Uçuş Menzili ve Hizmet Süresi Kısıtları Altında Çoklu İHA Yönlendirme için Etkin Genetik Algoritma

Murat KARAKAYA[1], Ender SEVİNÇ[2]

[1]Bilg. Müh., Atılım Üni., Ankara, Türkiye
[2]Bilg. Müh, Ortadoğu Teknik Üni., Ankara, Türkiye
murat.karakaya@atilim.edu.tr, ender@ceng.metu.edu.tr

(Geliş/Received:15.07.2016; Kabul/Accepted:22.10.2016)
DOI: 10.17671/btd.15848

***Özet*—** Son zamanlarda gerek askeri gerekse sivil amaçlı İnsansız Hava Aracı (İHA) kullanımı giderek popülerlik kazanmaktadır. Ancak İHA'ların etkin kullanımı için gerekli Servis Kalitesi(QoS)'ni karşılaması için benimsenen yaklaşımlarının sınırlamaları vardır. İHA'lar arasındaki kıyaslamanın önemli sınırlarından biri de uçuş menzilidir. Çoğu zaman, İHA'ların, yeterli enerji kaynağı olmaması ve böylelikle nispeten kısa uçuş menzillerine sahip olmalarıdır. İlave olarak, İHA'ları kullanan uygulamalar için, verilen hizmet süresi zamanı içerisinde ziyaret edilmesi gereken pek çok nokta olabilir. Üstelik, uygulamalar tarafından yönlendirilecek İHA'ların sayısı da sınırlıdır. Bu nedenle, gerçek hayattaki uygulamalarda, önceden belirlenen zaman aralığı pencerelerinde daha fazla noktanın ziyaret edilmesi maksadıyla belirli bir uçuş menzili olan çok sayıda İHA için bir optimizasyon sorunu ile karşılaşılmaktadır. Bu problemde biz, kullanılan İHA sayısını en aza indirmek ve zaman aralığı içerisinde ziyaret edilecek nokta sayısını en üst düzeye çıkarmak istiyoruz. Bu nedenle, Genetik bir algoritma tasarladık ve çeşitli uçuş menzilleri, servis zaman aralıkları ve nokta ağ topolojileri için kapsamlı simülasyon testleri yoluyla etkinliğini doğruladık. Daha da ötesinde, tasarlanan algoritma ile arzu edilen başarıyı sağlayabilecek rakip bir algoritma ile de sonuçlar karşılaştırılmıştır.

***Anahtar Kelimeler*—** İnsansız Hava Araçları, Rota planlama, Genetik Algoritma, Optimizasyon, Simülasyon

An Efficient Genetic Algorithm for Routing Multiple UAVs under Flight Range and Service Time Window Constraints

***Abstract*—** Recently using Unmanned Aerial Vehicles (UAVs) either for military or civilian purposes is getting popularity. However, UAVs have their own limitations which require adopted approaches to satisfy the Quality of Service (QoS). One of the important limitations of the UAVs encounter is the flight range. Most of the time, UAVs have very scarce energy resources and, thus, they have relatively short flight ranges. Besides, for the applications using UAVs, there could be many customers to be serviced for the given service time windows. Moreover, the number of UAVs managed by the applications is also limited. Therefore, in real life applications, we face with an optimization problem such that for a given number of UAVs with a specific flight range, they should be servicing more customers in the predetermined time windows. In this problem, we would like to minimize the number of used UAVs and maximize the number of serviced customers meeting the time window requirement. For this reason, we have designed a Genetic Algorithm and validated its effectiveness via extensive simulation tests for various flight ranges, time windows, and customer topologies. Furthermore, the results of the proposed algorithm with a rival algorithm supporting the expected success have also been compared.

***Keywords*—** Unmanned Aerial Vehicles, Route planning, Genetic Algorithm, Optimization, Simulation

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) can be defined as "a power-driven, reusable airplane operated without a human pilot on board" [1]. UAVs are designed to be employed in the situations such that it is either dangerous or ineffective having human pilots on board. While first examples of employment of UAVs are designed for military purposes, thanks to development in mechanics and electronics along with computer engineering, nowadays, UAVs are employed in many different civil applications which are gathered under the concept of remote sensing. For example, in precision agriculture (PA), geospatial techniques and sensors are used to identify variations in the field and UAVs are convenient tools for collecting such sensor data for PA applications [8, 9, 10]. Other UAV application areas can be listed as: forest health monitoring, wildlife surveys, avalanche patrols, air quality monitoring, groundwater discharge monitoring, etc. [11].

Despite the recent developments and improvements in enabling technologies, UAVs have still several restrictions. One of the critical aspects of UAVs is the limited flight time [2] due to scarce energy resources ported on UAV's platform. Therefore, planning UAV routes is critical for their effective usage. For example, ineffective route planning can cause to use excessive number of UAVs or take excessive amount of flight range and time. As a consequence, operational cost can increase or, even get worse so that the mission cannot be accomplished at all.

Moreover, applications have their own Quality of Services constraints. For example, in some remote sensing applications, UAVs are expected to be over some Point of Interests (PoIs), e.g. location of customers, military targets etc., for the given time window. Considering the number of PoIs, the travel time among them and the speed and flight range of deployed UAVs; it is not a simple task to schedule each UAV to these PoIs such that each PoI is served within the time window. In fact, this problem originating from TSP is a well-known NP-Hard problem [23, 26].

In this work, we proposed an iterative Genetic Algorithm (GA). In each iteration a route covering maximum number of PoIs and satisfying the application constraints is generated for a single UAV. Then, the PoIs included in this route are excluded from the existing PoI list. Afterwards, if there exists any unvisited PoI, we iterate the process again. This iterative process continues until all PoIs are scheduled. Then, we have a number of routes for each UAV such that all the given PoIs are covered under the given constraints.

In the proposed GA, each possible route is modeled as an individual (chromosome) for the given UAV respecting given Flight Range (FR) and Time Window (TW) constraints. At the beginning of any iteration, GA creates an initial population of individual by randomly selecting PoIs. Afterwards, crossover and mutation operations are applied on the initial population to create a new generation. The solution which covers more PoIs with less route distance is selected as the generation-best solution. The generation-best solution is compared with the best solution found so far, and if it is better the best solution found so far, the best solution found so far is updated with it. Then, this generation is also undergone crossover and mutation operations once more to generate the next generation as described above. The best solution found so far is always copied to the next generation as a requirement of elitism principle. Generation creation continues until a termination condition is met. When GA terminates, it outputs the best route found so far..

To evaluate the success of the proposed method, another approach proposed in [19] is used. In the alternative solution, route construction is based on the Nearest Neighbor (NN) heuristic. In the NN heuristic, route is created by selecting the nearest PoI to the UAV location. The nearest PoI must be in a distance from the UAV such that the UAV can arrive it during the given time window. Moreover, the UAV's remaining flight range after visiting this PoI must be enough to return to the base. The details of application of the NN heuristic into the problem in question are presented in Section 4.5.

We implemented both solutions using Java and compared them in numerous experiment tests with different parameter settings and various benchmark problem data files [12]. The success of the proposed GA method is observed in the simulation results by decreasing the number of UAVs and the travelled distance considerably compared to the NN heuristic in [19].

2. RELATED WORK

There are several proposed algorithms developed for the UAV route planning [28] in the literature. In one of our previous works, we proposed a method to minimize the number of UAVs used in a battle scenario to cover all the targets considering only the flight range [3]. In a follow up work, we designed an Ant Colony Optimization (ACO) based solution for creating routes to cover maximum number of targets when covering all of them is not possible. In another work, two different algorithms to create routes which avoid flying over risky territories [5, 6] are proposed. We proposed a novel routing algorithm which considers the mobile base, e.g. an aircraft carrier, in [4]. In [7], it has been developed an improved Genetic Algorithm (GA) for planning a route considering the flight range and service time window limitations for a single UAV whereas in [19] we developed a heuristic solution for the multiple UAVs. In this work, we aim to create more effective routes for multiple UAVs than those of in [19] by designing a novel Genetic Algorithm (GA) solution.

The problem of UAV routing problem resembles to two commonly known problems in the literature: Travelling Salesman Problem (TSP) and Vehicle Routing Problem

(VRP). In both of these problems, main goal is to schedule a mobile (salesman or vehicle) to visit some locations (cities or depots) considering the given constraints (distance, range, time, time window, cost, etc.). In this work, the mobile is a UAV, locations are PoIs, and, the main constrains are the UAV's flight range (FR) and the visit time window (TW). With respect to these basic assumptions, some versions of TSP and VRP and theier proposed solutions are worth mentioning.

The Travelling Salesman Problem (TSP) is the simplest form of a routing problem where the goal is finding the shortest path between a number of customers, with the constraints of visiting each customer once and ending the tour where it started. In Multiple Travelling Salesman Problem (mTSP), there are more than one salesman to be considered. The most basic form of a VRP can be considered as the direct successor of the TSP, where there is one vehicle and one depot, the rules are visiting all customers once and only once, and ending the route at the depot it started. In some VRP, each customer needs to be served within a given time interval. Such problems are named Vehicle Routing Problem with Time Windows (VRPTW) [29]. VRPTW is closely related with the presented problem and resembles a basic form of that. Both problems consider that multiple mobiles to visit locations according to the given time windows. Therefore, we would like to discuss the solutions for VRPTW in details below.

There are numerous solutions proposed for VRPTW in the literature. These solutions can be classified into two main groups: *Exact* solutions and *Heuristic-based* solutions. Exact solutions are based on some well-known methods such as; Branch-and-Price, Branch-and-Price-and-Cut, Pure Branch-and-Cut Algorithms, etc. Though Desrochers, Desrosiers, and Solomon [20] were the first to propose an exact algorithm capable of solving VRPTW instances of respectable size (up to 100 customers), finally Irnich and Villeneuve [21] devised a k-cycle elimination procedure for k-path inequalities as a Branch-and-Price Algorithm. Then Jepsen et al. [22] showed how cuts can be used in a branch-and price method for the VRPTW. This method being the most common one has an improved final version which uses tabu search method for heuristically solving the sub problems and rapidly generating negative reduced cost columns. Desaulniers, Lessard, and Hadjar [23] showed the efficiency of branch-and-price-and-cut approaches applied to the VRPTW.

Kallehauge, Boland and Madsen [24] is known to present the latest study as an example for Pure Branch-and-Cut Algorithms. In that study, the algorithm is based on an arc-flow formulation of the problem and typically require, for obtaining good lower bounds, a large number of valid inequality families.

Beside the exact solutions, there are heuristic-based solutions in the literature. These are, among the others, 2-interchange method, Simulated annealing, Tabu search, and Genetic algorithms. For example, Cordeau et al. proposed a simple and flexible tabu search algorithm for periodic and the multi-depot vehicle routing problems with time windows [25]. In another example work, G.B. Alvarenga et al. proposed one of the latest GA studies in which the behavior of a heuristic method is compared against exact methods, and global optimal solutions are found using by truncated calculation and total distance minimization [26].

All the solutions mentioned above to the VRPWT assume that the mobile (vehicle) can wait if it arrives earlier to a given location (depot) compared to the time window and this waiting duration does not have any effects on the travel distance of the vehicle and, hence, the vehicle range. This may be an acceptable assumption for the vehicles moving on the ground. However, a flying vehicle (e.g. UAV) has two options for waiting for the time window to open. Either UAV can continue fly around for passing the time or it can land on, wait, and take off again. The second option is not always feasible since not all the times territory is available for landing. On the other hand, the first option causes consuming energy which ultimately affects the flight range (FR) of the UAV. In essence, the solutions proposed for VRTTW cannot be directly applied for cases when UAVs are the vehicles. Thus, in this work, we propose a genetic algorithm to consider the effect of waiting duration on the UAV flight range in the route planning as described below.

3. PROBLEM DESCRIPTION

We assume that the coordinates and the visit time window of each Point of Interest (PoI) along with the flight range (FR) and the number of UAVs are given. The time window (TW) determines the ready and due times in which the UAV must visit an associated PoI. That is, the UAV must service to PoI after the ready time and before the due time given in TW of the PoI. Furthermore, we assume that UAV takes off and lands on the same stationary base. Additionally, flight speeds and ranges of all UAV are supposed to be fixed and the same.

The problem is to generate UAV routes such that (*i*) all the PoIs are visited, (*ii*) all visits are during the requested TW, (*iii*) each UAV's total route distance has to be equal or less than the specified flight range, (*iv*) the total number of UAVs is minimized. Thus, target function is to minimize the number of UAVs used in visiting all the PoIs respecting the given TW and FR constraints. In this work, we call this problem as Covering All PoIs with Minimum Number of UAVs Problem (CAP/MNU).

In order to visit any PoI within the given TW, a UAV may have to wait for some time before visiting PoI if UAV arrives earlier than the ready time. This time period is termed as "waiting time". In order to service to any PoI within the ready and due times, the UAV might need to delay on the route. This delay may be either in the air or as well be on the ground which saves energy consumption. In this work, we assume that UAV continues to fly during the waiting time and, thus, consumes its FR. In a nut shell, the PoIs included in a route must be visited between specified ready and due times of TW.

3.1. Genetic Algorithm (GA)

In the proposed algorithm, we use an evolutionary approach, namely, Genetic Algorithm (GA). GA is defined as search procedures based on the mechanics of natural selection and genetics – at least in some qualitative sort of way [13]. Thus, GA is an evolutionary algorithm which is an iterative and stochastic method that works on a set of individuals (population) [17]. The objective of Genetic Algorithm is to develop method and theory to allow the design of GAs that solves hard problems quickly, reliably and accurately [13, 14, 15].

In order to apply a genetic algorithm to a problem, first every potential solution is encoded into artificial chromosome which consist of genes. While each gene stores the PoI id to be visited and visit time, the chromosome encodes the path in the order of genes. The essential idea is to preserve a population of chromosomes, which represent candidate solutions to the problem that progress over time through a process of mating to merge two solution chromosomes to produce a better solution. Solution is modeled as a chromosome by an encoding/decoding system. Generally, initial population (a set of potential solutions) is randomly generated or by using a constructive heuristic. A fitness function is the measure of the goodness of each chromosome in the population with regard to the problem at hand. Thus, the GA uses quantitative information for guiding the search process for finding best possible solution.

GA makes use of selection, crossover, and mutation operators. Each chromosome in the population is calculated an associated fitness value to choose competitive chromosomes that will form the next generation. After applying crossover operator on the individuals, GA produces a new population. Mutation operation is employed to ensure that a better (possibly optimal) solution not existing in the chromosome pool can also be randomly generated. Thus, GA carries on by creating successive generations of better and better individuals by applying these simple operations. Thus, finding an optimal solution will be guaranteed if the GA algorithm is run for a very long time to create new generations [15].

4. GA FOR SOLVING CAP/MNU

In this section, we provide the details of the Genetic Algorithm (GA) designed for generating a solution to the Covering All PoIs with Minimum Number of UAVs Problem (CAP/MNU). We adopted generic operators of GA for the UAV routing problem at hand. Therefore, we name the proposed algorithm “GA for covering All PoIs with Multiple UAVs” (GA-AP/MU). GA-AP/MU applies a greedy approach. In each iteration, GA-AP/MU calculates a path for the given UAV, then the next UAV route is calculated according to the remaining PoIs. Although the solution might not be optimal, it can be accepted as a greedy approach in a reasonable time of execution.

The solution construction begins with assuming that all UAVs are on the base and the start time is 0. If any of the UAVs delays its take off due to time window (TW) of the first PoI, it means that UAV does not consume energy and, thus, does not decrease its flight range (FR). Similarly, if the waiting in air is due to TW of the next PoI, then FR is decremented as the amount of waiting time.

GA-AP/MU is designed to achieve two objectives at the same time: first, to maximize the number of PoIs in each route, and second to minimize the number of assigned UAVs. The details of the phases in GA-AP/MU are presented below.

4.1. Fitness Value

Fitness value of a chromosome in the GA-AP/MU is inversely proportional to the route distance calculated according to the PoIs stored in the genes of that chromosome. The fitness value is evaluated for all chromosomes, i.e. UAV route, to find the good solutions. The fitness function is given in Eq.1.

$$\text{Fitness (Route)} = 1/(Travel\ distance) \qquad (1)$$

Though minimizing the traveled distance is the first goal of the proposed GA solution, maximizing the number of nodes (PoIs) visited in each route is the second goal in order to minimize the number of assigned UAVs. Therefore, in the GA-AP/MU, the selection of possible PoIs is done considering to use the most of the flight range of each UAV and to minimize the waiting times in air. Finally, GA is terminated when calculated fitness value does not improve during a predefined number of generations. The number of generation is determined according to given convergence ratio given Table 1.

4.2 Creating Initial Population

As seen in Fig.1, a UAV route is encoded into a chromosome whose structure is similar to the one proposed in [7]. Selected PoIs for a route and their planned visit times are stored in genes which constitute the chromosome. The visit time of the PoI also includes the waiting time in order to control and obey the specified ready and due times in TW.

Initial population of chromosomes is created randomly by considering the given constraints of CAP/MNU. Afterwards, each chromosome is validated and sorted according to the fitness value calculated as explained above. The initial population is then subjected to crossover and mutation operations as explained below.

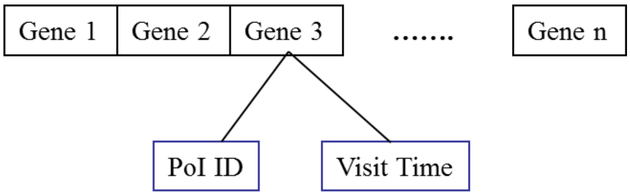

Figure 1. Employed Chromosome Structure (reproduced from [7])

4.3 Crossover

One of the important aspects of the proposed Crossover operator is that it should handle different-size chromosomes. The main goal of Crossover operator is to reach more PoI in a possible shorter path. We applied the developed crossover method in our earlier paper [7] into the Covering All PoIs with Minimum Number of UAVs Problem (CAP/MNU) as explained below.

Assume that Parent 1 (P1) and Parent 2 (P2) are the two chromosomes representing possible UAV routes. As seen in Fig.2, P1 is assumed to have 6 genes while P2 has 7 genes. Each gene has the same structure described in Fig.1. For example, 1st gene of P1 is "PoI23" is standing for PoI-23 in this study and "1" below PoI23 is standing for Visit Time which shows the time in minutes. Briefly, visit time shows the time for accessing that PoI. As the Flight Range (FR) is limited to 3 hours in the experiments, the visit time of the last gene of any chromosome will be smaller than 180 minutes in this study.

Due to the given CAP/MNU constraints, 1-point crossover is chosen for crossover implementation [7]. In order to achieve this, a cut point is found and decided in the chromosome due to crossover ratio. Crossover ratio is the ratio of the exchanged genes to the total. Although it seems to be an easy operation, the important point for making crossover between two chromosomes is finding a gene or PoI in common with the other parent.

If we find the same gene in two elected chromosome, it is decided as the cut point and from that PoI crossover is executed as described in Fig.1. However, if a common gene cannot be found, then another PoI, probably adjacent gene in the chromosome, is picked and then checked again. If that new PoI can be found in P2, then the crossover is done as stated above. Finally, if any common PoI in P1 and P2 cannot be found in the whole sequence of genes, then crossover cannot be executed for this pair and another pair of chromosomes is selected form the pool. One of the main advantages of this crossover operation is to look for a better solution which might possibly have either more PoIs in the path and/or the same number of PoIs with a shorter path.

By the help of crossover operation, the number of PoIs in a route can be increased to a possible extent; 7 genes for Offspring1 and 6 genes for Offspring2. At the same time, crossover operation will give us capability for decreasing possible "waiting in air" duration for the rest of the gene sequence. The resulting chromosome (Offspring1) will most probably have a shorter the path with decreased visiting times that originates from P2.

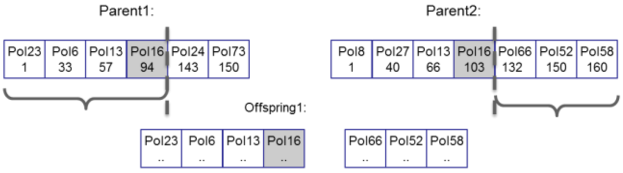

Figure 2. Crossover Operator

Offspring2 is created in the same way as of Offspring1 by exchanging the remaining halves of Parent1 and Parent2 in Fig.2. Since both offspring now have a different ordering of PoIs, they are validated to update their visit time according to the UAV flight speed and flight range as explained in Section 4.5. If any offspring cannot pass the validation test, then a new crossover point is searched. If all the possible crossover points fail, this pair of chromosomes is ignored and a new pair is selected.

4.4 Mutation

Mutation operation in the GA-AP/MU has two main functionalities. One of them is to work similarly due to mutation ratio as in common GA [16, 17]. Briefly, two genes are picked randomly in the chromosome, then exchanged and swapped to each other's place in order to look for whole search space. The other functionality of the GA-AP/MU mutation operator is believed to be more useful and valuable for the algorithm. Here, we look for a new PoI and try to add one more PoI/gene to chromosome. If possible, then the selected new PoI is inserted into the correct position in the chromosome. The insertion must be done without violating ready/due times of any PoI. This process helps to increase the number of PoIs in a route.

In summary, mutation operator either makes a swap of two genes or adds a new PoI into the chromosome as stated above. After any mutation operations, the new chromosome is validated as discussed below.

4.5 Validation

In validation operation**,** the visit time in the genes of the given chromosome is updated according to the flight speed and range of UAV along with the time windows of the PoIs. Thus, as shown in Fig.2, genes are assigned with the updated PoI visiting times. Validation operation is carried on for the chromosomes which are undergone the crossover or mutation operations, since these operations scramble the order of the genes in the chromosome. If any constraint, i.e. flight range, ready-due times of the PoI, is found to be violated for the given chromosome, then crossover or mutation operation is reverted back as discussed above.

4.6 NEAREST NEIGHBOR (NN) Heuristic for solving CAP/MUP

Nearest Neighbor (NN) heuristic is a simple to use, nonetheless effective for specific topologies [27]. In generic NN heuristic, one selects the nearest PoI as the next one. However, in the present study, there is another main constraint that comes up when UAV visits each PoI. This problem is called *Vehicle Routing Problem with Time Windows (VRPTW)*[20-26]. As this problem classified as a combinatorial optimization and integer programming problem [24], we use a greedy approach in order to reach a reasonable solution in an acceptable time period in this study.

We adopted the NN method as our first heuristic base solution to the CAP/MNU in [19]. In [19], route planning begins from the base and continues with the nearest PoI complying with the time window (TW) of the to-be-visited PoI. That is, NN heuristic eliminates the PoIs whose ready and due times do not fit to the UAV's arrival time to them. As the final step, the NN method selects the nearest PoI to the current one having providing that the UAV can have enough remaining flight range (FR) for returning to the base.

However, during simulation experiments in [19], we observed that in many cases the nearest PoI might have a rather late ready time than that of the second or the third closest PoI. This causes "*waiting in air*" situation which decreases the FR and consequently, the performance. In fact, the trade-off between *"flight distance" and "waiting in air"* is believed to be a good metric of greedy approach. Therefore, we improved the NN heuristic as follows. In the modified NN heuristic, the nearest PoI might not be selected if its TW causes long waiting in air. Instead, we first select a limited number of the closest PoIs to the current location of UAV. Then we can choose the PoI in that set with the earliest ready time. In, this work, this selection is limited to the closest three PoIs. That is, the improved NN heuristic, called the NN heuristic for Maximum PoI/ Multiple UAV (NN-MP/MU), selects one of the three closest PoIs set according to the earliest ready time. In [19], it has been shown that NN-MP/MU is able to visit more PoIs than that of the NN heuristic. This topic can be examined as a future work for NN Algorithm as well.

5. SIMULATION TESTS AND RESULTS

In this section, GA-AP/MU and NN-MP/MU algorithms are compared using different VRPTW benchmark problem data files [12] along with various UAV flight ranges.

5.1 Simulation Setup and Parameters

We have used both R and C data sets described in [12] in order to observe the effect of different topologies and time windows over the proposed solutions. In the experiments, R data sets, R101 thru R110, and C data sets, C101 thru C109 are used. The important GA-AP/MU parameters and their values are given at Table 1 whereas simulation parameters and their default settings are provided at Table 2.

Table 1. Parameters used in GA

Parameter	Value
Initial Population	400
Truncate Ratio	50%
Convergence ratio	95%
Crossover Type	Truncate, 1-point
Crossover Ratio	0.6 - 0.5
Mutation ratio	0.01 (1%)

For the UAV parameter values, MQ-1B Predator's specifications [18] are taken into consideration. We assume that average speed of the UAV can be 130 km/h and 165 km/h, and it can fly for 3 hours uninterruptedly. Therefore, chosen data sets have been experimented with two different speed/range combinations as explained below:

- ***Lower Flight Range (LFR): 130 km/h * 3h= 390 km.***
- ***Higher Flight Range (HFR): 165 km/h *3h= 495 km.***

Table 2. Simulation parameter settings

Parameter	**Default Value**	**Range**	**Notes**
Data Set	R101	R101-R112, C101-C109 RC100-RC108	29 Different VRPTW benchmark problems (All 100 series problems are taken from [12])
Number of PoIs	100	100	First of the 101 locations in the data set is selected as Base, and the rest is assumed to be PoIs to be visited
UAV Flight Range	Lower	Low Range: 390 km High Range: 495 km.	MQ-1B Predator UAV specs

5.2 Results of Experiments

In the following discussions, we provide the results of experiment test according to Flight Range (FR) parameter values. The results for the Lower Flight Range (LFR) and for the Higher Flight Range (HFR) are presented at Table 3 and 4 respectively. The performance improvement of the proposed GA-AP/MU over the alternative NN-MP/MU is presented at Table 5. All the test results given in the following tables and figures are obtained by taking the mean of the results of 10 independent runs. The best values in the comparison Tables 3, 4, and 5 are given in bold fonts.

As seen at Table 3, in 24 out of 29 data sets, GA-AP/MU employs fewer UAVs than NN-MP/MU. Only for R104 data set, both methods assign the same number of UAVs. Furthermore, with respect to the total distance travelled, for all data sets, GA-AP/MU generates shorter routes that these of NN-MP/MU. As a result, we can conclude that for the Lower Flight Range (LFR), the GA-AP/MU algorithm can create routes which employ less number of UAVs with shorter flight distance compared to NN-MP/MU algorithm.

Similarly, the results for Higher Flight Range (HFR) presented in Table 4 show the success of the proposed solution over the alternative one. In 22 out of 29 cases, GA-AP/MU generates routes which employ fewer numbers of UAVs than that of NN-MP/MU. In the remaining 7 cases, both algorithms assign the same number of UAVs. On the other hand, in terms of the total distance travelled, GA-AP/MU produces shorter routes than NN-MP/MU in all cases. Thus, as observed for the LFR, GA-AP/MU is better than NN-MP/MU considering the number of UAVs and the total traveled distance in HFR as well.

Table 3. Dataset Comparison in Lower Flight Range (LFR)

Datasets	**GA-MP/MU**		**NN-MP/MU**	
	# of UAVs for all PoI	Total Distance travelled	# of UAVs for all PoI	Total Distance travelled
R101	**8**	**1953,9**	10	2058,6
R102	**8**	**2057,2**	11	2076,1
R103	**7**	**1873,7**	11	2205,0
R104	7	**1594,0**	7	1722,7
R105	**6**	**1685,0**	9	2242,0
R106	**6**	**1827,1**	9	1987,1
R107	**6**	**1806,8**	9	1888,8
R108	**6**	**1566,6**	7	1773,0
R109	**5**	**1570,6**	6	1876,9
R110	**7**	**1788,5**	8	1941,0
R111	7	1741,8	7	2013,3
R112	**5**	**1341,3**	7	2005,4
C101	**14**	**3390,2**	23	3930,2
C102	**13**	**3102,5**	24	3782,8
C103	**12**	**2723,3**	25	3351,3
C104	**9**	**1978,6**	31	3049,2
C105	**11**	**3108,2**	16	3447,3
C106	**12**	**2903,8**	14	3267,5
C107	**11**	**2842,6**	13	3843,0
C108	**9**	**2299,6**	11	3301,2
C109	**8**	**2175,8**	9	3070,2
RC101	**7**	**1984,5**	9	2416,2
RC102	**7**	**2165,7**	9	2400,8
RC103	**7**	**2018,5**	9	2234,8
RC104	**7**	**1763,8**	9	2009,7
RC105	**9**	**2509,2**	10	2568,4
RC106	8	1866,3	8	2383,0
RC107	7	1917,8	7	2070,6
RC108	7	1655,0	7	2159,1

Table 5 summarizes the performance improvement of GA-AP/MU over NN-MP/MU. All positive values show the performance increase in percentage. Since NN-MP/MU could not overcome in any of the instances we don't have any negative values in Table 5. If examined, each data set has its own average scores just after finished and there is a final row which provides the average performance of all data sets used in the experiments. Considering the LFR, GA-AP/MU uses about 38% less number of UAV with travelling 20% less distances compared to NN-MP/MU on the average of all data sets. Similarly, for the HFR, GA-AP/MU routes about 44% fewer UAVs to travel 24% less distances compared to NN-MP/MU on the average considering all the data sets.

Another observation can be done according to the data sets used in experiments. As reported in [12], R problem sets are composed of randomly generated geographical data whereas in C problem sets, geographical data is created to form clusters. As the PoIs are clustered in C data sets, NN-MP/MU algorithm is expected to produce better results due to the nature of the NN heuristic [20-26]. On the contrary, for both FRs, GA-AP/MU outperforms NN-MP/MU in C data sets far better than R data sets. In Table 5, the averages according to the data set type are also provided. For instance, for HFR, the average performance improvement in number of used UAVs for R data sets is 24% whereas for C data sets, it is about 90%. Likewise, for LFR, GA-AP/MU uses 30% less number of UAV for R data sets and 68% less number of UAV for C data sets. The performance improvements in the total distance travelled supports these observations as well.

Table 4. Dataset Comparison in Higher Flight Range (HFR)

Datasets	**GA-MP/MU**		**NN-MP/MU**	
	# of UAVs for all PoI	Total Distance travelled	# of UAVs for all PoI	Total Distance travelled
R101	**7**	**2136,2**	9	2282,4
R102	**7**	**1879,7**	11	2492,3
R103	**7**	**1938,5**	8	2023,8

R104	**5**	**1482,9**	7	1920,9
R105	**5**	**1749,6**	7	2314,4
R106	**6**	**1861,4**	9	2287,2
R107	**5**	**1754,7**	7	1896,3
R108	5	**1437,0**	5	1717,2
R109	5	**1645,1**	5	1926,9
R110	5	**1603,5**	5	1687,9
R111	**6**	**1848,3**	7	2090,1
R112	5	1489,8	5	1864,5
C101	**11**	**3247,1**	20	3865,7
C102	**11**	**3028,1**	19	3511,0
C103	**9**	**2491,0**	23	3505,7
C104	**8**	**1862,0**	31	3086,3
C105	**10**	**3017,3**	15	3454,4
C106	**10**	**3024,9**	17	3670,4
C107	**8**	**2799,8**	12	3610,6
C108	**8**	**2284,6**	10	3447,0
C109	**7**	**2073,2**	8	3032,3
RC101	**6**	**2104,7**	7	2497,6
RC102	**6**	**2372,2**	9	2419,0
RC103	**6**	**2027,3**	8	2237,4
RC104	6	1781,9	6	2651,8
RC105	8	2427,7	8	2489,1
RC106	**5**	**1879,1**	7	2505,0
RC107	**6**	**1952,4**	7	2447,3
RC108	6	1825,1	6	2061,7

Table 5 provides the information about the effect of flight range parameter on the performance metrics. As the overall averages indicate that with Higher Flight Range, GA-AP/MU algorithm outperforms NN-MP/MU far better compared to Lower Flight Range, almost this difference reaches up to close levels of doubling it. This is because of evolving nature of GA-AP/MU algorithm can create better performance when the flight range is longer. However, in NN-MP/MU, the nearest PoI might be degradation in terms of distance and possible waiting duration in the air. Because of that reason, it can be inferred that NN-MP/MU has a greater local minima problem when compared to GA-AP/MU algorithm.

Table 5. Overall Performance Improvement of GA-AP/MU over NN-MP/MU

Datasets	**Lower Flight Range**		**Higher Flight Range**	
	UAV used	Distance Ratio	UAV used	Distance Ratio
R101	25,00%	5,36%	28,57%	6,84%
R102	37,50%	0,92%	57,14%	32,59%
R103	57,14%	17,68%	14,29%	4,40%
R104	0,00%	8,07%	40,00%	29,54%
R105	50,00%	33,06%	40,00%	32,28%
R106	50,00%	8,76%	50,00%	22,88%
R107	50,00%	4,54%	40,00%	8,07%
R108	16,67%	13,18%	0,00%	19,50%
R109	20,00%	19,50%	0,00%	17,13%
R110	14,29%	8,53%	0,00%	5,26%
R111	0,00%	15,59%	16,67%	13,08%
R112	40,00%	49,51%	0,00%	25,15%
R Set Ave	**30,05%**	**15,39%**	**23,89%**	**18,06%**
C101	64,29%	15,93%	81,82%	19,05%
C102	84,62%	21,93%	72,73%	15,95%
C103	108,33%	23,06%	155,56%	40,73%
C104	244,44%	54,11%	287,50%	65,75%
C105	45,45%	10,91%	50,00%	14,49%
C106	16,67%	12,52%	70,00%	21,34%
C107	18,18%	35,19%	50,00%	28,96%
C108	22,22%	43,56%	25,00%	50,88%
C109	12,50%	41,11%	14,29%	46,26%
C Set Ave	**68,52%**	**28,70%**	**89,65%**	**33,71%**
RC101	28,57%	21,75%	16,67%	18,67%
RC102	28,57%	10,86%	50,00%	1,97%
RC103	28,57%	10,72%	33,33%	10,36%
RC104	28,57%	13,94%	0,00%	48,82%
RC105	11,11%	2,36%	0,00%	2,53%
RC106	0,00%	27,69%	40,00%	33,31%
RC107	0,00%	7,97%	16,67%	25,35%
RC108	0,00%	30,46%	0,00%	12,96%
RC Set Ave	**15,67%**	**15,72%**	**19,58%**	**19,25%**
Total Ave	***38,08%***	***19,94%***	***44,38%***	***23,67%***

As a conclusion, one can argue that GA-AP/MU is robust to changes in underlying PoI topology, given time windows, and flight ranges.

6. CONCLUSION

In this work we have extended the UAV routing problem previously defined in [7] and [19] by considering multiple UAVs with flight range and time window constraints. We have designed an iterative Genetic Algorithm (GA) to generate routes using fewer numbers of UAVs with less total travel distances. The proposed GA is compared with the enhanced NN heuristic for different flight ranges, time windows and underlying topologies. Considering the experiment results, we can argue the success of the proposed algorithm over the enhanced NN heuristic.

As a future work, we would like to improve the algorithm by considering all the UAVs together which might enable us to achieve better optimization. Moreover, we would like to extend the constraints by adding service time constraints as well.

REFERENCES

[1] Chao, HaiYang, YongCan Cao, and YangQuan Chen. "Autopilots for small unmanned aerial vehicles: a survey." International Journal of Control, Automation and Systems 8.1 (2010): 36-44.

[2] Waharte, Sonia, and Niki Trigoni. "Supporting search and rescue operations with UAVs." Emerging Security Technologies (EST), 2010 International Conference on. IEEE, 2010.

[3] Murat Karakaya, "En Az Sayıda İnsansız Hava Aracı Kullanarak Sabit Hedeflerin Gozetlenmesinin Planlanması", 15. Otomatik Kontrol Ulusal Toplantısı ve Sergisi (TOK2013), (2013): 519-524.

[4] Murat Karakaya, UAV Route Planning For Maximum PoI Coverage, Computer Science & Engineering: An International Journal (CSEIJ), Vol. 4, No. 1, ISSN: 2231 - 329X, 2014.

[5] Hamdi Demirel, Halil Savuran, Murat Karakaya, "İnsansız Hava Araçları İcin Radar Kaplama Alanlarından Kacınacak En Kısa Rotanın Hesaplanması", 7.Savunma Teknolojileri Kongresi (SAVTEK-2014), 2014.

[6] Senem Aktas, Nergiz Kilinc, Hazan Daglayan, Ibrahim Cereci, Murat Karakaya, “UAV Route Planing for Avoidıng Enemy Radars”, The 1st International Conference on Engineering and Natural Sciences (ICENS 2015), Skopje, Macedonia, 2015.

[7] Ender Sevinc, Murat Karakaya, Maximizing UAV PoI Coverage under Flight Range and PoI Service Time Constraints, Lecture Notes on Software Engineering, Vol. 3, No. 4, 290-294, ISSN: 2301-3559, 2015.

[8] Zhang, Chunhua, and John M. Kovacs. "The application of small unmanned aerial systems for precision agriculture: a review." Precision agriculture 13.6 (2012): 693-712.

[9] Fornace, Kimberly M., et al. "Mapping infectious disease landscapes: unmanned aerial vehicles and epidemiology." Trends in parasitology 30.11 (2014): 514-519.

[10] Ballesteros, R., et al. "Applications of georeferenced high-resolution images obtained with unmanned aerial vehicles. Part I: Description of image acquisition and processing." Precision Agriculture 15.6 (2014): 579-592.

[11] Watts, Adam C., Vincent G. Ambrosia, and Everett A. Hinkley. "Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use." Remote Sensing 4.6 (2012): 1671-1692.

[12] VRPTW benchmark problems, 2005, [WWW document; retrieved January 2016] http://w.cba.neu.edu/~msolomon/problems.htm

[13] Goldberg, David E. The design of innovation: Lessons from and for competent genetic algorithms. Vol. 7. Springer Science & Business Media, 2013.

[14] S.N.Sivanandam, S.N.Deepa, Introduction to Genetic Algorithms, Springer-Verlag Berlin Heidelberg, 2008.

[15] Sevinç, E., Coşar, A., An Evolutionary Genetic Algorithm for Optimization of Distributed Database, The Computer Journal, v.54/5, 2011

[16] Goldberg, David, The Design of Innovation: Lessons from and for Competent Genetic Algorithms. Norwell, MA: Kluwer Academic Publishers, Ch.1. ISBN 978-1402070983, 2002.

[17] S.N.Sivanandam, S.N.Deepa, Introduction to Genetic Algorithms, Springer-Verlag Berlin Heidelberg, 2008.

[18] UAV Specifications, [WWW document; retrieved January 2014] (MQ-1 Predator), http://www.bga-aeroweb.com/Defense/MQ-1-Predator-MQ-9-Reaper.html

[30]

[19] Murat Karakaya , Ender Sevinç, Planning Multiple UAVs to Visit Points of Interest Considering Flight Range and Service Time Constraints, The 2nd International Conference on Engineering and Natural Sciences (ICENS 2016), Sarajevo, Bosnia and Herzegovina, 2016.

[20] Desrochers, M., J. Desrosiers, M. M. Solomon. 1992. A new optimization algorithm for the vehicle routing problem with time windows. Oper. Res. 40 342–354.

[21] Irnich, S., D. Villeneuve. 2006. The shortest path problem with resource constraints and k-cycle elimination for k ≥ 3. INFORMS J. Comput. 18(3) 391–406.

[22] Jepsen, M., B. Petersen, S. Spoorendonk, D. Pisinger. 2006. A nonrobust branch-and-cut-and-price algorithm for the vehicle routing problem with time windows. Oper. Res. Forthcoming.

[23] Guy Desaulniers, François Lessard, Ahmed Hadjar, “Tabu Search, Partial Elementarity, and Generalized k-Path Inequalities for the Vehicle Routing Problem with Time Windows”, Transportation Science, Vol. 42, No. 3, August 2008, pp. 387–404

[24] Kallehauge, B., N. Boland, O. B. G. Madsen. 2007. Path inequalities for the vehicle routing problem with time windows. Networks 49 273–293.

[25] Cordeau, J-F.; Laporte, G.; Mercier, A., “A unified tabu search heuristic for vehicle routing problems with time windows”, Journal of the Operational Research Society, Volume 52, Number 8, 1 August 2001, pp. 928-936(9)

[26] G.B. Alvarenga, G.R. Mateus, G. de Tomic, A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows, Computers & Operations Research 34 (2007) 1561–1584.

[27] Sunil Arya, David M. Mount, Nathan S. Netanyahu, An optimal algorithm for approximate nearest neighbor searching fixed dimensions, Journal Journal of the ACM (JACM) JACM Homepage archive Volume 45 Issue 6, Nov. 1998 Pages 891-923

[28] Valavanis, Kimon P., and George J. Vachtsevanos. Handbook of Unmanned Aerial Vehicles. Springer Publishing Company, Incorporated, 2014.

[29] Olli Bräysy and Michel Gendreau, "Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms.", Transportation science, vol. 39, no. 1, pp. 104-118, 2005.