Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

The optimization of UAV routing problem with a genetic algorithm to observe the damages of possible Istanbul earthquake

Olası İstanbul depreminin hasarlarının gözlenmesi için İHA rotalama probleminin bir genetik algoritma ile eniyilenmesi

Muhammed HALAT[1], *Ömer ÖZKAN*[2*]

[1,2]Industrial Engineering Department, Turkish Air Force Academy, National Defense University, Istanbul, Turkey.
muhammed.halat.1730@gmail.com, o.ozkan@hho.edu.tr

Received/Geliş Tarihi: 26.07.2019 Revision/Düzeltme Tarihi: 24.06.2028 doi: 10.5505/pajes.2020.75725
Accepted/Kabul Tarihi: 28.06.2020 Research Article/Araştırma Makalesi

Abstract

In this study, the problem is to find a route for a UAV that takes off from Istanbul to observe the damages that may occur after the possible Istanbul earthquake within the first 24 hours. In the problem, 230 candidate grid points that UAV can visit on Istanbul are determined and the weight values combining the risk values based on earthquake degree zones and the population densities of the grid points are calculated for each candidate point. It is aimed to find a route for the UAV to maximize the total weights of the visited grid points under the UAV range constraint. The described problem is adapted to the Orienteering Problem in the literature. Since the Orienteering Problem is an NP-hard problem, a problem-specific genetic algorithm and a simulated annealing algorithm are developed to solve the problem. The parameters of the algorithms are tuned by experiments. 15 different scenarios including the daily number of visits (of taken images) and the airports that the UAV takes and lands off after the earthquake are created and tried to be solved exactly via ILOG and approximately via developed metaheuristics. While the optimal solutions are found for 2 of 15 scenarios via ILOG, the designed genetic algorithm has better solutions and can solve the problem within acceptable CPU times for the rest of the scenarios.

Keywords: UAV routing, Vehicle routing, Genetic algorithm, Orienteering problem, Istanbul earthquake, Disaster management.

Öz

Bu çalışmada, olası İstanbul depremi sonrası oluşabilecek hasarları havadan gözlemlemek için ilk 24 saat içerisinde İstanbul'dan kaldırılan bir İHA'nın hangi rotada uçması gerektiği problemi ele alınmıştır. Problemde, İstanbul üzerinde İHA'nın ziyaret edebileceği 230 aday grid nokta belirlenmiş ve her aday nokta için noktanın deprem riski ağırlığı ile nüfus yoğunluğunu birleştiren ağırlık değerleri belirlenmiştir. Problemde en fazla sayıda aday noktanın ağırlığını toplayacak şekilde İHA'nın menzil kısıtı altında rotalanması amaçlanmıştır. Tanımlanan bu problem, literatürdeki Oryantring problemine uyarlanmıştır. Oryantring problemi NP-zor bir problem olduğundan dolayı, problemin çözümü için probleme özgü bir genetik algoritma ve bir tavlama benzetimi algoritması geliştirilmiştir. Algoritmaların parametreleri deneyler ile ayarlanmıştır. Gerçek hayata uygun olarak deprem sonrası İHA'nın kalktığı havalimanı ile günlük ziyaret (veya görüntü sayısı) durumlarını kapsayan 15 farklı senaryo oluşturulmuş ve senaryolar ILOG ile kesin ve geliştirilen metasezgisel algoritmalar ile yaklaşık olarak çözülmeye çalışılmıştır. 15 senaryonun 2'sinde optimal çözüm bulunmuş olup diğer senaryolar için genetik algoritma daha iyi sonuçlar elde etmiş ve kabul edilebilir CPU süreleri içinde problemi çözebilmiştir.

Anahtar kelimeler: İHA rotalama, Araç rotalama, Genetik algoritma, Oryantring problemi, İstanbul depremi, Afet yönetimi.

1 Introduction

Istanbul connects two continents (Europe and Asia) and it is the largest city in Europe with over 15 million population. The Bosphorus is the name of the strait between the two continents and it unites the Black Sea and the Marmara Sea. As a result of this geological location, Istanbul has experienced major earthquakes throughout its history, because the North Anatolian Fault passes under the Marmara Sea. The researches point in 2002 that in the following 30 years a major earthquake under the Marmara Sea is expected with a 65% probability. There are approximately 800000 buildings in Istanbul and it is estimated that the expected earthquake will cause to demolish 35000-40000 buildings. The loss of life will range from 30000 to 40000. 70000 buildings will be heavily and 200000 will be moderately damaged. After the earthquake, 430000 to 600000 families will need shelters to live in. Just the damages of buildings will cost $11 billion. The damages in roads, bridges and other transportation systems, electricity, water, and gas distribution systems, communication systems, etc. will also cause to create higher risks [1],[2].

These results can be reduced by effective and efficient pre- and post-disaster management. Therefore this paper focuses on observing the damages of the possible Istanbul earthquake via an Unmanned Aerial Vehicle (UAV). UAV technology is growing rapidly and it is becoming cheaper to use the UAVs in all types of military and civilian missions. The UAV types are categorized as micro, light, medium, heavy and super-heavy weighted with fixed, single rotary or multi rotary wings. They also may have long (≥400 km.), medium (400-100 km.) or short (≤100 km.) flight ranges [3]. There are plenty of recent researches in the literature about the usage of UAVs for disaster management [4]-[8]. Most of the researchers focus on the conceptual usage of UAV, the image or video processing of a part of the land and there are also real implementations for small-sized areas with generally micro or light-weighted short-ranged UAVs for pre- and post-earthquake occasions.

[*] Corresponding author/Yazışılan Yazar

The main contribution of this paper is to solve a possible post-disaster case by using a long-range UAV for a quite big disaster area (i.e. Istanbul is 5461 km^2). The possible post-earthquake damages in Istanbul can be detected via routing that type of UAV. As the second contribution of this study to the literature that the UAV Routing Problem (UAVRP) is formulated based on the Orienteering Problem (OP) and a literature review of OP is prepared. The problem is tried to solved exactly. As the last utility, a problem-specific Genetic Algorithm (GA) and Simulated Annealing (SA) algorithm are developed to solve the problem and real data for Istanbul is generated and used.

The study is organized as follows: Section 2 gives the relevant literature about the OP and UAV routing. The problem definition and the mathematical model of the problem are described in Section 3. Section 4 covers the proposed SA and GA, and section 5 presents the case study of the possible Istanbul earthquake. Finally, section 6 concludes the paper.

2 Literature review

Travelling Salesman Problem (TSP), Vehicle Routing Problem (VRP) and OP are well known and studied transportation based problems and UAVRP is an extension of these optimization problems for flying vehicles. There is a considerable amount of literature about TSP, VRP, UAVRP, OP and their variants [9]-[14]. This paper focuses on the OP because the defined problem in this paper is an implementation of OP. Therefore, a detailed literature review for OP and its variants are presented in Table 1.

OP is defined by Golden et. al. [16] and it has NP-hard complexity [16]. Thence as seen from Table 1, there are exact, heuristic, metaheuristic, approximation algorithms to solve the OP and its variants. The review papers about OP can be found in [27],[50].

Table 1. A literature review for the OP.

Ref.	Year	Author(s)	Problem*	Method**	Ref.	Year	Author(s)	Problem*	Method**
[15]	1984	Tsiligirides	OP	H	[35]	2013	Özdemir	TOPTW	ABC
[16]	1987	Golden et. al.	OP	H	[36]	2013	Tarantilis et. al.	CTOP	BiF&F
[17]	1992	Ramesh	OP	B&B	[37]	2014	Angelelli et. al.	ClOP	B&C and TS
[18]	1996	Chao et. al.	OP	H	[38]	2014	Campos et. al.	OP	GRASP and PR
[19]	2001	Feillet et. al.	TSPP	-	[39]	2014	Cura	TOPTW	ABC
[20]	2002	Fomin and Lingas	OPTW	AA	[40]	2014	Evers et. al.	OPSWs	OPSWsH
[21]	2007	Boussier et. al.	TOP	B&P	[41]	2014	Hu and Lim	TOPTW	I3CH
[22]	2009	Righini and Salani	OPTW	DP	[42]	2014	Salazar et. al.	TOP	ALNS
[23]	2010	Bouly et. al.	TOP	MA	[43]	2014	Verbeeck et. al.	OPTW	ACS
[24]	2010	Tricoire et. al.	OPTW	VNS	[44]	2014	Zhang et. al.	SOPTW	VNS and VND
[25]	2011	Campbell et. al.	OP	VNS and DP	[45]	2015	Duque et. al.	OPTW	PA
[26]	2011	Labadie et. al.	TOPTW	GRASP and ELS	[46]	2015	Gavalas et. al.	AOP	AA
[27]	2011	Vasteenwegen et. al.	OP	-	[47]	2015	Gunawan et. al.	TOPTW	ILS
[28]	2012	Labadie et. al.	TOPTW	VNS	[48]	2015	Lin and Yu	TOPTW	SA
[29]	2012	Lin and Yu	TOPTW	SA	[49]	2015	Marinakis et. al.	OP	M-GRASP
[30]	2013	Afras and Labadie	TOP	CG and ELS	[50]	2016	Gunawan et. al.	OP	-
[31]	2013	Archetti et. al.	CTOP	B&P	[51]	2016	Yu et. al.	CorOP	MIQPbAA
[32]	2013	Dang et. al.	TOP	B&C	[52]	2017	Ostrowski et. al.	OP	EILIA
[33]	2013	Divsalar et. al.	OP	VNS	[53]	2018	Hu et. al.	TOPTW	CPMOEA/D
[34]	2013	Lin	TOP	MSA	[54]	2019	Tsakirakis et. al.	TOP	SHHS

* OP	: Orienteering problem	ClOP	: Clustered orienteering problem
TSPP	: Travelling salesman problem with-profits	OPSWs	: Orienteering problem with stochastic weights
OPTW	: Orienteering problem with time windows	SOPTW	: Stochastic orienteering problem with time windows
TOP	: Team orienteering problem	AOP	: Arc orienteering problem
TOPTW	: Team orienteering problem with time windows	CorOP	: Correlated orienteering problem
CTOP	: Capacitated team orienteering problem		
** H	: Heuristic	SAA	: Sample average approximation
B&B	: Branch & bound	OPSWsH	: Orienteering problem with stochastic weights heuristic
AA	: Approximation algorithm	I3CH	: Iterative 3-component heuristic
B&B	: Branch & price	ALNS	: Adaptive large neighborhood search
DP	: Dynamic programming	ACS	: Ant colony system
MA	: Memetic algorithm	VND	: Variable neighborhood descent
VNS	: Variable neighborhood search	PA	: Pulse algorithm
GRASP	: Greedy randomized adaptive search procedure	ILS	: Iterated local search
ELS	: Evolutionary local search	M-GRASP	: Memetic-GRASP
SA	: Simulated annealing	MIQPbAA	: Mixed integer quadratic programming based anytime algorithm
CG	: Column generation		
B&C	: Branch & cut	EILIA	: Evolution-inspired local improvement algorithm
MSA	: Multistart simulated annealing	CPMOEA/D	: Multi-objective evolutionary algorithm based on decomposition and constraint programming
ABC	: Artificial bee colony		
BiF&F	: Bi-level filter-and-fan	SHHS	: Similarity hybrid harmony search algorithm
TS	: Tabu search	PR	: Path relinking

In TSP, the salesman starts with a node, visits all other nodes (i.e. customers) just one time and turns back to the starting node. In VRP, the vehicle does not have to turn back to the starting node or visit all other nodes. On the other hand different then TSP and VRP in OP, there are scores (i.e. weights) for all nodes separately and the traveler (i.e. runner or UAV for the solved problem in this paper) tries to collect maximum total score by visiting some of the nodes under a time constraint. The runner may or may not have to return to the starting node. In this paper, we also preferred to solve the OP exactly by using an Integer Programming (IP) model and approximately by using two metaheuristics for a real case study.

3 Problem definition

The OP can be defined as a graph $G = (N, E)$ with a set of nodes (N) and a set of edges (E). Each node i has a score (i.e. weight) S_i. The t_{ij} is the euclidean distance between nodes i and j. The T_{max} is the range limit to finish the tour and it is not necessary to visit all nodes. The goal of the OP is to find a route that maximizes the total collected score from the one time visited nodes. The starting and ending node may or may not be the same (in this paper, the UAV turns back to the departure base).

The OP can be formulated as an IP model and it is presented below [27]. The following decision variables X_{ij} are used to represent the visited nodes, that X_{ij} = 1 if a node is visited, otherwise X_{ij}= 0. In the model, u_i and u_j are the positions of node i and j in the path that is used to prevent the sub-tours.

$$Max. \sum_{i=2}^{N-1} \sum_{j=2}^{N} S_i X_{ij} \tag{1}$$

s.t.

$$\sum_{j=2}^{N} X_{1j} = \sum_{i=1}^{N-1} X_{iN} = 1 \tag{2}$$

$$\sum_{i=1}^{N-1} X_{ik} = \sum_{j=2}^{N} X_{kj} \leq 1;\ \forall k = 2, \ldots, N-1 \tag{3}$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} t_{ij} X_{ij} \leq T_{max} \tag{4}$$

$$2 \leq u_i \leq N;\ \forall i = 2, \ldots, N \tag{5}$$

$$u_i - u_j + 1 \leq (N-1)(1 - X_{ij});\ \forall i, j = 2, \ldots, N \tag{6}$$

$$X_{ij} \in \{0, 1\};\ \forall i, j = 1, \ldots, N \tag{7}$$

The objective function (1) maximizes the total collected score. Constraint (2) ensure that the route starts in node 1 and ends in node N (1st and Nth nodes are the same nodes i.e. the base for this study). Constraint (3) guarantee the connectivity of the route and guarantee that every node is visited at most once. Constraint (4) ensures a limited UAV flight range. Constraints (5) and (6) are essential to prevent sub-tours.

4 The methodology

In this work, two metaheuristics (GA and SA) are developed to solve the problem. Their operators and parameters are described in the following subtitles.

4.1 The genetic algorithm

GA is a metaheuristic inspired by natural evolution [55]. The GA starts with multiple initial solutions (individuals) and this initial population evolves to better populations by the implementation of crossover, mutation, and selection (parental and environmental) operators. Better individuals create better children and in the end, the fittest solution can be reached. In this paper, since the OP is NP-hard [16], a problem-specific GA and SA are designed to solve the OP based UAVRP. GA is chosen because it is the most preferred and well-studied metaheuristic to solve the optimization problems in the literature. The pseudo-code of the proposed GA is presented in Figure 1.

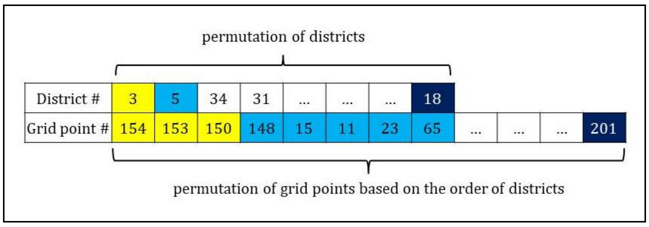
```
Begin
Set initial parameters ($\mu, \lambda, k, P_c, P_m, e, g_{max}$)
Generate $\mu$ # of individual (initial population)
Evaluate initial population by using $f$
$g = 0$     /* $g$: generation number */
repeat
    Apply tournament to select parents by using $k$
    Generate $\lambda$ # of new children with crossover by using $P_c$
    Apply mutation by using $P_m$
    Evaluate new children (new population) by using $f$
    Apply elitism by using $e$
    Apply environmental selection (update population)
    $g = g + 1$
until $g = g_{max}$
Show best solution
End
```

Figure 1. Pseudo-code of the proposed GA.

The following subtitles cover the basic operators and functions of the proposed GA based on Figure 1. The initial parameters presented in Figure 1 are also described in the following subtitles and they are tuned before implementing the case study.

4.1.1 Representation

The proposed GA uses a permutation-based representation. The representation has 2D permutation arrays and a sample is described in Figure 2.



District #	3	5	34	31	...	...	...	18			
Grid point #	154	153	150	148	15	11	23	65	...	...	...	201

Figure 2. The 2D array permutation representation sample.

The first array is the permutation of districts and the second array is the permutation of candidate grid points that the UAV visits. The second row is related to the first row and the order of the second row is convenient to the first one. For instance in Figure 2, the first element of the first row is district 3. In the second row, the first 3 candidate grid points are the points in district 3 (i.e. 154, 153, and 150 respectively). The second element of the first row is district 5 and the interval between the 4th and the 9th elements of the second row are the points that are located in district 5. The following cells in Figure 2 are obeying the same rule. As seen from Figure 2, since the number of the districts is less than the number of the candidate grid points, the length of the first row is shorter than the second row.

This representation is chosen because the OP based UAVRP is about finding a proper path for a UAV. Therefore, this representation helps not only see the order of the candidate grid points but also finds an efficient order of the districts to be visited. The following operators are using both rows to diversify the candidate solutions.

4.1.2 Fitness function

The fitness function (f) of the proposed GA is a normalized function of the original objective function (Eq.1) and it is presented in Equation (8).

$$Max. \sum_{i=2}^{N-1} \sum_{j=2}^{N} S_i X_{ij} \, / \sum_{i=2}^{N-1} S_i \, ; \; i,j = 1, \ldots, N \qquad (8)$$

In Equation (8), the denominator is the total score of all candidate grid points. The obtained value of the Equation (8) ranges between "0" and "1", therefore the calculated value provides a better approach to understand the total collected score by the visited candidate grid points via a UAV in a distance range constraint. When calculating an f of a candidate solution, the distances between the visited grid points without considering the neighborhood of the districts are summed until reaching the range limit according to the order of the grid points. If the range of the UAV is reached before the end of the second row, it is accepted that the UAV is only visited some of the grid points in the order. If the entire candidate grid points are visited for a candidate solution, then the f of the individual becomes "1", otherwise it is <1. The proposed GA does not produce any infeasible solution, but the length of the second row in the representation also does not change according to the individuals.

4.1.3 Initial population

The initial population of the proposed GA has μ number of solutions (i.e. μ=population size) and this parameter is tuned before case study. The initial individuals are created with a simple heuristic. Instead of using random solutions, it is preferred to use a heuristic, because it is important to create a balance between the diversification and intensification property of a GA. The random solutions help to diversify, but they generally have worse f value. The proposed heuristic uses neighborhood properties of the districts to also increase the intensification. The heuristic starts with selecting a random neighbor district of the base station (i.e. an airport). Then, the heuristic randomly selects a neighbor district of the previously selected district. The process continues until selecting all districts. If all of the neighbors of a selected district are already in the solution, the heuristic selects a random district and the procedure continues. After finding a proper permutation for the districts, the heuristic finds an order for the candidate grid points of the districts convenient to the permutation of the districts. The candidate grid points of a district in the order are sorted randomly as the second row. The heuristic helps to find good initial solutions and prevents the UAV to fly inefficiently between distant grid points.

4.1.4 Parental selection and crossover

After generating the initial population, the proposed GA determines the parents by using tournament selection. In tournament selection, the k numbers of randomly selected individuals encounter in a tournament and the best f valued winner individual becomes a parent. Then by using crossover probability (P_c), the order 1 crossover is implemented which is convenient to the permutation-based representations. A sample for the order 1 crossover is presented in Figure 3.

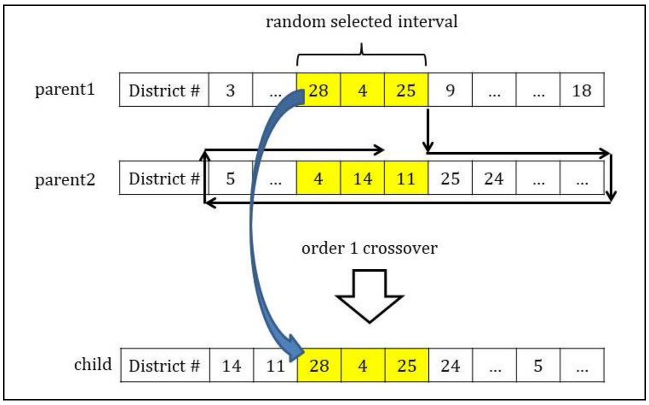


Figure 3. The order 1 crossover sample.

As seen in Figure 3, the order 1 crossover transfers a randomly selected interval (i.e. 28, 4 and 25) of the parent 1 with in the same position to the child. Then, it continues with the parent 2 from the following cell and copies the unvisited districts to the child until reaching to the end of the solution. When reaches the end, it turns back to the initial cells of the parent 2 and continues to transfer the unvisited districts to the child till the end. The order 1 crossover prevents to create an inconvenient child for the permutation representation. Finally, the candidate grid points of the parents are transferred to the child from the original parent that the gene (i.e. districts) is copied. And, the order of the second row is convenient with the first row (i.e. the order of the districts) in the produced child. The order 1 crossover does not change the internal order of candidate grid points in districts.

4.1.5 Mutation

The order 1 crossover generally tries to find the best route for the districts. However, the order of the candidate grid points is also important to reach better solutions. Therefore, two different mutations are used by the proposed GA to help to find better routes. The first mutation is district swap mutation and a sample for the mutation operator is covered in Figure 4.

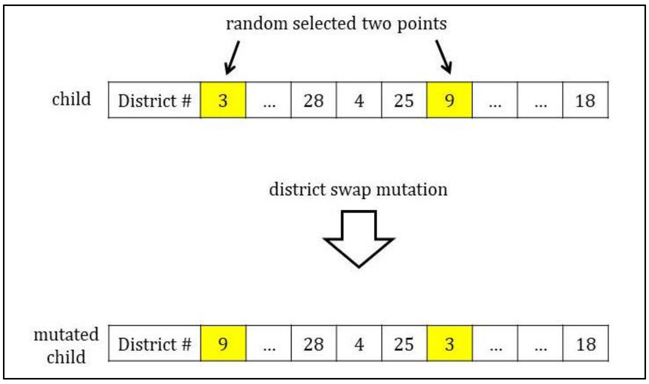


Figure 4. The district swap mutation sample.

The district swap mutation simply swaps the orders of the two random selected districts of a selected solution (i.e. 3 and 9 in Figure 4). The candidate grid points of the selected districts are also changing their orders according to the new arrangement.

The second mutation is candidate grid point scramble mutation and a sample for the created problem-specific mutation operator is covered in Figure 5.

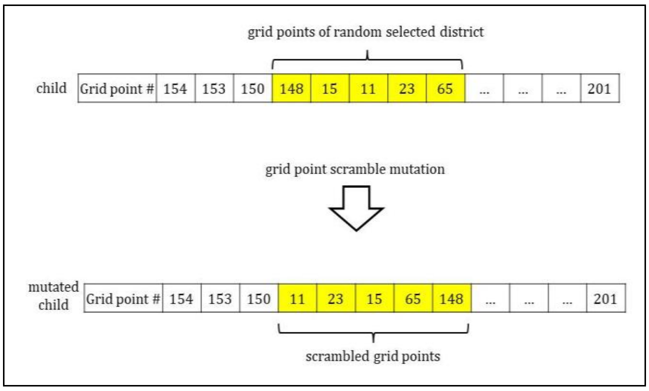


Figure 5. The grid point scramble mutation sample.

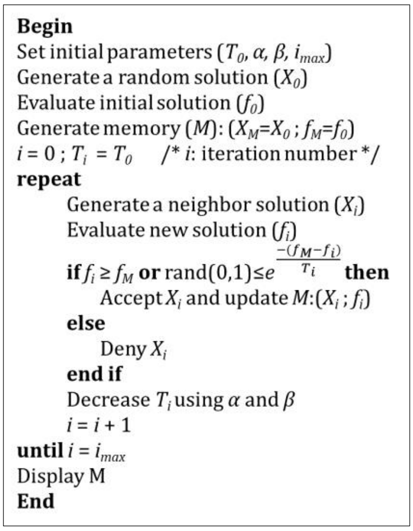
The grid point scramble mutation tries to find better sub-routes for the internal order of the districts. As seen in Figure 5, five grid points of a randomly selected district are scrambled randomly to find a new solution. The mutation probability (P_m) is used to select the children to be mutated and it is tuned before case study. The designed mutation operators are trying to make small changes in the order of the selected individuals to increase the intensification property of the proposed GA.

4.1.6 Environmental selection, elitism and stopping criteria

The proposed GA produces $\lambda = \mu - e * \mu$ number of new children in every generation where 0< e<1. In the environmental selection, the $e * \mu$ number of elitist (i.e. the best) solutions of the previous generation join the λ number of new solutions to generate the next generation. Therefore, μ stays constant during generations. The proposed GA stops when it reaches to g_{max} number of generations. The e and g_{max} parameters are also tuned before the case study.

4.2 The simulated annealing

The SA is a single solution based metaheuristic that can escape from local extremums by using a temperature parameter and a cooling schedule for this parameter [56]. The pseudo-code of the generated SA for the problem is illustrated in Figure 6.



```
Begin
Set initial parameters ($T_0$, $\alpha$, $\beta$, $i_{max}$)
Generate a random solution ($X_0$)
Evaluate initial solution ($f_0$)
Generate memory ($M$): ($X_M$=$X_0$ ; $f_M$=$f_0$)
$i$ = 0 ; $T_i$ = $T_0$     /* $i$: iteration number */
repeat
        Generate a neighbor solution ($X_i$)
        Evaluate new solution ($f_i$)
        if $f_i \geq f_M$ or rand(0,1)≤$e^{\frac{-(f_M-f_i)}{T_i}}$ then
            Accept $X_i$ and update $M$:($X_i$ ; $f_i$)
        else
            Deny $X_i$
        end if
        Decrease $T_i$ using $\alpha$ and $\beta$
        $i$ = $i$ + 1
until $i$ = $i_{max}$
Display M
End
```

Figure 6. Pseudo-code of the SA.

The second row of the representation in the GA (i.e. grid points) is used as the representation of the solutions in the SA. The fitness value calculation of the GA is the same for the SA. The initial temperature (T_0) of SA is calculated by a Markov chain-based algorithm proposed by [56]. In the algorithm, the T_0 is set at a temporary large number for the Markov chain at the start and after a full Markov chain is accomplished, T_0 is calculated by using the following Equation (9).

$$\gamma = \frac{m_1 + m_2 \, x \, e^{\frac{-\Delta f}{T_0}}}{m_1 + m_2} \tag{9}$$

In Equation (9), m_1 covers the number of moves that have resulted in an increased f value, m_2 represents the moves when the f value decreased relative to the previous step, and Δf represents the average f increase after m_2 moves. The acceptance ratio (γ) of the worse results is set to 95% at the start, it is decreased subsequently. The determined maximum iteration number (i_{max}) is divided into β parts equally and at the end of each division, the temperature is decreased to $T_{i+1} = \alpha * T_i$ $(i = 0,1,2, \ldots)$ beginning with T_0. The α, β, and i_{max} parameters are tuned before the case study.

5 The case study: The Istanbul earthquake

Istanbul is selected as the case study for the OP based UAVRP implementation to the post-earthquake situations. Because it is the most crowded city in Europe and the coming earthquake near Istanbul is inevitable. The degree of earthquake zones in Istanbul is presented in Figure 7.

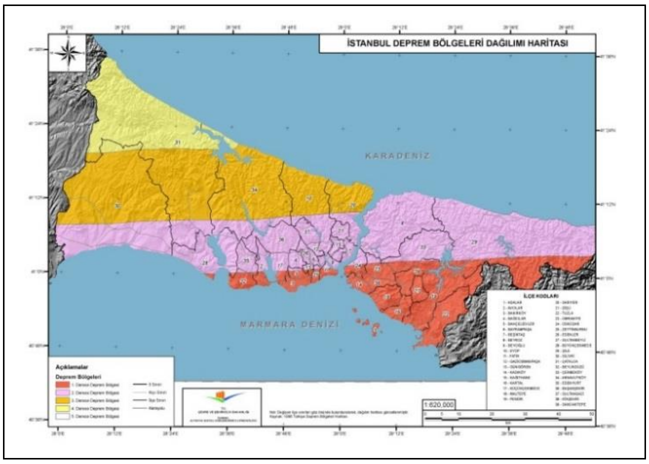


Figure 7. The degree of earthquake zones in Istanbul [57],[58].

As seen from Figure 7, four different zones have different earthquake degrees in Istanbul (from 1st to 4th degree and illustrated as red, purple, orange and yellow-colored, respectively). The risk of the zones decreases when the zone becomes away from the Marmara Sea.

The map in Figure 7 is placed on Google Earth software and the map is overlapped with the exact Istanbul map on Google Earth. The overlapped maps are presented in Figure 8.

By using the map in Figure 8, the exact locations of the earthquake zones are determined. Then, the Istanbul map is divided into grid zones which have 10 km. to 10 km. edge lengths (i.e. the grid zones are covering 100 km^2) and the GPS coordinates of the 230 number of candidate grid points in Istanbul are determined as presented in Figure 9. The airports (i.e. the base stations) in Istanbul are also marked as stars in

Figure 9 and their GPS coordinates are also determined via Google Earth. The GPS coordinates of the candidate grid points and the airports are used to calculate Euclidean distances between them.

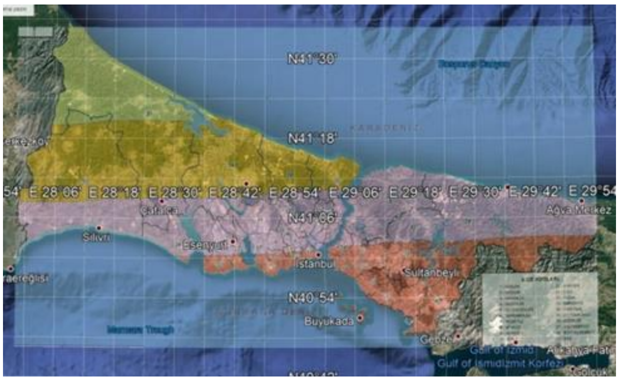


Figure 8. The overlapped Istanbul earthquake map on Google Earth.

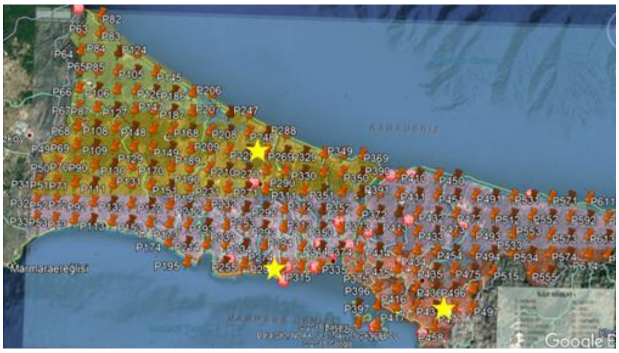


Figure 9. 230 candidate grid points (pins) and 3 base stations (stars, i.e. the airports in Istanbul).

The length of the edges of the grid zones is determined conveniently to the selected UAV which is named Bayraktar. The Bayraktar UAV is a long-ranged (3000 km.) Turkish made UAV and it is an appropriate UAV for that kind of mission. The Bayraktar can get the desired images and videos from that sized zones by flying in a proper altitude. A sample figure for the coverage of the grid zones by visiting the grid points in the center of the grid zones via a Bayraktar UAV is illustrated in Figure 10.

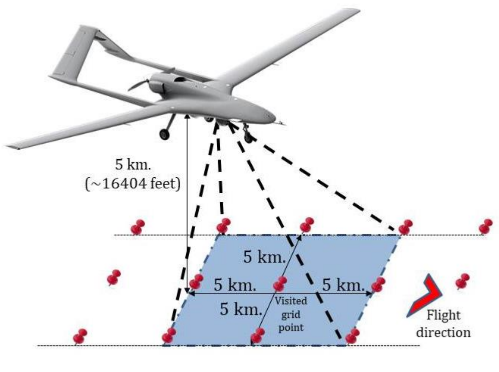


Figure 10. The coverage of a grid zone by visiting the grid point in the center (i.e. pin in the center of the covered area) via a Bayraktar UAV.

After determining the candidate grid points and their locations, the population and the earthquake risk values for each grid point are calculated. Istanbul has more than 15 million population and has 39 districts. Most of the population is close to the Marmara and that can increase the losses for a possible earthquake. The density of the population in districts is illustrated in Figure 11.

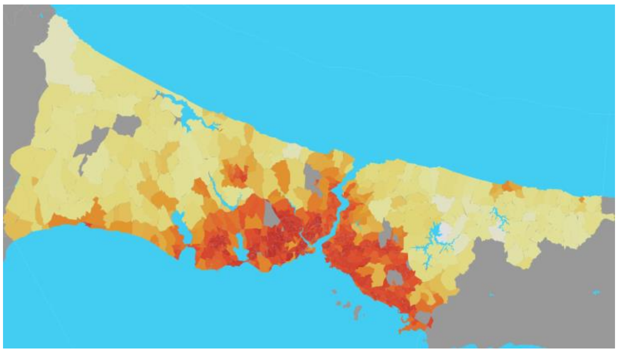


Figure 11. The population density in districts of Istanbul [60].

The exact population of districts, their neighbor district(s) and the total number of grid points located in the districts are presented in Table 2. As figured out from the table, there are 34 rows, because 5 of 39 districts are added to the nearest neighbor districts since they do not have any grid points on them. While calculating the population of a grid point, the population of the district that the grid is located is divided into the total number of grid points in that district.

After the calculation of the population of the grid points, the risk values of the grid points are determined. The 4 different degrees of earthquake zones in Istanbul illustrated in Figure 7 are evaluated by the experts in Kandilli Observatory and Earthquake Research Institute, Boğaziçi University in Turkey. Their assessments are presented in Table 3.

Table 3 also provides information about the number of grid points in the 4 different degrees of earthquake zones. More than half of the grid points are placed in the first 2 degrees of earthquake zones which can cause to increase the possible losses after a possible earthquake.

The overall weights (scores) of the grid points are calculated by multiplying their population and the risk values. Consequently, 15 different scenarios are created to solve the problem within different cases and parameters. The scenarios are summarized in Table 4.

According to the scenarios, the airports in Istanbul are selected as the origin nodes that the Bayraktar UAV takes and lands off. Since the Bayraktar has 3000 km. range, the range is divided into 5 different numbers to represent the number of daily visits to the selected grid points in candidate solutions. The number of daily visits is ranged from 1 to 5 and therefore the range of Bayraktar is limited with 3000 to 600 km., respectively. Visiting the grid points more than one time is considered to monitor the effects of possible aftershock earthquakes and the new good and bad incidents. After such an earthquake, new collapses, fires or explosions may happen or aftershock earthquakes can trigger more devastating events. In a good manner, some roads can be cleared or some fires may be put out. Therefore, it seems reasonable to observe the grid points for more than one time in the first 24 hours of the possible earthquake.

Table 2. The districts of Istanbul and their specifications [59].

#	District(s)	Population	Neighbor District(s)	Total # of Grid Points
1	Adalar	14907	19,21,23	4
2	Arnavutköy	261655	7,12,13,15,16	23
3	Ataşehir	423372	19,23,25,33,34	2
4	Avcılar	435682	7,11,15,22	2
5	Bağcılar+Güngören	748483+296967	6,7,8,17,22	1
6	Bahçelievler+Bakırköy	598454+222370	5,17,22	2
7	Başakşehir	396729	2,4,5,9,15,22,29	4
8	Bayrampaşa+Esenler	274197+454569	5,7,16,17,18,29	2
9	Beşiktaş+Beyoğlu	185447+236606	16,17,20,26,31	2
10	Beykoz	251087	14,30,33,34	13
11	Beylikdüzü	314670	4,12,15	1
12	Büyükçekmece	243474	2,11,15,27	6
13	Çatalca	69057	2,12,27	34
14	Çekmeköy	248859	10,24,25,30,33	7
15	Esenyurt	846492	2,4,7,11,12	2
16	Eyüp	381114	2,8,9,17,18,20,26,29,31	12
17	Fatih+Zeytinburnu	433873+287378	5,6,8,9,16	2
18	Gaziosmanpaşa	497959	8,16,29	1
19	Kadıköy	451453	3,23,34	1
20	Kağıthane	442694	9,16,31	1
21	Kartal	463433	3,23,24,25,28	3
22	Küçükçekmece	770393	4,5,6,7	2
23	Maltepe	497586	3,19,21,25	2
24	Pendik	698260	14,25,28,30,32	13
25	Sancaktepe	402391	3,14,21,23,24,28,33	3
26	Sarıyer	344876	9,16,31	7
27	Silivri	180524	11,13	41
28	Sultanbeyli	329985	21,24,25	1
29	Sultangazi	528514	7,8,16,18	1
30	Şile	35131	10,14,24	28
31	Şişli	274196	9,16,20	1
32	Tuzla	252923	24	4
33	Ümraniye	699901	3,10,14,25,34	1
34	Üsküdar	533570	3,10,19,33	1
	Total	15029231	-	230

Table 3. Earthquake zones in Istanbul and their risk values.

Degree of the Earthquake Zone	Risk Value	Total # of Grid Points
1st	0.4	42
2nd	0.3	96
3rd	0.2	63
4th	0.1	29
Total	1	230

Table 4. Scenarios.

#	Base Station	Number of daily visits	Range of UAV (km.)
1	Atatürk Airport	1	3000
2		2	1500
3		3	1000
4		4	750
5		5	600
6	Sabiha Gökçen Airport	1	3000
7		2	1500
8		3	1000
9		4	750
10		5	600
11	Istanbul Airport	1	3000
12		2	1500
13		3	1000
14		4	750
15		5	600

The parameters of the proposed GA and SA are tuned by using the first scenario. 30 runs are made for each combination of the parameters by running with the candidate values. The best-resulted values for the parameters are selected according to 2430 runs for GA and 270 runs for the SA. The assigned values are presented in Table 5. The μ and g_{max} parameters are considered as pairs. Since the GA is using $\mu * g_{max}$= 10000 number of solutions to reach better solutions in all generations, the i_{max} parameter of the SA is also set to 10000 to make a fair comparison.

Table 5. Tuned and assigned parameters.

Algorithm	Parameter(s)	Candidate values	Assigned value(s)
GA	μ, g_{max}	20 , 500 50 , 200 100 , 100	20, 500
	e	0.1 0.2 0.3	0.2
	P_c	-	1
	P_m	0.1 0.2 0.3	0.3
	k	3 5 7	7
SA	T_0	-	0.1035°
	α	0.9 0.95 0.99	0.99
	β	200 500 1000	200
	i_{max}	-	10000

The proposed GA and SA are coded in MATLAB, and the mathematical model is tried to be solved via ILOG CPLEX 12.6.1. The runs of metaheuristics and the mathematical model are made on the same computer that has Intel Core I5-7500U CPU, 3.40 GHz, and 8 GB RAM specifications. By using the tuned parameter values, the provided results by the proposed GA, SA and the model for each scenario are illustrated in Table 6.

According to the results, the ILOG CPLEX found the optimal solutions just for the 2 of 15 scenarios. Both optimal solutions can cover 230 of 230 grid points on Istanbul (i.e. f=1 for the 6th and the 11th scenarios) that means even by considering the weight values of the grid points the UAV can monitor the possible damages in overall the city in the first 24 hours after the possible earthquake within 3000 km. flight range. Without considering the weights of the grid points the UAV can fly in a straighter route and can get more information. The best route of the UAV for scenario 11 found by ILOG CPLEX is drawn in Figure 12. As seen, the overall grid points in Istanbul are visited just one time.

For the rest of the scenarios except 6 and 11, the GA has the highest fitness values than the SA as seen in Figure 6. The GA has better solutions in almost all scenarios for the best, mean and worst columns. The GA has relatively high fitness values (i.e. f > 0.9), therefore these are good results for the problem that the UAV is observing most of the highest scored (weighted) grid points (i.e. more risky and crowded regions). It can be interpreted that most of the effects of the possible earthquake in Istanbul can be monitored by a Bayraktar UAV in a day in all scenarios. The visited grid numbers can be increased by refueling the UAV at the airport after the first flight. In the other scenarios, visiting the same selected grid points 5 times in a day is decreasing the best f values of GA to 0.880, 0.881 and 0.899. And the visited grid points are ranging between 29 and 42 for the GA. In these scenarios, just some of the important grid points can be observed. All results are ranging between 189 and 29 grid points. The best route of the UAV for scenario 11 found by GA is illustrated in Figure 13.

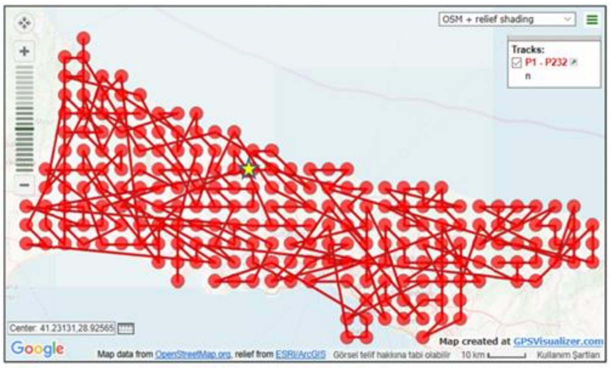

OSM + relief shading
Tracks:
P1 - P232
Center: 41.23131,28.92565
Google
Map created at GPSVisualizer.com
Map data from OpenStreetMap.org, relief from ESRI/ArcGIS Görsel telif hakkına tabi olabilir 10 km Kullanım Şartları

Figure 12. The best route result of ILOG CPLEX for scenario 11 with covering 230 of 230 grid points (i.e. f=1) starting from and ending at Istanbul Airport (i.e. star).

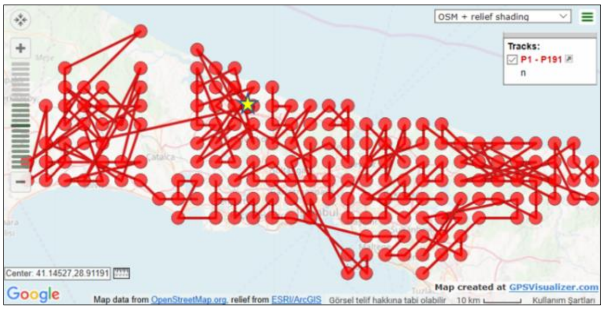

OSM + relief shading
Tracks:
P1 - P191
Center: 41.14527,28.91191
Google
Map created at GPSVisualizer.com
Map data from OpenStreetMap.org, relief from ESRI/ArcGIS Görsel telif hakkına tabi olabilir 10 km Kullanım Şartları

Figure 13. The best route result of GA for scenario 11 with covering 189 of 230 grid points (i.e. f=0.999) starting from and ending at Istanbul Airport (i.e. star).

The t-tests are conducted for each scenario and according to the test results GA has better results than SA in all scenarios (p-values are ranging from 0.037889 to 2.82x10^{-26}). The boxplots of the results that illustrate the superiority of the proposed GA are also presented in Figure 14. The proposed GA can solve the scenarios between 4.51 to 10.97 seconds, and the SA is 0.81 to 3.58 seconds as seen in Table 7.

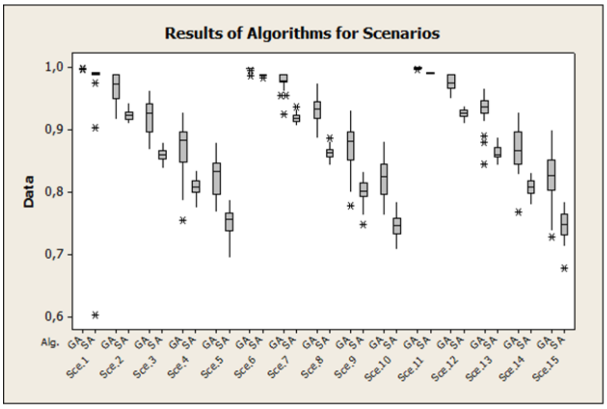


Figure 14. The boxplots of the results

Table 6. The results for the scenarios.

#	Exact solution	Visited # of grid points for the exact solution	GA				SA			
			Best fitness	Mean fitness	Worst fitness	Visited # of grid points for the best fitness	Best fitness	Mean fitness	Worst fitness	Visited # of grid points for the best fitness
1	N/A	N/A	**0.999**	**0.998**	**0.997**	170	0.993	0.974	0.603	154
2	N/A	N/A	**0.989**	**0.968**	**0.917**	106	0.943	0.923	0.909	85
3	N/A	N/A	**0.964**	**0.922**	**0.868**	66	0.880	0.860	0.838	62
4	N/A	N/A	**0.929**	**0.870**	0.754	60	0.835	0.807	**0.774**	47
5	N/A	N/A	**0.880**	**0.823**	**0.768**	41	0.788	0.751	0.695	38
6	**1**	**230**	0.999	0.998	0.987	185	0.990	0.987	0.982	139
7	N/A	N/A	**0.990**	**0.978**	**0.924**	107	0.936	0.919	0.907	78
8	N/A	N/A	**0.975**	**0.931**	**0.886**	49	0.887	0.862	0.842	61
9	N/A	N/A	**0.932**	**0.873**	**0.778**	45	0.833	0.802	0.747	45
10	N/A	N/A	**0.881**	**0.821**	**0.763**	29	0.785	0.745	0.707	37
11	**1**	**230**	0.999	0.999	0.997	189	0.993	0.991	0.989	156
12	N/A	N/A	**0.990**	**0.974**	**0.949**	97	0.938	0.926	0.910	81
13	N/A	N/A	**0.966**	**0.932**	**0.844**	47	0.889	0.863	0.843	63
14	N/A	N/A	**0.928**	**0.867**	0.767	41	0.831	0.807	**0.780**	47
15	N/A	N/A	**0.899**	**0.822**	**0.727**	42	0.784	0.748	0.677	40

Table 7. The CPU times (s.).

#	Exact	GA			SA		
		Best	Mean	Worst	Best	Mean	Worst
1	N/A	9.62	10.33	10.97	1.70	3.20	3.58
2	N/A	6.45	7.13	7.61	1.75	1.84	2.00
3	N/A	5.55	6.13	6.67	1.24	1.32	1.47
4	N/A	5.10	5.48	5.10	0.99	1.06	1.16
5	N/A	4.71	4.97	5.43	0.83	0.90	0.98
6	432	9.74	10.26	10.77	2.91	3.13	3.25
7	N/A	6.80	7.32	7.81	1.69	1.81	1.98
8	N/A	5.45	5.90	6.31	1.19	1.32	1.46
9	N/A	4.72	5.27	5.59	0.97	1.05	1.18
10	N/A	4.59	4.81	5.06	0.82	0.88	0.93
11	817	9.89	10.11	10.44	3.18	3.37	3.52
12	N/A	6.91	7.28	7.67	1.75	1.89	2.04
13	N/A	5.60	6.06	6.79	1.24	1.38	1.53
14	N/A	5.00	5.30	5.74	0.97	1.11	1.18
15	N/A	4.51	4.87	5.15	0.81	0.91	1.03

The convergence of the proposed GA from initial solutions to better individuals can be seen in Figure 15. In Figure 15, the best and the mean f values of the generations are given for a sample run. The proposed GA is increasing the initial best f value from 0.5457 to 0.9375 levels at the end of the run (i.e. in 500 generations). The mean f values of the generations are also close to the best f values of the generations that mean the individuals in the generations are becoming similar to each other as a clue for the convergence.

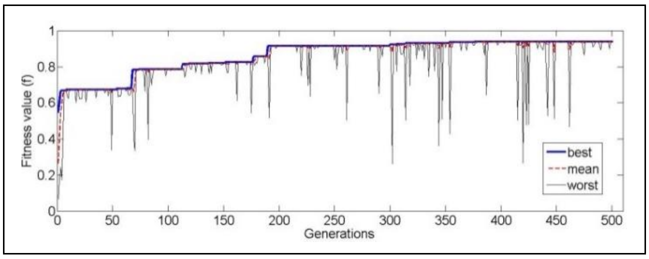


Figure 15. The convergence of GA for a sample run.

6 Conclusion

In this paper, the problem is to find a route for a UAV to observe the possible damages that can be occurred by an expected Istanbul earthquake. The UAV takes off from Istanbul within the first 24 hours. In the problem, 230 candidate grid points that can be visited by UAV at Istanbul are determined. The weight values (scores) for all candidate grid points are determined by combining the risk values based on earthquake degree zones and the population densities of the points. It is aimed to find a route for the UAV to maximize the total weights of the visited grid points under a range constraint. The described problem is adapted to the OP in the literature. Since the OP is an NP-hard problem, a problem-specific GA and SA are developed to solve the problem. The parameters of the GA and SA are tuned by experiments. 15 different scenarios including the daily number of visits and the airports that the UAV takes and lands off after the earthquake are created. The scenarios are solved with proposed GA, SA, and a mathematical model. The IP model found optimal solutions for 2 of 15 scenarios. The proposed GA has better solutions for the remaining scenarios, and it has reasonable CPU times (4.51 to 10.97 CPU seconds). Bayraktar UAV can visit up to 230 of 230 grid points in a day after a possible earthquake depending on the scenario. The solutions seem acceptable for such a disaster case. It is believed that the defined problem is very vital and there should be more studies about the post-earthquake scenarios for the inevitable Istanbul earthquake. For future research, another metaheuristic, matheuristic, hyperheuristic, or simheuristic methods and their hybrid versions can be tried for the defined problem. The results can be compared with the proposed GA results. Besides, the problem can be turned into a dynamic problem or there can be used more than one UAV. Some of the UAVs can keep going to monitor when they capture a happening incident (a fire for instance) by flying over the incident continuously. Other UAVs can continue to cover the remaining grid points dynamically. Small types of UAVs (i.e. short ranged drones) can also be tried for each district separately, and the total costs can be compared with the proposed GA solution. Other constraints can be added or multi-objective models are also can be created.

7 Acknowledgment

We would like to thank Kandilli Observatory and Earthquake Research Institute staff for their valuable support in the process of determining risk values of the earthquake zones. We are grateful to the anonymous reviewers and the editorial board for their helpful and constructive advices.

8 Author contribution statements

In the scope of this study, Muhammed Halat contributed to the formation of the idea, literature review, algorithm design, obtaining the results. Ömer Özkan contributed to the formation of the idea, literature review, algorithm design, obtaining and evaluating the results and writing and reviewing the manuscript.

9 Ethics committee approval and conflict of interest statement

There is no need to obtain permission from the ethics committee for the article prepared.

There is no conflict of interest with any person / institution in the article prepared.

10 References

[1] Boğaziçi Üniversitesi, Kandilli Rasathanesi ve Deprem Araştırmaları Enstitüsü. "İstanbul Metropolitan Alanının Deprem Risk Analizi". http://www.koeri.boun.edu.tr/arastirma_3.depmuh (02.07.2019).

[2] Erdik M, Durukal E. "Earthquake risk and its mitigation in Istanbul". *Natural Hazards,* 44, 181-197, 2008.

[3] Yu X, Zhang Y. "Sense and avoid technologies with applications to unmanned aircraft systems: review and prospects". *Progress in Aerospace Sciences,* 74, 152-166, 2015.

[4] Griffin GF. "The use of unmanned aerial vehicles for disaster management". *Geomatica,* 68(4), 265-281, 2014.

[5] Liu P, Chen AY, Huang YN, Han JY, Lai JS, Kang SC, Wu TH, Wen MC, Tsai MH. "A review of rotorcraft unmanned aerial vehicle (UAV) developments and applications in civil engineering". *Smart Structures and Systems,* 13(6), 1065-1094, 2014.

[6] Mersheva V. UAV Routing Problem for Area Monitoring in a Disaster Situation. PhD Thesis, Alpen-Adria Universitat, Klagenfurt, Austria, 2015.

[7] Nedjati A, Vizvari B, Izbirak G. "Post-earthquake response by small UAV helicopters". *Natural Hazards,* 80, 1669-1688, 2016.

[8] Zhu M, Du X, Zhang X, Luo H, Wang G. "Multi-UAV rapid-assessment task-assessment problem in a post-earthquake scenario". *IEEE Access,* 7, 74542-74557, 2019.

[9] Coutinho WP, Battarra M, Fliege J. "The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review". *Computers & Industrial Engineering,* 120, 116-128, 2018.

[10] Seyis A, Karacin Y, Ozkan O. "*Optimal path planning with minimum number of UAVs by using genetic algorithm*". *28th European Conference on Operational Research (EURO 2016)*, Poznan, Poland, 3-6 July 2016.

[11] Ozkan O. "*İnsansız hava araçları ile Türkiye'deki orman yangınlarının tespiti probleminin tavlama benzetimi ile eniyilenmesi*". *38. Ulusal Yöneylem Araştırması ve Endüstri Mühendisliği Kongresi (YAEM 2018)*, Eskişehir, Türkiye, 26-29 Haziran 2018.

[12] Kaya M, Ozkan O. "Sınır koruma görevi için insansız hava araçlarının rotalanması probleminin genetik algoritma ile eniyilenmesi". *38. Ulusal Yöneylem Araştırması ve Endüstri Mühendisliği Kongresi (YAEM 2018)*, Eskişehir, Türkiye, 26-29 Haziran 2018.

[13] Ercan C, Gencer C. "Dinamik insansız hava sistemleri rota planlaması literatür araştırması ve insansız hava sistemleri çalışma alanları". *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi,* 19(2), 104-111, 2013.

[14] Sahin Y, Karagul K. "Gezgin satıcı probleminin melez akışkan genetik algoritma (MAGA) kullanarak çözümü". *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi,* 25(1), 106-114, 2019.

[15] Tsiligirides, T. "Heuristic methods applied to orienteering". *Journal of the Operational Research Society,* 35(9), 797-809. 1984.

[16] Golden BL, Levy L, Vohra R. "The orienteering problem". *Naval Research Logistics,* 34(3), 307-318, 1987.

[17] Ramesh R, Yoon YS, Karwan MH. "An optimal algorithm for the orienteering tour problem". *ORSA Journal of Computing,* 4(2), 155-165, 1992.

[18] Chao IM, Golden BL, Wasil EA. "A fast and effective heuristic for the orienteering problem". *European Journal of Operational Research,* 88(3), 475-489, 1996.

[19] Feillet D, Dejax P, Gendreau M. "Traveling salesman problems with profits: An overview". *ORP^3 Conference,* Paris, France, 26-29 September, 2001.

[20] Fomin FV, Lingas A. "Approximation algorithms for time-dependent orienteering". *Information Processing Letters,* 83, 57-62, 2002.

[21] Boussier S, Feillet D, Gendreau M. "An exact algorithm for the team orienteering problem". 4OR, 5(3), 211-230, 2007.

[22] Righini G, Salani M. "Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming". *Computers & Operations Research,* 36(4), 1191-1203, 2009.

[23] Bouly H, Dang DC, Moukrim A. "A memetic algorithm for the team orienteering problem". 4OR, 8(1), 49-70, 2010.

[24] Tricoire F, Romauch M, Doerner KF, Hartl RF. "Heuristics for the multi-period orienteering problem with multiple time windows". *Computers & Operations Research,* 37(2), 351-367, 2010.

[25] Campbell AM, Gendreau M, Thomas BW. "The orienteering problem with stochastic travel and service times". *Annals of Operations Research,* 186(1), 61-81, 2011.

[26] Labadie N, Mansini R, Melechovsky J, Calvo WR. "Hybridized evolutionary local search algorithm for the team orienteering problem with time windows". *Journal of Heuristics,* 17(6), 729-753, 2011.

[27] Vansteenwegen P, Souffriau W, Van Oudheusden D. "The orienteering problem: A survey". *European Journal of Operational Research,* 209(1), 1-10, 2011.

[28] Labadie N, Mansini R, Melechovsky J, Calvo WR. "The team orienteering problem with time windows: An LP-based granular variable neighborhood search". *European Journal of Operational Research,* 220(1), 15-27, 2012.

[29] Lin SW, Yu VF. "A simulated annealing heuristic for the team orienteering problem with time windows". *European Journal of Operational Research,* 217(1), 94-107, 2012.

[30] Afsar HM, Labadie N. "Team orienteering problem with decreasing profits". *Electronic Notes in Discrete Mathematics,* 41, 285-293, 2013.

[31] Archetti C, Bianchessi N, Speranza MG. "The capacitated team orienteering problem with incomplete service". *Optimization Letters,* 7(7), 1405-1417, 2013.

[32] Dang DC, Guibadj RN, Moukrim A. *A Branch-And-Cut Algorithm for Solving the Team Orienteering Problem*. Editors: Gomes C., Sellmann M. Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2013) Lecture Notes in Computer Science, vol 7874, Berlin, Heidelberg, Springer, 2013.

[33] Divsalar A, Vansteenwegen P, Cattrysse D. "A variable neighborhood search method for the orienteering problem with hotel selection". *International Journal of Production Economics,* 145(1), 150-160, 2013.

[34] Lin SW. "Solving the team orienteering problem using effective multi-start simulated annealing". *Applied Soft Computing,* 13(2), 1064-1073, 2013.

[35] Özdemir M. Zaman Kısıtı Altında Takım Oryantiring Problemlerinin Yapay Arı Kolonosi Yaklaşımı ile Çözümü. Yüksek Lisans Tezi, İstanbul Üniversitesi, İstanbul, Türkiye, 2013.

[36] Tarantilis CD, Stavropoulou F, Repoussis PP. "The capacitated team orienteering problem: A bi-level filter-and-fan method". *European Journal of Operational Research,* 224(1), 65-78, 2013.

[37] Angelelli E, Archetti C, Vindigni M. "The clustered orienteering problem". *European Journal of Operational Research,* 238(2), 404-414, 2014.

[38] Campos V, Marti R, Sanchez-Oro J, Duarte A. "GRASP with path relinking for the orienteering problem". *Journal of the Operational Research Society,* 65(12), 1800-1813, 2014.

[39] Cura T. "An artificial bee colony algorithm approach for the team orienteering problem with time windows". *Computers & Industrial Engineering,* 74, 270-290, 2014.

[40] Evers L, Glorie K, van der Ster S, Barros AI, Monsuur H. "A two-stage approach to the orienteering problem with stochastic weights". *Computers & Operations Research,* 43, 248-260, 2014.

[41] Hu Q, Lim A. "An iterative three-component heuristic for the team orienteering problem with time windows". *European Journal of Operational Research,* 232(2), 276-286, 2014.

[42] Salazar-Aguilar MA, Langevin A, Laporte G. "The multi-district team orienteering problem". *Computers & Operations Research,* 41, 76-82, 2014.

[43] Verbeeck C, Sörensen K, Aghezzaf EH, Vansteenwegen P. "A fast solution method for the time-dependent orienteering problem". *European Journal of Operational Research,* 236(2), 419-432, 2014.

[44] Zhang S, Ohlmann JW, Thomas BW. "A priori orienteering with time windows and stochastic wait times at customers". *European Journal of Operational Research,* 239(1), 70-79, 2014.

[45] Duque D, Lozano L, Medaglia AL. "Solving the orienteering problem with time windows via the pulse framework". *Computers & Operations Research,* 54, 168-176, 2015.

[46] Gavalas D, Konstantopoulos C, Mastakas K, Pantziou G, Vathis N. "Approximation algorithms for the arc orienteering problem". *Information Processing Letters,* 115, 313-315, 2015.

[47] Gunawan A, Lau HC, Lu K. "Well-Tuned ILS for Extended Team Orienteering Problem with Time Windows". Living Analytics Research Center, Singapore, Technical Report, 2015.

[48] Lin SW, Yu VF. "A simulated annealing heuristic for the multiconstraint team orienteering problem with multiple time windows". *Applied Soft Computing,* 37, 632-642, 2015.

[49] Marinakis Y, Politis M, Marinaki M, Matsatsinis N. *A Memetic-GRASP Algorithm for the Solution of the Orienteering Problem*. Editors: Le Thi HA, Dinh TP, Nguyen N.T. Modelling, Computation and Optimization in Information Systems and Management Sciences, Advances in Intelligent Systems and Computing: 360, 105-116, Switzerland, Springer, 2015.

[50] Gunawan A, Lau HC, Vansteenwegen P. "Orienteering Problem: A survey of recent variants, solution approaches and applications". *European Journal of Operational Research*, 255, 315-332, 2016.

[51] Yu J, Schwager M, Rus D. "Correlated orienteering problem and its application to persistent monitoring tasks". *IEEE Transactions on Robotics,* 32(5), 1106-1118, 2016.

[52] Ostrowksi K, Karbowska-Chilinska J, Koszelew J, Zabielski P. "Evolution-inspired local improvement algorithm solving orienteering problem". *Annals of Operations Research,* 253, 519-543, 2017.

[53] Hu W, Fathi M, Pardolos PM. "A multi-objective evolutionary algorithm based on decomposition and constraint programming for the multi-objective team orienteering problem with time windows". *Applied Soft Computing Journal,* 73, 383-393, 2018.

[54] Tsakirakis E, Marinaki M, Marinakis Y, Matsatsinis N. "A similarity hybrid harmony search algorithm for the team orienteering problem". *Applied Soft Computing Journal*, 80, 776-796, 2019.

[55] Holland JH. *Adaptation in Natural and Artificial Systems,* Michigan, USA, University of Michigan Press, 1975.

[56] Aarts E, Korst J. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing,* New York, USA, Wiley, 1989.

[57] Türk Mühendis ve Mimar Odaları Birliği (TMMOB). "İstanbul Deprem Raporu" https://www.tmmob.org.tr/sites/default/files/rapor_2017_son.pdf (02/07/2019).

[58] T.C. Çevre ve Şehircilik Bakanlığı. "İstanbul Deprem Bölgeleri Dağılımı Haritası". https://istanbulakdm.csb.gov.tr/istanbul-deprem-bolgeleri-dagilimi-haritasi-i-3712 (02/07/2019).

[59] Türkiye İstatistik Kurumu "Merkezi Dağıtım Sistemi". https://biruni.tuik.gov.tr/medas/?kn=95&locale=tr (02/07/2019).

[60] Atlasbig. "İstanbul Mahalleleri Nüfus Yoğunluğu". https://www.atlasbig.com/tr/istanbul-mahalleleri-nufus-yogunlugu (02/07/2019).