



**University of
Sunderland**

Matthopoulos, P-P and Sofianopoulou, Stella (2018) A Firefly Algorithm for the Heterogeneous Fixed Fleet VRP. *International Journal of Industrial and Systems Engineering*. ISSN 1748-5037 (In Press)

Downloaded from: <http://sure.sunderland.ac.uk/8818/>

Usage guidelines

Please refer to the usage guidelines at <http://sure.sunderland.ac.uk/policies.html> or alternatively contact sure@sunderland.ac.uk.

A Firefly Algorithm for the Heterogeneous Fixed Fleet VRP

Panagiotis-Petros Matthopoulos

Department of Industrial Management & Technology,
University of Piraeus,
80 Karaoli & Dimitriou St.,
185 34 Piraeus,
Greece
Email: gimatthop@gmail.com

Stella Sofianopoulou*

Department of Industrial Management & Technology,
University of Piraeus,
80 Karaoli & Dimitriou St.,
185 34 Piraeus,
Greece
Email: sofianop@unipi.gr
Current Institution
University of Sunderland, Business School, UK
*Corresponding author

Abstract: Vehicle routing is a key success factor in logistics problems. A variation of Vehicle Routing problem (VRP), the heterogeneous fixed fleet VRP in which the vehicles available for distribution activities are characterized by different capacities and costs, is tackled. A hybrid firefly algorithm for optimizing the routing of heterogeneous fixed fleet of vehicles in logistics distribution systems is presented. The principles and key steps of the proposed firefly algorithm are introduced in detail. Experimental results from solving the heterogeneous fixed fleet vehicle routing problem when tested on benchmark datasets are demonstrated. Moreover, the algorithm is compared with other algorithms solving similar problems in order to prove the effectiveness of the proposed hybrid firefly algorithm.

Keywords

Firefly algorithm; Nature Inspired Metaheuristic Algorithms; Combinatorial optimization; Vehicle routing problem

Biographical notes: Mr P-P Matthopoulos is graduate of the MSc in Industrial Management & Technology of the University of Piraeus, Greece.

Dr Stella Sofianopoulou is a Professor at the University of Piraeus, Department of Industrial Management and Technology, Greece. She holds a PhD from London School of Economics and Political Science, UK. Her main research interests include production management, logistics and supply chain management, operations management, operations

research, discrete optimization. Current Institution is University of Sunderland, Business School, UK.

1. Introduction

Logistics distribution is one of the core activities of every supply chain network. One of the most significant components of cost directly related to the supply chain network is caused by that activity (Ballou, 1999). Therefore, the vehicle routing problem (VRP), associated with the solution of distribution problems, has attracted much attention from researchers. The classical VRP is a generic problem. However, many variations of the problem exist in order to tackle all different aspects that can emerge from it. The current article deals with the variation of heterogeneous fixed fleet VRP in which the vehicles available for distribution activities are characterized by different capacities and costs. For tackling this variant, a new metaheuristic method of swarm intelligence called firefly algorithm (Yang, 2008) was employed. Despite of the fact that firefly algorithm was originally proposed for optimization problems with continuous variables, many attempts have been made in order to implement it in practical problems which are discrete in nature with very promising results. These include job scheduling problems (Sayadi et al., 2010), knapsack problems (Baykasoglu and Ozsoydan, 2014) and vehicle routing problems (Pan et al., 2013; Simic et al., 2015; Wang et al., 2015; Osaba et al., 2016; Saraei and Ali Ghaheri, 2017).

The aim of vehicle routing problem (VRP) is to determine optimal routes for collection or delivery of goods by a fleet of vehicles in a transportation network (Laporte, 1992), i.e. to minimize the total cost of transportation by travelling less distance and using fewer vehicles. It is a NP-hard (nondeterministic polynomial time-hard) problem that generalizes the classical travelling salesman problem by requiring an assignment of vertices to vehicles and a sequencing of these vertices within each vehicle route to obtain a solution.

VRP can be described as follows. Suppose there are M vehicles each one having a capacity of Q and N customers who must be served by a depot. The distance between every customer and the demand of each customer are known in advance. The vehicles start from the depot, deliver the goods to the customers and return back to the depot. The goal is to use the least number of vehicles and cover the shortest distance to serve the customers and at the same time meet the following two conditions: 1) the capacity of each vehicle in each route must not be exceeded, and 2) each customer is served by only one vehicle.

The classical VRP is a generic problem, therefore by adding different constraints that must be satisfied, different variants of the problem exist. The most important variants are the following. Similar to the classical VRP is the capacitated VRP (CVRP) which has the additional constraint of capacity, meaning that every vehicle must have uniform capacity. Heterogeneous fleet VRP (HFVRP) is the variant in which we have a fleet of vehicles with different capacities and costs for goods' distribution. VRP with time windows (VRPTW) is the variant in which every customer is associated with a time window $[b_i, e_i]$. Customers must be served during this time window. Moreover, there is a time window associated with the depot and during this window vehicles must leave and return back to the depot. VRP with multiple depots (VRPMD) is the variant in which there are multiple depots for serving customers. Each customer is assigned to a depot from which he must be served. Each depot has its own fleet of vehicles. VRP with pick-up and delivery (VRPPD) is a variant in which the possibility that customers return some commodities is contemplated. Hence, there is a need of taking into consideration that the returned goods must fit to the serving vehicle. VRP with backhauls (VRPb) is the variant in which customers can demand or

return some commodities. The quantities of commodities that are demanded or returned are known in advance. The basic constraint is that all deliveries must be made in each route before any pick-ups of returned commodities can be made. The stochastic VRP (SVRP) is the variant in which one or more components of the problem are random. These components can be for example the number of customers to be served or the demand of each customer. Finally, split delivery VRP (SDVRP) is the variant in which it is allowed that a customer can be served by different vehicles if this reduces overall costs. This variant is significant if the demand of a customer is as big as the capacity of a vehicle.

The remaining of the article is organized as follows: In Section 2 a comprehensive literature review on the VRP and its variants is presented. In Section 3, the mathematical formulation of HFFVRP is presented, together with the basic concepts of firefly algorithm. In Section 4, the methodology employed to implement the firefly algorithm in heterogeneous fixed fleet VRP is discussed. Section 5 is devoted in discussing computational and experimental results and finally Section 6 draws conclusions and discusses open problems for future research that could be addressed.

2. Literature Review

The HFVRP can be categorized into two sub variants. These are the fleet size and mix VRP (FSMVRP) and the heterogeneous fixed fleet VRP (HFFVRP). In FSMVRP there is an unlimited number of vehicles of each type while in HFFVRP the number of vehicles of each type is fixed. FSMVRP is suitable for making strategic decisions concerning the size and composition of a company's fleet. HFFVRP is suitable for tactical and operational decisions concerning the available vehicles that will be used in order to serve customers (Brandao, 2011). The current article addresses HFFVRP variant.

Taillard (1999) proposed a column generation heuristic method, based on adaptive memory procedure with the use of a tabu search, to solve the problem. A method that incorporates a threshold accepting metaheuristic, where a set of fixed threshold values are given during the search process, was proposed by Tarantilis et al., (2003). In addition, Tarantilis et al., (2004) proposed an algorithm that uses a simple swap plus extraction – reinsertion moves to reach high quality solutions. This algorithm seeks a balance between diversification and intensification by changing the threshold values dynamically. According to computational results, the second method, (Tarantilis et al., 2004), produces far better results than the first one (Tarantilis et al., 2003). Moreover, Li et al., (2007) adapted the record-to-record travel algorithm proposed by Duesk (1993) in order to solve the HFFVRP. This algorithm allows uphill movements in order to escape from local optimum and continues the search in the search space only if the objective function of the new solution, after escaping local optimum, is within a given percentage of the old solution's objective function. Li et al., (2010) proposed a multi-start adaptive memory programming for solving the problem of HFFVRP. A heuristic method is used to generate vehicle routes and then a modified tabu search algorithm improves the quality of the solutions, followed by a post optimization procedure applying 2-Opt and 3-Opt moves in the best-found solution in order to further improve it.

To solve the HFFVRP, Brandao (2011) developed a tabu search metaheuristic method, while Naji-Azimi and Salari (2013) proposed an integer linear programming based heuristic procedure that could be used to improve the quality of existing methods, assuming that an initial feasible solution is been given. The method follows a disrupt and repair paradigm

where the initial solution is destroyed and repaired by solving the integer linear programming based model to optimality.

Particle swarm optimization techniques have been used to solve optimization problems. For example, a particle swarm optimization algorithm was introduced by Belmecheri et al. (2013) in order to solve the HFVRP with mixed backhauls and time windows. One of the latest algorithms of swarm intelligence is firefly algorithm introduced by Yang (2008). This algorithm was introduced to solve continuous optimization problems and mainly NP-hard problems. Despite that, modified firefly algorithm is also used to solve discrete problems, such as HFFVRP.

Despite the fact that there exist some very effective techniques for solving routing problems, such as exact methods (Laporte, 1992), heuristics and metaheuristics, the development of new metaheuristics for addressing routing problems still attracts much attention. Recently attempts of solving vehicle routing problems have been made with the use of simulated annealing (Sajjadi and Cheraghi, 2011; Seidgar et al., 2016; Javad and Karimi, 2017).

Due to the fact that many real life practical problems are discrete, variations of firefly algorithm attracted much attention within recent years for solving these problems. Khadwilard et al., (2011) applied the firefly algorithm for solving the job shop scheduling problem. The results indicated that although firefly algorithm could produce best known solutions it also produced local optima in several instances. Wang et al., (2012) developed a modified firefly algorithm for solving the path planning problem for uninhabited combat air vehicle. As mentioned by the authors, compared to other population-based optimization algorithms, the firefly algorithm outperformed them. Sayadi et al., (2013) applied a discrete firefly algorithm for solving the manufacturing cell formation problem, a discrete combinatorial optimization problem. In their paper the authors mention that the proposed algorithm often out-performed previously presented algorithms. Jati and Suyanto (2011) developed a discrete firefly algorithm for solving the symmetric TSP. Moreover, Jati et al. (2013) as well as Kumbharana and Pandey (2013) presented an evolutionary discrete version of firefly algorithm for solving TSP where the distance between two fireflies was measured using Hamming's method.

Few attempts have been made in order to solve vehicle routing problems with the use of firefly algorithm. Regarding the vehicle routing problem and its variants Pan et al. (2013) used firefly algorithm for solving vehicle routing problem with time windows. VRPTW is developed from CVRP, therefore, they adopted the real-coded schema of CVRP as well as the real-coded schema for PSO algorithm in VRP introduced by Wu et al. (2007). Simic et al., (2015) applied a hybrid firefly algorithm for solving the HFFVRP. Comparing the experimental results, applied in realistic data of a company, with the company's empirical results the algorithm outperformed them. Wang et al., (2015) applied a discrete firefly algorithm for optimizing the route planning of a milk-run system. Computational results showed that the proposed algorithm outperformed other metaheuristic algorithms such as tabu search. Osaba et al., (2016) proposed four different versions of an evolutionary discrete firefly algorithm for solving the VRPTW with promising results, while Osaba et al., (2017) applied a discrete firefly algorithm for solving a rich vehicle routing problem outperforming all other algorithms. Recently Koc et al., (2015a) presented a heuristic for the HFFVRP while a review article on heterogeneous vehicle routing by Koc et al., (2015b) classifies the literature on heterogeneous vehicle routing problems.

As seen in detailed reviews of firefly algorithm (Fister et al., 2013; Ali et al., 2014) the firefly algorithm has been successfully applied to many optimization problems. Nevertheless, the lack of research in applying firefly algorithm in HFFVRP combined with the growing scientific interest in bio-inspired algorithms and the good performance shown by the firefly algorithm since its proposal, as well as the fact that the particular VRP variant can cover more practical situations in transportation and logistics, has motivated its use in this study.

3. Mathematical Formulation of HFFVRP and Basic Concepts of Firefly Algorithm

3.1 Mathematical formulation of HFFVRP

As mentioned above the current article deals with the variation of heterogeneous fixed fleet VRP in which the vehicles available for distribution activities are characterized by different properties, i.e. capacities and costs. For tackling this variant, a new metaheuristic method of swarm intelligence called firefly algorithm. To model the heterogeneous fixed fleet vehicle routing problem some basic assumptions must be followed. These are: The variable cost which depends on the distance travelled as well as the fixed cost which is different for every type of vehicle must be considered to calculate the total cost. There is only one central depot from which all vehicles start and finish their routes. The demand of each customer is fixed and known in advance as well as the number of vehicles of each type. The distance between customers is calculated using the Euclidean two-dimensional space.

The HFFVRP can be formulated as follows, where $N=\{0,1,2,3,\dots,n\}$ are the nodes-customers, k are the vehicle types and $K=\{1,2,\dots,m\}$ are the vehicles. F_k is the fixed cost for vehicle k and C_{ij}^k is the variable cost for vehicle k traveling from node i to node j . Q_k is the capacity of vehicle k and q_j is the demand of node j . Moreover, y_{ij} is the vehicle's load between node i and node j . Finally, x_{ij}^k is equal to 1 if the vehicle k goes from node i to node j , otherwise it is equal to 0. For determining the central depot, variable $X_{i,j}^k$ is used. When index i is equal to 0, i.e. when variable becomes $X_{0,j}^k$, we refer to the central depot. Specifically variable $X_{0,j}^k$ is used to denote that vehicle k goes from the central depot to node j .

$$\text{Min } \sum_{k \in K} \sum_{j \in N} F_k X_{0j}^k + \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} C_{ij}^k X_{ij}^k \quad (1)$$

Subject to:

$$\sum_{k \in K} \sum_{i \in N} X_{ij}^k = 1 \quad \forall j \in N \quad (2)$$

$$\sum_{i \in N} X_{ij}^k - \sum_{i \in N} X_{ji}^k = 0 \quad \forall k \in K \quad \forall j \in N \quad (3)$$

$$\sum_{j \in N} X_{0j}^k \leq m_k \quad \forall k \in K \quad (4),$$

$$\sum_{i \in N} y_{ij} - \sum_{i \in N} y_{ji} = q_j \quad \forall j \in N \quad (5)$$

$$q_j X_{ij}^k \leq y_{ij} \leq (Q_k - q_i) X_{ij}^k \quad \forall k \in K, \forall i \in N, \forall j \in N \quad (6)$$

$$y_{ij} \geq 0 \quad \forall i \in N, \forall j \in N \quad (7)$$

$$X_{ij}^k \in \{0,1\} \quad \forall i \in N, \forall j \in N, \forall k \in K \quad (8)$$

The objective function (1) is to minimize the total costs consisting of the variable costs and the fixed costs of the different types of vehicles who serve the nodes. Constraint (2) makes sure that each customer can be visited only once by the vehicle k while constraint (3) makes sure that the same vehicle that visits a customer must leave from that customer. The maximum number of vehicles available for each vehicle type is imposed by constraint (4). Constraint (5) ensures that the quantity of goods a vehicle carries before and after visiting a customer is equal to the demand of that customer. With constraint (6) we ensure that the vehicle capacity is never exceeded. Finally, constraint (7) indicates that the vehicle's load y_{ij} is non-negative and constraint (8) that X_{ij}^k is a binary variable.

3.2 Basic concepts of firefly algorithm

Firefly algorithm is a relatively new swarm intelligence optimization method that is inspired by social behavior of fireflies and the phenomenon of bioluminescent communication. Fireflies communicate and attract each other with varied flashing patterns. The flashing characteristics of fireflies were idealized in order to design a firefly inspired algorithm. Firefly algorithm follows only three idealized rules (Yang, 2010): 1) Fireflies are unisex; therefore, they attract each other regardless of sex. 2) Attractiveness is proportional to firefly brightness. The less bright firefly will move towards the brighter one. Brightness decreases when the distance between two fireflies increases. If there are no brighter fireflies for a particular firefly, this firefly will move randomly in the space. 3) The brightness of a firefly is determined by the objective function. For a maximization problem brightness is proportional to the value of the objective function.

The basic concepts of the firefly algorithm are as follows:

- i. Relative brightness

The brightness of a firefly is represented by I and is given by the following equation:

$$I = I_0 \times e^{-\gamma r_{ij}^2} \quad (9)$$

Where I_0 is its initial brightness at $r=0$. I_0 is associated with the objective function value at $r=0$ and the smaller the function value is the bigger I_0 will be. The degree of light attenuation is represented by γ .

- ii. Degrees of attraction

Degrees of attraction can be calculated by the following equation:

$$\beta = \beta_0 \times e^{-\gamma r_{ij}^2} \quad (10)$$

Where β_0 is the largest degree of attraction, namely the degree of attraction at $r=0$.

- iii. Distance between fireflies

Distance between firefly i and firefly j is defined by their Cartesian distance as shown below:

$$r_{i,j} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (11)$$

Where $x_{i,k}$ is the k th component of firefly i 's position vector and d is the position vector's dimension.

In the two-dimensional space equation (11) can be simplified as follows:

$$r_{i,j} = \sqrt{(x_i - x_j)^2 - (y_i - y_j)^2} \quad (12)$$

iv. Position update

Position update when firefly i is attracted to the brighter one j can be calculated as follows:

$$x_i = x_i + \beta_0 \times e^{-\gamma r_{ij}^2} \times (x_i - x_j) + \alpha \times (\text{rand} - 1/2) \quad (13)$$

The step factor is represented by α that is a random number in the range of $[0,1]$ while rand is a random number generator uniformly distributed in the range of $[0,1]$.

v. Parameters values selection

The value of light attenuation γ can determine the convergence speed of the algorithm (Imanirad et al., 2013). For most cases γ lies between $[0.1, 1]$. Moreover, the algorithm's convergence speed can be determined also from the number of fireflies. For an improved convergence speed, the number of fireflies should be between 20 and 50 (Yang, 2009). Furthermore, for most cases parameter β_0 is equal to 1 and parameter $\alpha \in [0,1]$ (Yang, 2009).

4. Implementation of Firefly Algorithm for Solving HFFVRP

HFFVRP is a problem of logistics operations faced daily by many companies. Mismanagement of the available vehicle fleet can lead to increased transportation costs. Due to the fact that bio-inspired algorithms have shown good performance, a firefly algorithm is proposed for implementing the problem in order to minimize transportation costs and to better manage the vehicle fleet. The implementation of the firefly algorithm is presented below.

Firefly algorithm is a population-based algorithm. This means that in order the algorithm to begin an initial population of solutions is needed.

4.1 Creating Initial Population

The first step of the algorithm is to create the initial population of solutions. A greedy algorithm is applied for this reason. Every solution of the greedy algorithm is a firefly. Each solution is represented by a vector.

4.2 Parameter Selection

The second step is to determine the appropriate values for the important parameters. Therefore, for every parameter we assign the values as follows:

$$\beta_0=1, \gamma=1, \alpha=0.2, n=20 \text{ and iterations}=1000$$

The number of fireflies, i.e. the number of the solutions of the greedy algorithm is represented with the parameter n .

4.3 Greedy algorithm transformation

The third step is to transform the solution of the greedy algorithm in a firefly. For doing so we follow the coding introduced by Pan et al. (2013). Each firefly is represented by a vector of $n+k-1$ dimensions where n are the nodes and k are the vehicles. Greedy algorithm solution is a sequence of nodes inserted by the depot each time a vehicle must return to depot. With the adopted coding, every depot is replaced by a delimiter which states what vehicle has been used.

4.4 Calculation of brightness and attraction

The next step is to calculate the relative brightness I and the degrees of attraction β for every firefly using the equations (9) and (10) respectively.

4.5 Main firefly algorithm implementation

The main firefly algorithm begins in the current step. Initially, for every firefly we compare its brightness with the brightness of every other firefly. The less bright firefly will move towards the brighter one. This movement is the position update of the firefly calculated using equation (13). From this calculation the new firefly, i.e. the one that updated its position, is now represented with real numbers. Every other firefly is represented with integer numbers and that is why we decode the updated firefly in order to also express it with integer numbers. We adopted the decoding proposed by Pan et al. (2013). In order to create the new firefly we initially sort the vector and then assign the integer numbers to the real ones. This procedure is as follows:

Phase 0: Greedy algorithm output

1	4	2	6	9	3	5	10	7	8
---	---	---	---	----------	---	---	-----------	---	---

Phase 1: Coding in real numbers

Divide every value of Phase 0 with the highest number of the vector

0.1	0.4	0.2	0.6	0.9	0.3	0.5	1	0.7	0.8
-----	-----	-----	-----	-----	-----	-----	---	-----	-----

Phase 2: Position update

Position update performed by the firefly algorithm using equation (13)

0.35	0.28	0.1	0.9	0.73	0.86	0.57	0.4	0.6	1
------	------	-----	-----	------	------	------	-----	-----	---

Phase 3: Sorting the vector

Sorting the vector in an ascending order

0.1	0.28	0.35	0.4	0.57	0.6	0.73	0.86	0.9	1
-----	------	------	-----	------	-----	------	------	-----	---

Phase 4: Decoding in integer numbers

Example 1: given value 0.1 of phase 3, the slot number of the vector in phase 2 (slot number 3) is identified and then the value of the same slot in phase 0 is selected, i.e. value 2.

Example 2: given value 0.57 of phase 3, the slot number of the vector in phase 2 (slot number 7) is identified and then the value of the same slot in phase 0 is selected, i.e. value 5.

2	4	1	10	5	7	9	3	6	8
---	---	---	-----------	---	---	----------	---	---	---

A testing procedure then is following. First of all, the uniqueness of the new firefly is checked, followed by checking if there are delimiters in the beginning of the vector. If this is happening, they are transferred to the end of the vector. Moreover, it is checked if there is more than one delimiter in row. In this situation, the position of the first delimiter is kept and the rest is transferred at the end of the vector. Finally, a penalty factor is used in order to check the capacity of every vehicle in its route. With the use of the penalty factor the objective function is as follows:

$$\text{Min} \sum_{k \in K} \sum_{j \in N} F_k X_{0j}^k + \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} C_{ij}^k X_{ij}^k + R * \sum_{k=1}^K \max(\sum_{i=1}^N Q_k y_{ik} - q, 0) \quad (14)$$

4.6 Solution improvements

In the current stage, step six, techniques are implemented to improve the algorithm's solution. In the case that the penalty factor is equal to zero, i.e. the capacity is not violated, the solution cannot be improved, but if the penalty factor is not equal to zero the capacity has been violated and thus the solution in this form is not acceptable. For this reason, the given solution is improved using the following techniques. Initially, the nodes in the route are redistributed. Nodes that cannot be served by a vehicle in one route due to lack of capacity are transferred in another vehicle that has the appropriate capacity to serve them. During this procedure the aim is to minimize the increase of the total cost of the route or to maximize the decrease of the total cost.

If this procedure is successful then a local search heuristic (2-opt) is used to further improve the solution. The 2-opt method consists of two procedures. The one is, the intra-route in which two edges are removed from a sub-route and they are reconnected in such a way to reduce the overall cost of the sub-route. The other one is, the inter-route. In this procedure two edges from two different sub-routes are removed and reconnected by uniting edges from both original sub-routes in such a way that now new sub-routes are created in a way to reduce the overall cost of the solution. In the case of inter-route, there is always a capacity check so that the new solution will be valid and feasible.

For the solutions that do not exceed the capacity limit after the first transferring of nodes between vehicles, both the intra-route and inter-route procedures to further improve the solution are used. But for the solutions that exceed the capacity limit after the first transferring of nodes only the inter-route procedure can be used, in order to improve the solution and make it valid and feasible.

4.7 Objective function calculation

In this step, the objective function of the new solution is calculated using equation (14). If the value of the objective function of the new solution is up to 50% worse than the first solution, the decoded one, is kept because in future position updates it can provide a better solution.

4.8 Recalculation of fireflies

In the current step, the brightness, the degrees of attraction and the distances between fireflies are recalculated, because there are fireflies that have changed. Then the same procedure is applied for all the other fireflies. The algorithm ends when the number of iterations is reached. At the end of every iteration the best solution is calculated.

4.9 Computing the best global solution

This is the final step. After all the iterations, the best global solution is calculated by comparing the value of the objective function of every best solution of each iteration.

5. Computational Results

To illustrate the effectiveness and validity of the proposed algorithm various benchmark instances have been selected for the computations. All these datasets containing these instances have been used to solve the capacitated vehicle routing problem. These datasets have been used in the present study because there are not any datasets for solving the heterogeneous fixed fleet vehicle routing problem available in the literature. For solving the CVRP an adjustment of the algorithm is needed so as not to consider any fixed costs and to have a homogeneous fleet of vehicles. The assumption made is that, if the proposed algorithm is effective for solving the CVRP, it will also be effective and valid for solving the HFFVRP.

These datasets were first proposed by Christofides and Eilon (1969) and Augerat et al. (1995). In the following tables the benchmark instances (column 1), the best-known solution (column 2), the best solution found from other algorithms which solve the problem as well as the percentage difference between their best solution and the best-known solution are depicted. Moreover, the best solution given by our proposed algorithm and the percentage difference between our best solution and the best-known solution are also presented (last two columns).

Each dataset is categorized based on the number of nodes. Thus, there are small datasets with up to 39 nodes and larger ones with more than 40 nodes. This is done because for small datasets our algorithm behaves in a different way than for larger ones. The algorithms used to compare the proposed firefly algorithm to are: i) a hybrid heuristic algorithm (SA+NN) proposed by Abdelazziz et al. (2014), ii) a cluster and search heuristic (CLOVES) proposed by Ganesh and Narendran (2007), iii) a genetic algorithm (FAGA) proposed by Sivaran et al. (2014), iv) an evolutionary algorithm (DE) proposed by Kromer et al. (2013), v) a hybrid algorithm (SR-GCWS) proposed by Juan et al. (2010) and vi) a sweep algorithm (SWEEP) proposed by Na et al. (2011). The results provided for the above-mentioned algorithms have been respectively obtained from relevant research papers.

Table 1: Dataset A with up to 39 nodes by Augerat et al. (1995)

dataset	optimal	SA+NN	%	FAGA	%	DE	%	SR-GCWS	%	SWEEP	%	firefly	%
A-n32-k5	784	1012	29	804	3	1028	3	787	0,04	810	3	831	6
A-n33-k5	661	847	28	661	0	837	27	662	0,01	686	4	711	8
A-n33-k6	742	919	24			967	30	742	0	743	0,01	783	6
A-n34-k5	778	933	20			998	28			785	1	827	6
A-n36-k5	799	1126	41			1080	35			826	3	870	9
A-n37-k5	669	876	31			827	24	672	0,04	670	0	669	0
A-n37-k6	949	1180	24			1174	24			962	1	1021	8
A-n38-k5	730	920	26			921	26	733	0,04	749	3	787	8
A-n39-k5	822	1147	40	839	2	1069	30					898	9
A-n39-k6	831	1065	28			1237	49	833	0,02	856	3	868	4

Table 2: Dataset A with more than 40 nodes by Augerat et al. (1995)

dataset	optimal	SA+NN	%	FAGA	%	DE	%	SR-GCWS	%	SWEEP	%	firefly	%
A-n44-k6	937	1356	45			1136	21			957	2	1051	12
A-n45-k6	944	1210	28	957	1	1307	38	945	0,01	991	5	1108	17
A-n45-k7	1146	1361	19			1523	33	1146	0	1173	2	1281	12
A-n46-k7	914	1071	17			1316	44			946	4	1049	15
A-n48-k7	1073	1292	20	1101	3	1327	24			1113	4	1218	14
A-n53-k7	1010	1261	25			1512	50					1200	19
A-n54-k7	1167	1414	21			1601	37					1374	18
A-n55-k9	1073	1317	23	1089	1	1873	75	1074	0,01	1095	2	1324	23
A-n60-k9	1354	1733	28			2077	53	1355	0,01	1420	5	1650	22
A-n61-k9	1034	1285	24			1491	44	1039	0	1100	6	1362	32
A-n62-k8	1288	1604	25	1288	0	2025	57			1359	6	1566	22
A-n63-k9	1616	2001	24			2324	44	1622	0	1712	6	1976	22

A-n63-k10	1314	1542	17			2052	56			1386	5	1606	22
A-n64-k9	1401	1821	30	1401	0	2083	49			1499	7	1684	20
A-n65-k9	1174	1429	22			1980	69	1181	1	1223	4	1522	30
A-n69-k9	1159	1333	15			1869	61			1207	4	1534	32
A-n80-k10	1763	2318	31	1777	1	2772	57	1766	0	1866	6	2235	27

Table 3: Dataset B with up to 39 nodes by Augerat et al. (1995)

dataset	optimal	SA+NN	%	FAGA	%	DE	%	SR-GCWS	%	SWEEP	%	firefly	%
B-n31-k5	672	713	6	672	0			676	1	677	1	672	0
B-n34-k5	788	995	26							802	2	788	0
B-n35-k5	955	1006	5	968	1			956	0	962	1	986	3
B-n38-k6	805	888	10							817	1	834	4
B-n39-k5	549	683	24	549	0			553	1	575	5	605	10

Table 4: Dataset B with more than 40 nodes by Augerat et al. (1995)

dataset	optimal	SA+NN	%	FAGA	%	DE	%	SR-GCWS	%	SWEEP	%	firefly	%
B-n41-k6	829	927	12					834	1	843	2	919	11
B-n43-k6	742	789	6							746	1	793	7
B-n44-k7	909	1212	33							942	4	1003	10
B-n45-k5	751	943	26	751	0			754	0,01	797	6	843	12
B-n45-k6	678	844	24							732	8	751	11
B-n50-k7	741	948	28	741	0			744	0,01	779	5	838	13
B-n50-k8	1312	1560	19							1349	3	1414	8
B-n51-k7	1032	1177	14									1230	19
B-n52-k7	747	891	19					749	0,01	758	1	889	19
B-n56-k7	707	836	18	721	2			712	1	726	3	825	17
B-n57-k7	1153	1477	28									1469	27
B-n57-k9	1598	1744	9	1598	0			1602	0	1642	3	1802	13
B-n63-k10	1496	1688	13									1811	21
B-n64-k9	861	1027	19	861	0			868	1	1161	35	1162	35

B-n66-k9	1316	1578	20						1363	4	1560	19
B-n67-k10	1032	1287	25				1039	1	1080	5	1318	28
B-n68-k9	1272	1399	10				1276	0	1308	3	1512	19
B-n78-k10	1221	1363	12	1239	1		1228	1	1268	4	1625	33

Table 5: Dataset P with up to 39 nodes by Augerat et al. (1995)

dataset	optimal	SA+NN	%	FAGA	%	DE	%	SR-GCWS	%	SWEEP	%	firefly	%
P-n16-k8	450	546	21							513	14	450	0
P-n19-k2	212	253	19	212	0			212	0	219	3	212	0
P-n20-k2	216	267	24					217	0,01	217	0	216	0
P-n21-k2	211	288	36							211	0	211	0
P-n22-k2	216	274	27	216	0			218	0,01	216	0	216	0
P-n22-k8	603	667	11					589	-2	560	-7	603	0
P-n23-k8	529	743	40	529	0					554	5	529	0

Table 6: Dataset P with more than 40 nodes by Augerat et al. (1995)

dataset	optimal	SA+NN	%	FAGA	%	DE	%	SR-GCWS	%	SWEEP	%	firefly	%
P-n40-k5	458	563	23					461	0,01	467	2%	508	11
P-n45-k5	510	662	30	510	0							595	17
P-n50-k7	554	647	17									689	24
P-n50-k8	631	721	14	648	3			634	0			828	31
P-n50-k10	696	808	16					700	1			843	21
P-n51-k10	741	857	16	760	3			741	0			895	21
P-n55-k7	568	679	20									670	18
P-n55-k8	588	690	17	588	0							741	26
P-n55-k10	694	832	20									853	23
P-n60-k10	744	896	20	744	0			748	1			940	26
P-n60-k15	968	1159	20									1195	23

P-n65-k10	792	964	22	806	2	796	1			1020	29
P-n70-k10	827	989	20			830	0			1119	35
P-n76-k4	593	753	27			598	1	612	3	733	24
P-n76-k5	627	671	7			633	1			832	33
P-n101-k4	681	891	31			692	2	715	5	874	28

Table 7: Dataset E with up to 39 nodes by Christofides and Eilon (1969)

dataset	optimal	CLOVES	%	SR-GCWS	%	SWEEP	%	firefly	%
E-n22-k4	375	375	0	375	0	375	0	375	0
E-n23-k3	569	569	0			569	0	569	0
E-n30-k3	534	534	0	505	-5	543	2	534	0
E-n33-k4	835	835	0	837	0,01	852	2	835	0

Table 8: Dataset E with more than 40 nodes by Christofides and Eilon (1969)

dataset	optimal	CLOVES	%	SR-GCWS	%	SWEEP	%	firefly	%
E-n51-k5	521	521	0	525	1	532	2	626	20
E-n76-k7	682	690	1	689	1	703	3	922	35
E-n76-k10	830	867	4	835	1	907	9	1186	43
E-n76-k14	1021	1032	1	1024	0,01	1072	5	1406	38

In tables 1 to 8 it can be observed that for all the small datasets, the proposed firefly algorithm is very effective and efficient. It is either producing the best-known solutions or the best solution of the algorithm in each instance has very small percentage difference from the best-known solution. In 14 instances the proposed algorithm found the best-known solution.

On the other hand, in larger datasets the proposed algorithm is less effective because the percentage difference between the best solution of the algorithm and the best-known solution is much bigger than in smaller datasets. Nonetheless, in datasets A, B and P the proposed algorithm outperforms the hybrid heuristic algorithm (SA+NN) which is the only one with whom the proposed algorithm can be compared and provide safe conclusions. In dataset E all other algorithms outperform the proposed algorithm.

In order to explain the different effectiveness of the proposed algorithm between the smaller and the larger datasets some tests need to be conducted. We do this to better understand the proposed algorithm's behavior.

It should be noted that equation (13), that refers to the position update of a firefly, contains a term of randomness and a random number generator. For this reason, if the repositioning is not towards a local or global optimum, i.e. if there is no brighter firefly to approach, then the firefly will move randomly performing a random 'walk'. In order to investigate this, an empirical analysis was conducted, and the mean value of the objective function throughout all iterations was recorded. When the mean value is not being reduced, but presents fluctuations, then fireflies engage in a random walk and this seems to justify the reduced efficiency of the proposed firefly algorithm in larger datasets. Several researches refer to this, as for example (Jati et al., 2011) within the context of TSP, providing alternatives for the position update that seem to be problem specific and cannot be implemented in the HFFVRP.

The above observation explains why the algorithm exhibits different performance in various datasets. In smaller datasets, randomness is more limited during the random walk because the options for the movement are fewer than in larger datasets. For that reason, in small datasets it is easier for the algorithm to produce improved solutions. This is why the best or near best-known solutions are produced. On the other hand, in larger datasets there are more options for movement and thus randomness is greater. For this reason, the algorithm does not perform as good as in smaller datasets.

Nonetheless, the proposed algorithm outperforms in any case the hybrid heuristic algorithm proposed by Abdelazziz et al. (2014). It is the only algorithm with whom the proposed algorithmic procedure can be compared with due to the fact that it is the only algorithm providing results in every instance of the datasets. The comparison implies that the proposed algorithm is very promising in solving discrete optimization problems.

6. Conclusions

The present paper addresses the heterogeneous fixed fleet vehicle routing problem. A hybrid firefly algorithm to solve this problem was implemented. Despite the fact that firefly algorithm was originally proposed to solve continuous problems, the implementation of this algorithm for solving discrete problems is very promising.

Two major conclusions can be drawn from the current research. First, it can be concluded that the proposed algorithm doesn't guide the fireflies in their movement but the fireflies move according to a random walk. The concept of randomness exists in the firefly algorithm because the equation of position update incorporates a random parameter. We can understand if the fireflies move randomly, by checking the fluctuation of the mean value of the objective function.

Next, the effectiveness of the proposed algorithm is conclusive. To verify its effectiveness benchmark instances by various datasets were solved. The computational results indicate that the proposed algorithm is very effective in small datasets, up to 39 nodes. In fact, it produces either the best-known solutions or solutions within a very small percentage from the best-known ones. For larger datasets, the effectiveness of the algorithm is reduced and the provided solutions have bigger percentage differences from the best-known ones. The way the algorithm produces solutions is affected by the randomness of the fireflies'

movement. In larger datasets, this randomness is much larger than in smaller datasets and this is why the algorithm exhibits limited effectiveness. It is worth mentioning however, that the movement of a firefly towards any other brighter firefly does coherently involve randomness and random moving in firefly algorithm.

Concluding, the adaptation of firefly algorithm for solving HFFVRP has very promising results. Computational results showed that the proposed algorithm can minimize transportation costs and effectively manage the available vehicle fleet despite the random walk issue.

Future work should focus in resolving the random walk generation in Heterogeneous Fixed Fleet VRP, in order to make fireflies move in a desired manner towards local or global optima. A different calculation approach to the movement or position update might be in the right direction. Managing fireflies to move in that way, the effectiveness of the algorithm will increase in both small and large datasets and will produce much better solutions. Moreover, the implementation of the proposed algorithm to real data from the logistics industry is highly recommended to illustrate its effectiveness in actual problems. Swarm intelligence and in particular the firefly algorithm could be also used for solving more variations of the classical vehicle routing problem and so future work should concentrate on that area.

References

- Abdelaziz, M.M., El-Ghareeb, H.A., and Ksasy, M.S.M. (2014) 'Hybrid heuristic algorithm for solving capacitated vehicle routing problem', *International Journal of Computers and Technology*, Vol. 12 No.9, pp.3844-3851
- Ali, N., Othman, M.A., Husain, M.N., and Misran, M.H. (2014) 'A review of firefly algorithm', *ARPN Journal of Engineering and Applied Sciences*, Vol. 9 No. 10, pp.1732-1736
- Augerat, P., Belenguer, J.M., Benavent, E., Corberan, A., Naddef, D., and Rinaldi, G. (1995) 'Computational results with a branch and cut code for the capacitated vehicle routing problem', *Research report 949-M, University Joseph Fourier, Grenoble, France*
- Ballou, R.H. (1999) *Business Logistics Management: Planning, Organizing and Controlling the Supply Chain*, 4th int. ed., Prentice-Hall International, Upper Saddle River, NJ
- Baykasoglu, A. and Ozsoydan, F.B. (2014) 'An improved firefly algorithm for solving dynamic multidimensional knapsack problems', *Expert Systems with Applications*, Vol. 41, pp. 3712-3725
- Belmecheri, F., Prins, C., Yalaoui, F., and Amodeo, L. (2013) 'Particle swarm optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls and time windows', *Journal of Intelligent Manufacturing*, Vol. 24 No. 4, pp.775-789
- Brandao, J. (2011) 'A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem', *Computers & Operations Research*, Vol. 38 No.1, pp.140-151
- Christofides, N. and Eilon, S. (1969) 'An algorithm for the vehicle dispatching problem', *Operational Research Quarterly*, Vol.20 No.3, pp.309-318
- Koç C, Bektaş T, Jabali O, Laporte G, (2015a) A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows, *Computers & Operations Research*, Vol 64: 11-27

- Koç C, Bektaş T, Jabali, O, Laporte G, (2015b) Thirty years of heterogeneous vehicle routing, *European Journal of Operational Research*, Vol.249, pp.1-21
- Dueck, G. (1993) 'New optimization heuristics: the great deluge algorithm and the record-to-record travel', *Journal of Computational Physics*, Vol. 104 No.1, pp.86-92
- Fister, I., Fister Jr., I., Yang, X.S., and Brest, J. (2013) 'A comprehensive review of firefly algorithms', *Swarm and Evolutionary Computation*, Vol. 13, pp.34-46
- Ganesh, K. and Narendran, T.T. (2007) 'CLOVES: A cluster-and-search heuristic to solve the vehicle routing problem with delivery and pick-up', *European Journal of Operational Research*, Vol. 178, pp.699-717
- Imanirad, R., Yang, X.S., and Yeomans, J.S. (2013) 'A biologically – inspired metaheuristic procedure for modeling-to-generate-alternatives', *International Journal of Engineering Research and Applications*, Vol.3 No.2, pp.1677-1686
- Jati, G. and Suyanto, (2011) 'Evolutionary discrete firefly algorithm for travelling salesman problem', *Adaptive and Intelligent Systems, Lecture notes in computer science*, Springer-Verlag, Berlin Heidelberg, pp.393-403
- Jati, G., Manurung, R. and Suyanto, (2013) 'Discrete firefly algorithm for traveling salesman problem: A New Movement Scheme', *Swarm Intelligence and Bio-inspired Computation*, pp.295-312
- Javad, M. O. M. and Karimi, B. (2017) 'A simulated annealing algorithm for solving multi-depot location routing problem with backhaul', *International Journal of Industrial and Systems Engineering*, Vol. 25 No. 4, pp. 460-477
- Juan, A.A., Faulin, J., Ruiz, R., Barrios, B. and Caballe, S. (2010) 'The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing problem', *Applied Soft Computing*, Vol. 10, pp.215-224
- Khadwilad, A., Chansombat, S., Thepphakorn, T., Thapatsuwan, P., Chainate, W. and Pongcharoen, P. (2011), 'Application of firefly algorithm and its parameter setting for job shop scheduling' in *The First Symposium on Hands-On Research and Development*, pp. 1-10
- Kromer, P., Abraham, A., Snasel, V., Berhan, E. and Kitaw, D. (2013) 'On the differential evolution for vehicle routing problem', in *IEEE (SoCPaR 2013): Fifth international conference of soft computing and pattern recognition*, Hanoi, Vietnam
- Kumbharana, S. and Pandey, G. (2013) 'Solving travelling salesman problem using firefly algorithm', *International Journal for Research in Science and Advanced Technologies*, Vol. 2 No.2, pp.53-57
- Laporte, G. (1992) 'The vehicle routing problem: an overview of exact and approximate algorithms', *European Journal of Operational Research*, Vo. 59 No.3, pp.345-358
- Li, F., Golden, B. and Wasil, E. (2007) 'A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem', *Computations Operational Research*, Vol. 34 No.9, pp.2734-2742

- Li, X., Tian, P. and Aneja, Y.P. (2010) 'An adaptive memory programming metaheuristic for the heterogeneous fixed fleet vehicle routing problem', *Transportational Research*, Vol. 46 No.6, pp.1111-1127
- Na, B., Jun, Y. and Kim, B.I. (2011) 'Some extensions to the sweep algorithm', *International Journal of Advanced Manufacturing Technology*, Vol. 56 pp.1057-1067
- Naji-Azimi, Z., and Salari, M. (2013) 'A complementary tool to enhance the effectiveness of existing methods for heterogeneous fixed fleet vehicle routing problem', *Applied Mathematical Modeling*, Vol. 37 No.6, pp.4316-4324
- Osaba, E., Carballedo, R., Yang, X.-S and Diaz, F. (2016) 'An evolutionary discrete firefly algorithm with novel operators for solving the vehicle routing problem with time windows', *Nature-Inspired Computation in Engineering*, pp. 1-20
- Osaba, E., Yang, X.-S., Diaz, F. and Onieva, E. (2017) 'A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy', *Soft Computing*, Vol. 21 No. 18 pp. 5295-5308
- Pan, F., Ye, C., Wang, K. and Cao, J. (2013) 'Research on the vehicle routing problem with time windows using firefly algorithm', *Journal of Computers*, Vol. 8 No. 9, pp.2256-2261
- Sajjadi, S.R. and Cheraghi, S.H. (2011) 'Multi-products location-routing problem integrated with inventory under stochastic demand', *International Journal of Industrial and Systems Engineering*, Vol. 7 No. 4, pp. 454-476
- Saraei, M. and Ali Ghaheri, S. (2017) 'An effective hybrid algorithm for vehicle routing problem by indicating capacity using genetic and firefly algorithm', *International Journal of Engineering Education*, Vol. 9 No. 2
- Sayadi, M.K., Hafezalkoton, A. and Naini, S.G.J. (2013) 'Firefly-inspired algorithm for discrete optimization problems: An application to manufacturing cell formation', *Journal of Manufacturing Systems*, Vol. 32, pp. 78-84
- Sayadi, M.K., Ramezani, R. and Ghaffari-Nasab, N. (2010) 'A discrete firefly metaheuristic with local search for makespan minimization in permutation flow shop scheduling problems', *International Journal of Industrial Engineering Computations*, Vol. 1, pp-1-10
- Seidgar, H., Abedi, M., Rad, S. T. and Rezaeian, J. (2016) 'An efficient hybrid of genetic and simulated annealing algorithms for multi server vehicle routing problem with multi entry', *International Journal of Industrial and Systems Engineering*, Vol. 24 No. 3 ,pp.333-360
- Simic, D., Kovacevic, I., Svircevic, V. and Simic, S. (2015) 'Hybrid firefly model in routing heterogeneous fleet of vehicles in logistics distribution', *Logic Journal of the IGPL*, Vol. 23 No. 3, pp.521-532
- Sivaran Kumar, V., Thansekhar, M.R. and Saravanan, R. (2014) 'Multi objective optimization for vehicle routing problem using fitness aggregated genetic algorithm (FAGA)', in *ICAIEA 2014: Proceedings of Second international conference on advances in industrial engineering applications*, Anna university, Chennai

- Taillard, E.D. (1999) 'A heuristic column generation method for heterogeneous fleet VRP', *RAIRO*, Vol. 33 No.1, pp.1-14
- Tarantilis, C., Kiranoudis, C. and Vassiliadis, V. (2003) 'A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem', *Journal of the Operational Research Society*, Vol. 54 No.1, pp.65-71
- Tarantilis, C., Kiranoudis, C. and Vassiliadis, V. (2004) 'A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem', *European Journal of Operational Research*, Vol. 152 No.1, pp.148-158
- Wang, G., Guo, L., Duan, H., Liu, L. and Wang, H. (2012) 'A modified firefly algorithm for UCAV path planning', *International Journal of Hybrid Information Technology*, Vol. 5 No. 3, pp. 123-144
- Wang, X., He, M. and Jiang, H. (2015) 'A discrete firefly algorithm for routing optimization of milk-run' in *ICADME 2015: Advances in Engineering Research* Vol. 39, Shenzhen, China, pp. 163-169
- Wu, Y., Ye, C.M., Ma, H.M. and Xia, M.Y. (2007) 'Parallel particle swarm optimization algorithm for vehicle routing problems with time windows', *Computer Engineering and Applications*, Vol. 43 No.14, pp.223-226
- Yang, X.S. (2008) *Nature-Inspired Meta heuristic Algorithm*, Luniver Press, pp. 83-96
- Yang, X.S. (2009) 'Firefly algorithms for multimodal optimization, in: *Stochastic algorithms: Foundations and applications*', Springer, pp.169-178
- Yang, X.S., (2010) *Engineering optimization: an introduction with metaheuristic applications*, Hoboken, New Jersey, Wiley