A Comparison of NSGA II and MOSA for Solving Multi-depots Time-dependent Vehicle Routing Problem with Heterogeneous Fleet

Behrouz Afshar-Nadjafi[a,*], Arian Razmi-Farooji[b]

[a] *Assistant Professor, Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*

[b] *MSc, Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*

Received 20 July, 2014; Revised 31 August, 2014; Accepted 20 September, 2014

Abstract

Time-dependent Vehicle Routing Problem is one of the most applicable but least-studied variants of routing and scheduling problems. In this paper, a novel mathematical formulation of time-dependent vehicle routing problems with heterogeneous fleet, hard time widows and multiple depots, is proposed. To deal with the traffic congestions, we also considered that the vehicles are not forced to come back to the depots, from which they were departed. In order to solve our bi-objective formulation, we presented two well-known Meta-heuristic algorithms, namely NSGA II and MOSA and compared their performance based on a set of randomly generated test problems. The results confirm that our MILP model is valid and both NSGA II and MOSA work properly. While NSGA II finds closer solutions to the true Pareto front, MOSA finds evenly- distributed solutions which allows the algorithm to search the space more diversely.

Keywords: Time-dependent Vehicle Routing Problem, Bi-objective optimization, Meta-heuristics, NSGA II, MOSA.

1. Introduction

Transportation is one of the most vital parts of today's tough economy. Transportation facilitates the mobility of goods and people; hence, it plays a central role in economic infrastructures of a country. One of the novel topics in transportation science is Vehicle Routing Problem (VRP). Firstly introduced by Dantzig and Ramser (1959), classic VRP tries to find a set of optimal routes, which starts from a depot, visiting some customers, and finishing to that same depot.
According to Toth and Vigo (2002), VRP can reduce up to 25% of transportation costs. This reason pushed scientists to work on different versions of VRP, each focusing on one industrial and economical need. One of the most significant but under-researched variants of VRP, is Time-dependent Vehicle Routing Problem (TDVRP), The cornerstone of TDVRP is this fact, that the travel speed of a vehicle is not constant, and due to reasons such as traffic congestions, bad weather conditions or road accidents, vehicles' speed can vary over time.
TDVRP was first proposed by Malandraki (1989) and Malandraki and Daskin (1992). This problem was formulated, using a mixed-integer linear programming model. Malandraki and Daskin (1992) used a greedy search method, which was based on travel times between nearest neighbor customers and a branch and cut algorithms, the drawback of Malandraki and Daskin´s work (1992) was loss of First In-First Out (FIFO) property. Based on this property, when a vehicle leaves customer i to serve customer j at time t, any identical vehicle, which leaves customer i to serve customer j at $t+\varepsilon$, where $\varepsilon \succ 0$, will arrive later.
The above-mentioned drawback of Malandraki and Daskin´s work (1989) was modified by Ichoua and et al. (2003). They formulated a TDVRP model with soft time-windows and a Tabu-search based solution approach, which was implemented on Solomon test problems. To modify the FIFO concept, they assumed the time distribution as a piecewise linear function, from which the speed distribution can be derived,
Fleischman et al. (2004) proposed an incapacitated vehicle routing problem with and without presence of time-windows; and to solve this formulation they suggested a route construction procedure based on insertions and a 2-opt local search mechanism. Van Woensel et al. (2008) presented a Tabu-search solution approach for solving a capacitated time-dependent vehicle routing problem. The innovation of their work was considering a queuing theory-based assumption for determining the travel speeds over time. They finally tested their approach on a set of test problems with 32 to 80 customers.
Kok (2010) formulated a TDVRP mathematical model and solved this formulation by optimizing the departure

* Corresponding author Email address: afsharnb@alum.sharif.edu

times and considering driver breaks. Finally, he also considered driving hours´ regulations and managed to obtain feasible solutions. Figliozzi (2012) proposed an Iterative Route Construction and improvement (IRCI) method for dealing with TDVRP with hard time-windows. Testing his solution approach on a set of well-known Solomon benchmark problems, he reported competitive results. Despite all researches conducted on VRP, multi-objective VRPs did not obtain the researchers' interests. There are three different approaches to solving a multi-criteria optimization problem: scalar methods, Pareto methods like NSGA II and MOSA and those, which do not belong to either of the aforementioned techniques. In multi-objective VRPs, the Pareto approach within an evolutionary framework is more common.

Ulungu et al. (1999) employed a MOSA-based algorithm to solve a particular type of routing problems. In their proposed problem, a customer asks to load a quantity at one place and to transport it to another place. The goal of defining this problem was to determine the daily routes of a fleet of trucks, where satisfying a set of customers was mandatory. Żelazny (2012) employed a Memetic algorithm to tackle multi-objective VRP. His proposed genetic algorithm was able to use a local search procedure on non-dominated solutions in order to further improve the Pareto frontier.

Jagiłło and Żelazny (2013) presented a parallel Tabu Search (TS) algorithm on Graphics processing Units (GPUs) to solve multi-criteria discrete optimization of Distance-Constrained VRP. Based on their computational results, they reported that Parallel Tabu Search on GPUs outperforms the classic Tabu Search Method.

To conclude, in spite of the undeniable importance of vehicle routing problem and its variants in both manufacturing and service industry and all of the studies conducted on VRPs, a problem which contains nearly all of constraints we encounter in our real-world problems is still missing in the current literature. In this research, we consider time-varying travel speeds, hard time-windows, multiple depots, heterogeneous fleet and intra depot routes. Moreover, to solve our bi-objective formulation, we employ two well-known multi-criteria meta-heuristics; namely, Non-dominated Sorting Genetic Algorithm II (NSGA II) and Multi Objective Simulated Annealing (MOSA). To the best of our knowledge, no paper has been published yet, covering all the above-mentioned constraints and solving meta-heuristics simultaneously.

In section 2, we define and formulate our bi-objective Mixed Integer Linear Programming (MILP) mathematical model. In section 3, we propose two meta-heuristics; namely, NSGA II and MOSA to solve our model. Section 4 deals with the computational results and discussions. Finally, section 5 concludes this paper.

2. Problem Definition

The Time-dependent Vehicle Routing Problem with hard time-windows, heterogeneous fleet and multiple depots (TDVRPTWHFM) studied in this article is defined as follows: $G\ (N, A)$ is a directed graph, where N is the node set and A is the arc set. The node set N consists of two other subsets, $W = \{1,2,\dots,m\}$, the depots' set and $V = \{m+1,\dots,n\}$, which is the customers' set. The arcs' set A is also defined as $A = \{(i,j): i \neq j\ \wedge i,j \in N\}$. Each customer has a fixed and known demand $q_i \geq 0$, it is also assumed that depots have no demands ($q_i = 0\ \ \forall i \in W$). Serving customers should take place in their time-windows $[a_i, b_i]$, and each customer has a service time $s_i \geq 0$. $F = \{1,\dots,k\}$ is the set of heterogeneous fleets' types. It is assumed that there are unlimited vehicles from each type, each having a capacity C_k, fixed cost cf_k and variable cost cv_k. Each arc from i to j has a fixed distance d_{ij}. However, the cost of travelling this distance is different for each vehicle type and equals $c_{ij}^k = cv_k \text{ x } d_{ij}$. Travelling each arc from i to j has a travel time t_{ij}^k, which depends on the travelling start time (leaving node i). Planning occurs daily and each day is divided into several intervals $u \in I, I = \{1,\dots,U\}$, which means due to daily traffic congestion, travelling from i to j has no fixed travelling time and it is dependent on the travel starting time and the day interval, in which the travel took place.

This problem involves the minimization of the following objectives in order of priority:

1. Number of vehicles from each type
2. Total costs, including travelling costs and vehicle costs

It is worth noting that these two objective functions are contradictory. It means that, having more vehicles and in other words more routes leads to higher total travel costs. There are two decision variables: x_{ij}^{ku} is a binary decision variable, which indicates whether vehicle k, travels from node i to j in time interval u. The real number y_i^k is also used to indicate when the vehicle k arrives to customer i.

The primary and secondary objective functions are defined by (1) and (2), respectively. The constraints are as follows: all customers must be served (3) and (4);

$$min \sum_{k=1}^{k} \sum_{i=1}^{m} \sum_{j=m+1}^{n} \sum_{u=1}^{U} x_{ij}^{ku} \tag{1}$$

$$min \sum_{k=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{u=1}^{U} c_{ij}^{k}\ x_{ij}^{ku} + \sum_{k=1}^{k} \sum_{i=1}^{m} \sum_{j=m+1}^{n} \sum_{u=1}^{U} c_{f_k} x_{ij}^{ku} \tag{2}$$

$$\sum_{k=1}^{k}\sum_{i=1}^{n}\sum_{u=1}^{U} x_{ij}^{ku} = 1 \qquad \forall\, j = m+1,\dots,n \tag{3}$$

$$\sum_{k=1}^{k}\sum_{j=1}^{n}\sum_{u=1}^{U} x_{ij}^{ku} = 1 \qquad \forall\, i = m+1,\dots,n \tag{4}$$

$$\sum_{i=1}^{m}\sum_{j=m+1}^{n}\sum_{u=1}^{U} x_{ij}^{ku} \le 1 \qquad \forall\, k = 1,\dots,k \tag{5}$$

$$\sum_{i=m+1}^{n}\sum_{j=1}^{m}\sum_{u=1}^{U} x_{ij}^{ku} \le 1 \qquad \forall\, k = 1,\dots,k \tag{6}$$

$$\sum_{i=1}^{n}\sum_{u=1}^{U} x_{ij}^{ku} - \sum_{j=1}^{n}\sum_{u=1}^{U} x_{rj}^{ku} = 0 \qquad \forall\, k = 1,\dots,k \qquad \forall\, r = m+1,\dots,n \tag{7}$$

$$a_i \sum_{j=1}^{n} x_{ij}^{ku} \le y_i^k \le b_i \sum_{j=1}^{n} x_{ij}^{ku} \qquad \forall\, k = 1,\dots,k \qquad \forall\, u = 1,\dots,U \tag{8}$$

$$y_i^k + s_i + t_{ij}^k\left(y_i^k + s_i\right) - y_j^k \le M(1 - x_{ij}^{ku}) \qquad \forall\, k = 1,\dots,k \qquad \forall\, u = 1,\dots,U \tag{9}$$

$$y_i^k + s_i - T_u \le M(1 - x_{ij}^{k}) \qquad \forall\, k = 1,\dots,k \qquad \forall\, u = 1,\dots,U \quad \forall\, i,j \in A \tag{10}$$

$$y_i^k + s_i \ge T_{u-1}\, x_{ij}^{ku} \qquad \forall\, k = 1,\dots,k \qquad \forall\, u = 1,\dots,U \quad \forall\, i,j \in A \tag{11}$$

$$\sum_{i=1}^{m} q_i \sum_{j=1}^{n}\sum_{u=1}^{U} x_{ij}^{ku} \le C_k \qquad \forall\, k = 1,\dots,k \tag{12}$$

each vehicle should leave one of the depots in all time intervals at most once (5); each vehicle should arrive at one of the depots in all time intervals at most once (6); when a vehicle arrives to a customer, it should also leave that customer (7); customers should be served in their associated Time-windows (8); service start time must allow for travel time between customers (9); the connection between departure time $(y_i^k + s_i)$ with the time interval, so that the appropriate piece of the travelling time is employed (10) and (11). These constraints, which were introduced by Balseiro et al. (2011), guarantee if a vehicle leaves node i to reach node j in second interval, it will start its next route in second or third interval, because time never goes back. Vehicle's capacity should not be violated (12).

3. Solution Approach

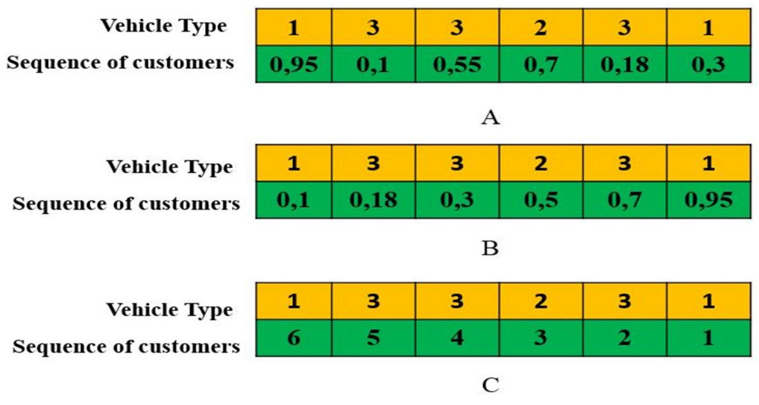
Toth and Vigo (2002) explain that the classic Vehicle Routing problem is derived from the famous Travelling Salesman Problem (TSP). Since the TSP is an NP-Hard combinatorial optimization problem, VRP and its variants belong to this group of problems. Belonging to NP-hard problems, finding exact solutions for TDVRPTWHFM is not only time-consuming, but also impossible in cases that the size of problems get bigger. So, to solve our bi-objective formulation, we decided to employ two well-known multi-objective meta-heuristic approaches; namely, Non-dominated Sorting Genetic Algorithm II (NSGA II) and Multi-objective simulated Annealing (MOSA).

3.1 The NSGA II Approach

Srinivas and Deb (1995) proposed a Genetic Algorithm based method to solve multi-criteria optimization problems, called Non-dominated Sorting Genetic Algorithm (NSGA). This algorithm could find several Pareto solutions in each iteration. However, the algorithm functioned in a very complex way and it lacked elitism selection procedures. These drawbacks made Deb et al. (2002) to modify the previous algorithm and propose its second version; namely, Non-dominated Sorting Genetic Algorithm II (NSGA II). They also added Crowded Distance to their algorithm, a new feature which leads the searching process and chooses the solutions from the most crowded regions, i.e. regions with more densely populated solutions. Followings are the different parts of our algorithm design.

3.1.1 Solutions' Representations

The goal of solving TDVRPTWHFM is assigning vehicles to the customers, with regard to less total travel costs and fewer number of routes. To assign vehicles to the customers, we employed a two-string chromosome, in which the first string is vehicles assigned to the customers and the second string represents the available customers.
As mentioned earlier in Part 2, there are k vehicle types available in each depot, so the probability density function of selecting vehicles is uniform. Accordingly, in the first string a number between 1 and k shows the corresponding vehicle type assigned to that customer. In the second string, a randomly generated number between 0 and 1 depicts the sequence of servicing the customers. Sequencing the customers first happens in discrete space and by employing Smallest Position Value (Tasgetiren et al. 2009) it will then be transformed to the continuous space. Figure 1 shows the above-mentioned procedures.



Vehicle Type	1	3	3	2	3	1
Sequence of customers	0,95	0,1	0,55	0,7	0,18	0,3

A

Vehicle Type	1	3	3	2	3	1
Sequence of customers	0,1	0,18	0,3	0,5	0,7	0,95

B

Vehicle Type	1	3	3	2	3	1
Sequence of customers	6	5	4	3	2	1

C

Fig 1-A. The proposed two-string chromosome before sorting the randomly generated the numbers in the second string
Fig 1-B. The proposed two-string chromosome after sorting the randomly generated the numbers in the second string
Fig 1-C. Proposed two-string chromosome with the sequenced customers

3.1.2 Initial Population

The first step in Genetic Algorithm is forming the initial population. There are two different ways to form the initial population: seeding method, which lets genetic algorithm search in space, where solutions are more likely or generating the initial population randomly. In this research, we formed the initial population by generating two-string chromosomes corresponding to the population size (PopSize).

3.1.3 Parents' Selection Mechanism

After forming the first generation, some of the individuals must be selected to form the next generation. Because of our bi-objective formulation, the selection criteria are based on the followings:

1) Location of solutions in a Pareto front: Lower Pareto fronts are superior to the upper ones.
2) Crowded Distance: If there are two solutions in the same lower Pareto front, one with the least crowded distance, i.e. solution located in a less sparse region of the front, will be selected.

3.1.4 Reproduction

After selecting the parents, the new generation should be formed by changing some of the parents' characteristics. In our algorithm design, we used both Mutation and Recombination as follows:

3.1.4.1 Mutation

```
Start
    1. Two genes are selected randomly from the second strings of the each parent. (Fig.2-A)
    2. These two genes will be multiplying by two randomly generated numbers between 0 and 1, and new solutions are formed. (Fig. 2-B)
    3. For each $k$,
    4. The vehicle type assigned to each customer changes, and in each case the objective functions are calculated.
    5. end for
    6. The less objective function and its associated vehicle type assignment are chosen.
End
```

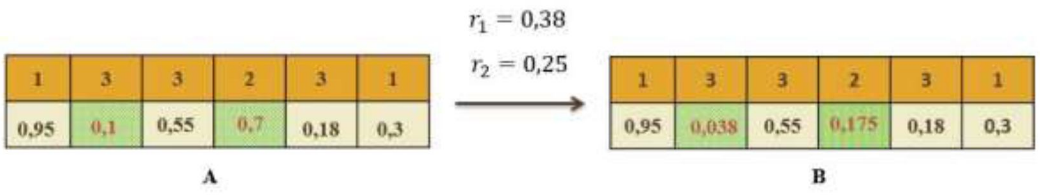


Fig 2. The proposed Mutation

3.1.4.2 Crossover

The following Pseudo-code shows, how parents are chosen for recombination

```
Start
    1. To select parents for recombination, it is started from the lowest front.
    2. if the individuals in the first front ( $nf$) are fewer than the individuals which should be selected by recombination ($nc$ ),
    3. All of the individuals in the first front will be selected and the rest individuals are selected from the second front with the lowest crowded distance.
    4. else
    5. Individuals with the less crowded distance will be selected.
    6. end if
End
```

After selecting parents for recombination, as Figure 3 illustrates, a two-point crossover will be implemented on our proposed two-string chromosomes.

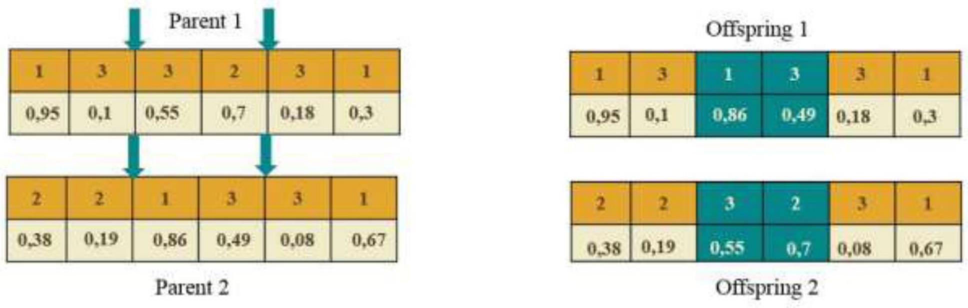


Fig 3. Two-point Crossover

3.1.5 Objective Functions

To evaluate the fitness of each generation, the following pseudo-code was implemented.

```
Start
    1. $S$ ← Set of customers which have not been served yet.
    2. $while\ S \neq \{\}$
    3. $for$ each vehicle ∈ $K$
    4. one depot is selected randomly
    5. customers are selected randomly and assigned to the vehicle
    6. checking the vehicle's capacity
    7. checking the customers' time windows
    8. checking depots' time windows
    9. end of $for$
    10. deleting served customers from $S$
End
```

Based on this pseudo-code, first set S, the set of customers, who have not been served yet, is formed. Next, for all available vehicle types and till the set S is not empty, one depot will be chosen randomly. Then, a customer is selected and assigned to one of vehicles randomly. If the vehicle's capacity is not exceeded, the arrival time of vehicle at the depot will be checked. If and only if, all of these conditions are met, that served customer will be removed from the set S and travel costs and vehicle capacity will be updated. If due to the full capacity of the current vehicle, we cannot use it anymore, a new vehicle will be chosen and one router will be added to the number of routes.

3.1.6 Survival Selection Mechanism

To replace the older generation, from all the individuals (parents and offsprings) some individuals will be selected. The selection mechanism is exactly like parents' selection.

3.1.7 Stopping Criteria

To be able to judge about the performance of our meta-heuristic approaches, we set time limit as a termination criterion. The stopping criteria for large test problems are three minutes and for other types it will be calculated, respectively.

3.1.8 Parameter Tuning

Since NSGA II has three parameters, Population size (PopSize), Crossover Rate (P_c), and Mutation Rate (P_m)), we designed an experiment based on the Taguchi (1986) method on 48 randomly generated large size test problems, with respect to maximizing the signal to noise ratio . The output data were analyzed by Minitab and the tuned parameters were selected. Table 1 and Figure 4 show the experiment levels and the Minitab software output.

Table 1
Levels of Taguchi Experiment

	Levels of Experiment		
	Low	Middle	Up
$PopSize$	50	100	200
P_c	0.8	0.85	0.9
P_m	0.025	0.05	0.075

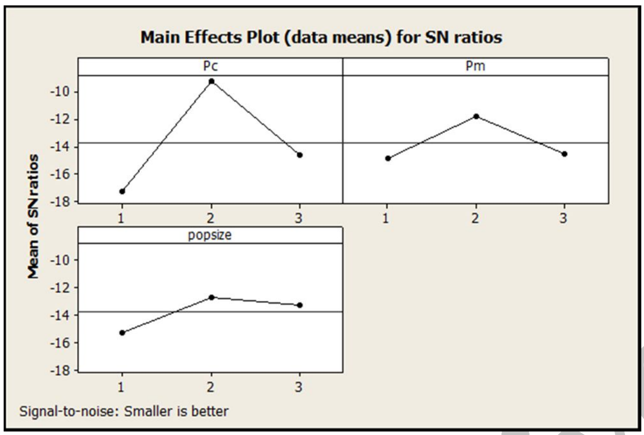


Fig 4. Minitab Output for Taguchi Experiment

According to Figure 4, the middle level is accepted and the optimal values for mentioned parameters are $PopSize = 100, P_C = 0.9$, and $P_m = 0.075$.

3.2 Multi-objective Simulated Annealing Approach

The second part of our solution approach considers one of the most novel multi-criteria solution methods, Multi-objective Simulated Annealing (MOSA). First introduced by Serafini (1992), this solution approach tries to adapt the well-known Simulated Annealing approach to solve multi-objective problems. One of the advantages of MOSA over other evolutionary algorithms is that, MOSA does not need lots of memory space and uses no more complicated algorithms to develop Pareto Optimal Front (POF) solutions.
Our MOSA solution approach, as Nam and Park (2000) offer, consists of the following three phases

3.2.1 Neighbor generating and Annealing

Since TDVRPHFMD is a finite state combinatorial optimization problem, the most general neighbor generating method is permute operation, which must satisfy the reachability and symmetry conditions. So, equation (13) is the most frequently used one for the annealing phase in such problems:

$$T_k = \alpha^k T_0 \qquad (13)$$

Where $0 < \alpha < 1$ is the cooling rate and T_0 is the initial temperature.

3.2.2 Transition Probability

In single objective Simulated Annealing problems, one can use the Metropolis or Logistic method for transition phase. However, since these functions support only scalar cost functions, they are not directly applicable to multi-objective optimization problems. The following equation shows the transition probability from state i to j:

$$P_t(i,j) = min\{e^{(-\frac{C(i,j)}{T})}, 0\} \qquad (14)$$

Where $c(i,j)$ is the cost criterion for transition from state i to j and T is the annealing temperature. Nam and Park (2000) evaluated six different schemes and later in their article, they suggest using random, average and fixed criteria. However, since the cost function in our TDVRPHFMD problem is the sum of travel costs and costs of using certain vehicles and it can be always calculated with fixed amounts, so the cost criterion in our problem is a fixed one.

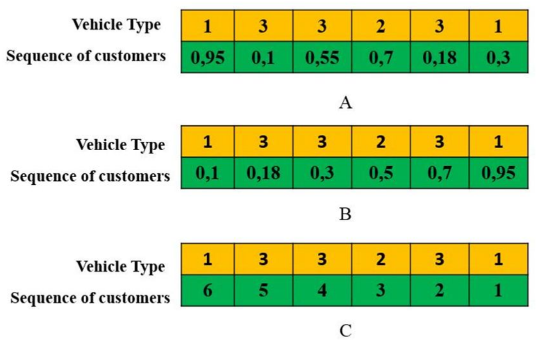
3.2.3 Deciding whether to stay or move in non-dominated situation

It was pointed out earlier that in multi-objective optimization problems we confront non-dominated solutions. If the new state is the same level of value as the current state, there can be two different scenarios: move to a new state or stay in the current state. Nam and Park (2000) suggested that the move state is better, because in stay state the algorithm is not able to search the solutions, which are in the middle of Pareto frontier, but in move state the algorithm can move freely and continue the search in the middle part of the Pareto fronts. The following shows the pseudo-code of the proposed MOSA.

```
Start
    1.  S=$S_0$
    2.  T=$T_0$
    3.  Repeat
    4.    Generate a neighbor S'=N(S)
    5.    If C(S') dominates C(S)
    6.          move to S'
    7.    else if C(s) dominates C(S')
    8.          move to S' with the transition
        probability $P_t(C(S), C(S'), T)$
    9.    else if C(S) and C(S') do not dominate
        each other
    10.         move to S'
    11.   end if
    12. T= annealing (T)
End repeat (until the termination is satisfied)
```

3.2.4 Solution Representation

Since we propose two meta-heuristic approaches to solve the TDVRPHFMD problem, introduced in this paper and in order to judge which approach is better, all the possible conditions are chosen to be the same. So, the solution representation in MOSA is like the NSGA II representation. Figure 5 shows the representation of the two string solution:



A						
Vehicle Type	1	3	3	2	3	1
Sequence of customers	0,95	0,1	0,55	0,7	0,18	0,3

B						
Vehicle Type	1	3	3	2	3	1
Sequence of customers	0,1	0,18	0,3	0,5	0,7	0,95

C						
Vehicle Type	1	3	3	2	3	1
Sequence of customers	6	5	4	3	2	1

Fig 5-A. The proposed two-string representation before sorting the randomly generated the numbers in the second string
Fig 5-B. The proposed two-string chromosome after sorting the randomly generated the numbers in the second string
Fig 5-C. Proposed two-string chromosome with the sequenced customers

3.2.5 Parameter Tuning for MOSA

Since simulated annealing has only two parameters; namely, initial temperature (T_0) and cooling rate (α), in order to tune these parameters we used trial and error method. Based on the results, we considered $T_0 = 1200$ and $\alpha = 0.9$

4. Results

To evaluate the performance of our proposed algorithms and also to be able to compare the efficiency of these two different approaches, we tested our solution approach on three various problem types; namely, small, medium and large. These test problems are generated randomly by $MATLAB^{®}$ 7.12 and run on Lenovo computer with a 2.40 GHz $Intel^{®}$ $Core^{TM}$ $i5$ CPU and 4 GB RAM. Table 2 presents the detailed information about the test problems.
Since the two proposed meta-heuristics function differently, and in order to be able to compare the efficiency of these two algorithms, we considered running time as the stopping criterion. We set the running time of large size test problems equal to three minutes or 180 seconds and by using the following equations, we calculated the running time of other problem types.

$$\theta = \frac{180}{4 \, x \, 75} = 0.6 \qquad (15)$$

Table 2
Specifications of Randomly Generated Test Problems

	Number of Problems	Number of Customers	Number of Depots	Time (Seconds)
Large Size	10	15	2	18
Medium Size	10	45	3	81
Small Size	10	75	4	180

In equation (15), 180 is the running time set for solving large size test problems, while numbers 4 and 75 are the numbers of depots and customers in large size test problems, respectively. After multiplier θ was calculated, running time of small and medium size test problems are calculated by using equation (16).

$$t = v.m.\theta \qquad (16)$$

Where v and m are the number of customers and depots in small and medium test problems.
In small size test problems, we compared the results with outputs obtained from solving our mathematical formulation using $GAMS^{®}$ software and $Cplex^{®}$ solver. The metrics used to compare the performance of small size test problems was relative Percent Deviation (RPD), which was introduced by Naderi et al. (2011). Equation (15) shows the RPD concept.

$$RPD = \frac{Alg_{sol} - min_{sol}}{min_{sol}} \, x \, 100 \qquad (17)$$

Since in small size problems two different approaches (meta-heuristics and $GAMS^{®}$ output) are compared together, a new modification to RPD is necessary. Equation (18) depicts the new modification to RPD.

$$RPD = \frac{Alg_1 - min(Alg_1 - Alg_{2)}}{min(Alg_1 - Alg_{2)}} \, x \, 100 \qquad (18)$$

In each table, gap 1, and gap 2 compare the performance of each two solutions by considering number of routes (the first objective) and travel cost (the second objective), respectively. The value gap 3 is a simple average of gap 1 and gap 2. In each table one solution approach (shown above word gap) was considered as the first algorithm.

Table 3
Average of gaps for small size test problems for NSGA II and Model

	NSGA II Vs. Model	Model Vs. NSGA II
$gap\ 1_{ave}$ (%)	17.71	15.00
$gap\ 2_{ave}$ (%)	15.24	0.58
$gap\ 3_{ave}$ (%)	16.48	7.79

Table 4
Average of gaps for small size test problems for MOSA and Model

	MOSA Vs. Model	Model Vs MOSA
$gap\ 1_{ave}$ (%)	23.19	0.00
$gap\ 2_{ave}$ (%)	4.56	5.68
$gap\ 3_{ave}$ (%)	13.88	2.84

To compare the efficiency of NSGA II and MOSA, we employed the following metrics:

4.1 Spacing

First introduced by Schott (1995), Spacing is one of the metrics used to measure the distance variance of neighboring vectors in a known Pareto front. Equation 11 defines the Spacing metric.

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d} - d_i)^2} \qquad (19)$$

and

$$d_i = min_j(|f_1^i(\vec{x}) - f_1^j(\vec{x})/+|f_2^i(\vec{x}) - f_2^j(\overrightarrow{x})|) \qquad (20)$$

Where $i, j = 1, \dots, n$, $\bar{d}$ is the mean of all d_i and n is the number of vectors in the known Pareto front. When $S = 0$, all members are spaced evenly apart. The advantage of using this metric is that the researcher does not need to know the true Pareto front.

4.2 Generational Distance (GD)

This metric, which was proposed by Van Veldhuizen and Lamount (1998), reports how far on average the known Pareto front is from the true Pareto front. It is obvious that this metric requires the researcher to know the true Pareto front. The following equation shows the generation distance metrics.

$$GD = \sqrt{\frac{\sum_{i=1}^{n} d_i^2}{n}} \qquad (21)$$

Where n the number of vectors is in known Pareto front, d_i is the Euclidean distance between each number and the closest member of the true Pareto front. When $GD = 0$, known Pareto front and true Pareto front are the same.

Tables 5 and 6 show the average of these two metrics for NSGA II and MOSA for the small, medium, and large test problems.

Table 5
The average of Spacing Metric for NSGA II and MOSA

Test Problems	NSGA II	MOSA
Small Size	3,95E+05	3,78E+05
Medium Size	4,30E+05	3,60E+05
Large Size	4,24E+05	3,69E+05

Table 6
The average of Generational Distance Metric for NSGA II and MOSA

Test Problems	NSGA II	MOSA
Small Size	64.15	64.77
Medium Size	52.13	64.13
Large Size	50.44	61.70

4.3. Discussion

To test how valid our mathematical model is, we compared the results obtained from NSGA II and MOSA with the results of our MILP model, solved by using $GAMS^{®}$ software and $Cplex^{®}$ solver for small size test problems. Table 3 shows that our NSGA II approach works in 16.48 % of cases worse than MILP model, while MILP model has a worse performance in 7.79% of the cases. When it comes to the objective functions, it is clear that in terms of number of routes and total travel costs, MILP has a better performance than NSGA II.
According to Table 4, our MILP model has a better performance than MOSA, as well. In terms of number of routes, MILP model always finds fewer routes. In fact MOSA has a worse performance, in terms of number of routes in 23.19 % of cases However, in terms of total travel costs, MOSA has a better performance and is able to find routes with less total travel costs. In 4.56 % of cases, MOSA has a worse performance in terms of total travel costs, while MILP model, works in 5.68% of the cases worse than MOSA. These two tables show that the MILP model is robust and valid, and comparable to our NSGA II and MOSA approaches, which can find nearly good solutions in only 18 seconds of running times.
As mentioned in the results section, to judge the efficiency of two Meta-heuristics, we have used Spacing and Generational Distance metrics. The reason why we employed these metrics is that, in multi-objective optimization problems, there are some non-dominated solutions which have no special superiority to each other. Based on Table 5, which summarizes the information about the average Spacing metrics for NSGA II and MOSA for three types of problems, MOSA has a less Spacing metrics than NSGA II in all of three problem types. Less spacing metrics shows that solutions are distributed more evenly. This is an important fact, because more evenly distribution of solutions causes the algorithm to reach further space and diversifies the solution search procedure.

Table 6 compares the efficiency of NSGA II and MOSA based on Generational Distance metric. Generational Distance metrics measure how far the solutions from the true Pareto front are. According the Table 6, NSGA II has less Generational Distance than MOSA in all three problem types. In other words, the solutions which are found by NSGA II are closer to true Pareto front, compared to MOSA.

5. Conclusion

In this paper, a novel variant of vehicle routing problems, Time-dependent Vehicle Routing Problem with hard time-windows, heterogeneous fleet and multiple depots (TDVRPTWHFM) with intra depot routes was formulated. To the best of our knowledge, no paper has been published yet, which considered all of these assumptions simultaneously. To solve this bi-objective particular formulation, we employed two well- known multi-objective meta-heuristics; namely, NSGA II and MOSA. We tested our approaches on three different randomly generated problem types (small, medium and large size). In small size test problems, we compared our solutions with our mathematical formulation output, obtained by solving MILP model by using $GAMS^{®}$ software and $Cplex^{®}$ solver. The results showed both robustness of the model and the algorithms. To judge which multi-objective meta-heuristic (NSGA II or MOSA) is better, we compared the performance of algorithms, using Spacing and Generational Distance metrics. Results illustrated that MOSA found more evenly distributed solutions, which let MOSA search more diversely, while NSGA II was capable of finding closer solutions to the true Pareto front. For future research, it is advised to expand this problem and work on other variants of vehicle routing problem, such as Time-dependent Pick-up and Delivery Problem with stochastic demands or Time-dependent Vehicle Routing Problem with heterogonous fleet and traffic restriction, in which some routes are not open in some time intervals, or not all the vehicles can pave all the routes.

References

[1] Balseiro, S. R., Loiseau, I., Ramonet, J. (2011). An Ant Colony algorithm hybridized with insertion heuristics for the Time Dependent Vehicle Routing Problem with Time Windows, Computers and Operations Research, 38, 957-966.

[2] Dantzig, G.B., Ramser, J.H., (1959). The truck dispatching problem, Management Science, 6-80.

[3] Deb, K., Agrawal, S., Pratap, A., Meyarivan, T. (2002). A fast Elitist Non-dominated Sorting Genetic Algorithm for multi-objective Optimization (NSGA II), IEEE Transactions on Evolutionary Algorithms, 6, 182-197.

[4] Figliozzi, M.A. (2012). The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics, Transportation Research Part E, 48, 616-636.

[5] Fleischmann, B., Gietz, M., Gnutzmann, S. (2004). Time-varying travel times in vehicle routing, Transportation Science, 38, 160–173.

[6] Ichoua, S., Gendreau, M., Potvin, J.Y. (2003). Vehicle dispatching with time-dependent travel times, European Journal of Operational Research, 144, 379–396.

[7] Jagiełło, S., Żelazny, D., (2013). Solving Multi-criteria Vehicle Routing Problem by Parallel Tabu Search on GPU, International Conference on Computational Science, 2529-2532.

[8] Kok, A.L. (2010). Congestion Avoidance and Break Scheduling within Vehicle Routing, PhD. Thesis, University of Twente, Enschede, the Netherlands.

[9] Malandraki, C. (1989). Time Dependent Vehicle Routing Problems: Formulations, Solution Algorithms and Computational Experiments, Ph.D. Dissertation, Northwestern University, Evanston, Illinois.

[10] Malandraki, C., Daskin, M.S. (1992). Time-dependent vehicle-routing problems – formulations, properties and heuristic algorithms, Transportation Science, 26, 185–200.

[11] Naderi, B., Fatemi Ghomi, S.M.T., Aminyari, M., Zandieh, M. (2011). Scheduling open shops with parallel machines to minimize total completion times, Journal of Computational and Applied Mathematics235, 1275-1287.

[12] Nam, D., Park, C., (2000). Multiobjective simulated annealing: A comparative study to Evolutionary algorithms. International Journal of Fuzzy Systems, 2, 87–97.

[13] Schott J. (1995). Fault tolerant design using single and multicriteria genetic algorithms optimization, Department of Aeronautics and Astronautics. Cambridge: Master's thesis, Massachusetts, Institute of Technology.

[14] Serafini, P., (1992). Simulated Annealing for Multiobjective Optimization Problems in Proceeding of the 10th International Conference on Multiple Criteria Decision Making, Taipei Taiwan, 87-96.

[15] Solomon Test Problems: http://w.cba.neu.edu/~msolomon/problems.htm

[16] Srinivas, N., Deb, K. (1994). Multi-Objective function Optimization using non-dominated genetic algorithms', Evolutionary Computation, 2, 221-248.

[17] Taguchi, G. (1986). Introduction to quality engineering, White Plains: Asian Productivity Organization, UNIPUB.

[18] Tasgetiren, M., Faith, Suganthan, P., Gencyilmaz G., Chen, Angela H-L. (2009). A smallest position value approach, Invited book chapter in Differential Evolution: A handbook for Global Permutation-Based Combinatirial Optimization edited by Godfrey Onwubolu and Donald Davendra, Springer-Verlag, 121-138.

[19] Toth, P., Vigo, D., (2002). The Vehicle Routing Problem, Society for Industrial and Applied Mathematics Philadelphia.

[20] E. Ulungu, J. Teghem, P. Fortemps, D. Tuyttens, (1999). Mosa method: A tool for solving moco problems, Journal of Multi-Criteria Decision Analysis, 8, 221-236.

[21] Van Veldhuizen, D., Lamont G., B., (1998), Evolutionary Computation and Convergence to a Pareto Front. Proceeding at the Genetic Programming Conference, Stanford University, California, Stanford University Bookstore. 221-228,

[22] Van Woensel, T., Kerbache, L., Peremans, H., Vandaele, N. (2008). Vehicle routing with dynamic travel times: a queueing approach, European Journal of Operational Research, 186, 990–1007.

[23] Żelazny, D., (2012). Multicriteria optimization in vehicle routing problem, in: National Conference of Descrite Processes Automation, 157-163.