and time machines as well as fine objective function are added. If the answer is zero, the cost of the intersection or mutations may be violating capacity restrictions and the penalties that makes it, is infeasible solutions with objective function value is far greater. The individual values of the response function of the maximum of the objective function (objective function value is the worst solution) is low. The values obtained are the more, the better answer and the division of each of the values obtained on fitness solutions.

Naturally, the overall fitness of the population is equal to 1. Then the solutions are sorted in descending order according to the fitness value so the answer would be the best answer. Due to the fitness values of the parents candidate elected to the intersection.

### **Parental choice**

The number of half the population and at the intersection of each pair is selected for intersection pair produced two children. Parents are selected according to the fitness and pool their way coupling. The cumulative value of the fitness function of the population (the population is already sorted) is calculated. The random number between zero and one is selected and the first thing that cumulative fitness random number is greater than or equal, first elected as a parent. Similarly, the second parent is selected. After filling the pool coupling, the intersection operation is performed on each pair of parents.

### **Intersection operator**

The parents of two children per couple are produced. Parent couples who have been elected to the intersection, machines that are used exclusively in one of the parents, and the net will appear in both children. And machines that are used in both parents, the chance to tour a parent of a

## Heterogeneous Vehicle routing problem by Considering Multi-Path between Customer and Delivery Time Windows with Genetic Algorithm

child in the child's other parent no longer appear and the net will appear. Then each child in cities where there is several tours, the accident remains one of the tours and tours of the rest are deleted. The tours are only included depot (such tours may remain after removing duplicates cities) of the children were removed. At each intersection of the two parents, two children are produced.

### **Performance of Mutation**

Performance of mutation is performed on children. Performance of mutation so that is a predetermined probability (the probability according to Taguchi test results equal to 0.1 is considered), answers mutate-up. Jump so that accidentally or two nodes of the selected tour and return to their place in Tours if the machine is not used or is in response to a current vehicle accident with a car is not used, but instead remains constant touring. The attention that may result from the intersection or jump, answer machines violate the limits of capacity, due to the violation of these restrictions, the objective function is considered fine and the objective function is added. The penalty, if the parameters of the problem are feasible, if the answer to space out genetic algorithms will try to answer back atmosphere. Because the objective function value is less for an answer, the answer will be calculated for fitness and as I said earlier, the choices of a pair of parents who have done their intersection affect the propriety. Thus at the end of this stage, the exact number of the population is a new child produced. The children are added to the previous population.

### **Step add new answers**

To make the algorithm does not suffer from premature convergence of genetic algorithms has been improved. That's 10% of the population; a new solution is generated and added to the current population and number of children. At the end of this stage, we have doubled the number of initial population (current population and children) plus 10% of the population, we will have the answer.

### **Eliminate inefficient solutions**

The elegance and answers regarding the suitability account will be sorted in descending order. The initial population, the best solution is to keep and delete the rest.

### **Stop condition algorithm**

The requirement is intended to stop the algorithm. If you reach a certain number of generations (according to test results Taguchi the number 100 is intended generation) according to algorithm stops. Also, the population converge solution of the algorithm stops. The crossover, mutation, adding new solutions and eliminate inefficient solutions continues until one of the stopping criteria.

### **1.1. Numerical results**

6 questions are made small by comparison between Lingo and genetics on these examples is done. Examples of these parameters are made uniform discrete distribution. Lingo 8 and to run MATLAB 2013 Dell Inspiron 3 core operating system windows 7, 32 bit is used. The results are listed in Table 1.

**Table 1.** Results of Lingo and genetics.

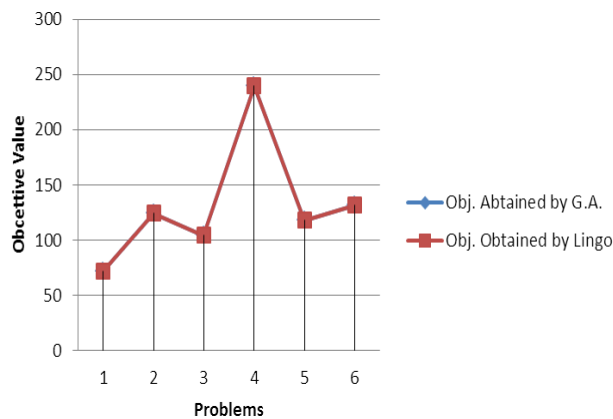
Name of problem	Number of nodes	Number of Cars	Optimum by Lingo	Lingo execution time (seconds)	Answer Genetics	Genetic execution time (seconds)	Genetic errors (percent)
pn5	5	3	72	1	72	0.87	0
pn6	6	3	124.1	5	124.1	1.66	0
pn7	7	4	104.4	8	104.4	1.92	0
pn7-2	7	5	239.6	7	239.6	2.24	0
pn8	8	5	118	241	118	2.68	0
pn9	9	5	131.3	1475	131.3	4.35	0

The error is calculated as follows:

$$err = \frac{genetic\ result - optimum}{optimum}$$

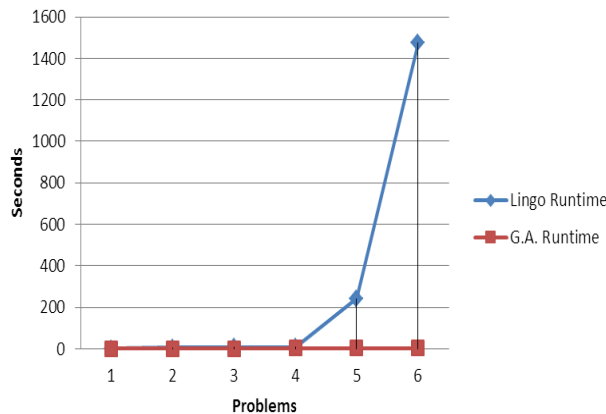
The number of nodes includes the depot. For each problem, genetic algorithms were run 10 times and the results are listed in Table. Meta-heuristic algorithms are not optimal answer and the answer is close to the optimal solution and the low tolerance with respect to time to answer is negligible, but here we are with excellent design innovative genetic algorithm have been able to examples of small sizes, the algorithm error is zero.

A graph of the function obtained by Lingo and genetics is visible in Figure 3.



**Figure 3.** A graph of the function obtained by Lingo and genetics.

It can be seen that error because the genetic algorithm to 6 above zero, Lingo objective function values coincide and genetics and the blue line is not seen. Lingo and genetic execution time diagram in Figure 4 is visible.



**Figure 4.** Lingo and genetic execution time diagram.



## Heterogeneous Vehicle routing problem by Considering Multi-Path between Customer and Delivery Time Windows with Genetic Algorithm

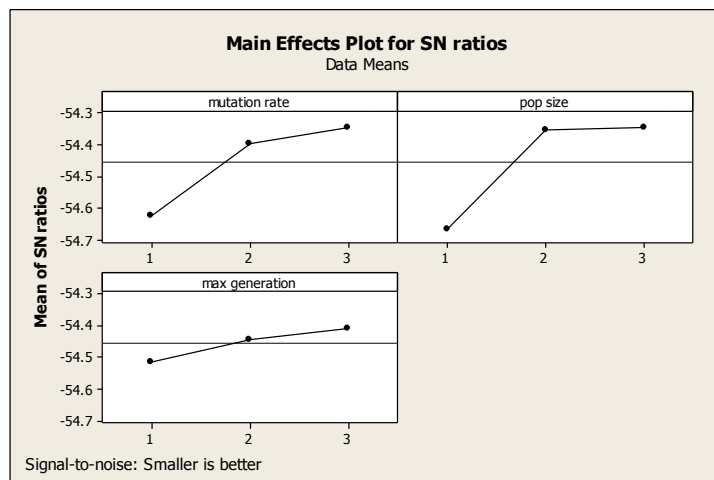
As can be seen, by increasing the scale of the problem, Lingo has dramatically increased since the implementation of the characteristics of the problem is NP-Complete. The complexity of these issues is polynomials of degree higher. So it is effective to solve large scale problems, the exact solution is to try and better heuristic or meta-heuristic algorithms use acceptable and near optimal solution in a reasonable time to lose. According to the model presented in this paper, we design a genetic algorithm. As you can see the issue of genetically solution increases linearly with very little slopes.

### 1.1. Taguchi test

Taguchi test determine genetic algorithms is used. In this case, for each one of the compounds, the algorithm runs 10 times and the results averaged. In this case, genetic algorithms on a large scale example with 20 nodes and 15 machines have been performed. The test for the mutation rate is the number of initial population and the number of generations, 3. For each value is considered as follows:

Mutation rate	Total Population	Number of generations
0.01	50	25
0.05	75	50
0.1	100	100

Test results can be seen below in Figure 2.2:



**Figure 5.** Test results Taguchi.

As the charts Taguchi test signal is specified, the values of mutation rate is equal to 0.1, the number for Population is 100 and the maximum number of generations is equal to 100.

### Conclusions and suggestions for future research

In this paper the mathematical model of vehicle routing problem of heterogeneous hardware and software time window is presented, then, an algorithm is presented based on genetic algorithms for the design was innovative. Computational results show that genetic algorithms presented in this paper has good performance and a reasonable time to provide satisfactory answers. For future research in this area, time of loading and unloading of vehicles can be considered in the model. The vehicles will be allowed on the node before the delivery time to wait to come down hard time windows or after delivery could wait until the next node down hard when the window is not violated. You can increase the number depot. You can also come with innovative algorithms to solve this issue and examined the performance of the algorithms.

**REFERENCES**

- [1] Reeves, C.R. Modern heuristic techniques for combinatorial problems, John Wiley & Sons, Inc ,1993
- [2] Kumar, S. N. and Panneerselvam, R., "A Survey on the Vehicle Routing Problem and Its Variants", *Intelligent Information Management*, Vol. 4, pp. 66-74. 2012
- [3] Dantzig, g., fulkerson, r. and johnson, s., "Solution of a large-scale traveling-salesman problem", *Journal of the operations research society of America*, Vol. pp. 393-410. 1954
- [4] Clarke, G. and Wright, J. W., "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", *Operations Research*, Vol. 12, pp. 568-581. 1964
- [5] Golden, B. L., Magnanti, T. L. and Nguyen, H. Q., "Implementing vehicle routing algorithms", *Networks*, Vol. 7, pp. 113-148. 1977
- [6] Toth, P. and Vigo, D. , *The vehicle routing problem*, Siam, Philadelphia. 2002
- [7] Yaghini, Kazem Zadeh, M., *meta-heuristic optimization algorithms*, SID Amirkabir University of Technology, 2011.
- [8] Eshghi, A; Karimi, M., *combinatorial optimization and meta-heuristic algorithms*, igneous October, 2012.
- [9] Holland, J. H., *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, U Michigan Press ,1975
- [10] Nasri, M. Nasr, H., *harmony search meta-heuristic algorithm*, bell, 2011.