Route Optimization Method for Unmanned Air Vehicle Launched from a Carrier

Halil Savuran and Murat Karakaya

Abstract—**In this paper, we propose a route optimization method for a carrier-launched Unmanned Air Vehicle (UAV). In a real life use case, the carrier keeps on moving on its own route as the UAV executes its own mission of visiting the targets dispersed on a geographical area. Due to carrier mobility, determining the UAV take-off and land-on locations with a route which minimizes the total tour length is a crucial research question and a practical challenge. In order to resolve this problem, we have designed a solution based on the Genetic Algorithm (GA). We have observed the performance of the proposed approach on some well-known TSP problems by comparing its results against the results of the Nearest Neighbor (NN) heuristic.**

Index Terms—**Carrier-launched unmanned air vehicle, genetic algorithm, mobile depot, nearest neighbor heuristic, route plan optimization, traveling salesman problem, vehicle routing problem.**

I. Introduction

Since their first appearance in 1920s, the usage of UAVs for military and civilian purposes has been keeping on expanding [1]. Considering the ongoing acquisition and development projects, it is assumed that the utilization of carrier platforms to augment the effectiveness of these vehicles will gain wider use in the near future [2]-[4]. Due to their higher endurance (see Table I), UAVs can be tasked on geographically dispersed multiple ground targets. In terms of mission effectiveness and efficiency, minimizing the total route length covering all the given targets is significant.

TABLE I : Endurance and Range Values for Some of the UAVs [1]

UAV	Endurance(hr)	Range(km)
Global Hawk	30	22000
GNAT	40	4818
A 160	30	4625
Predator	20	740
Predator B	24	1500
Heron	40	3300

As there are many different constraints in real life scenarios, there have been various works to resolve the routing problems assuming different constraints [5]-[8]. One of the scenarios is to deploy UAVs on a mobile platform, e.g. aircraft carriers, to exploit the platform mobility. In such cases, as the carrier will be on the move while the UAV executes its mission, determination of take-off and land-on locations as well as the ordering of targets to be visited by the UAV with the constraint of minimizing the total tour length requires an effective route design. In this paper, we consider this use case and propose a method based on the GA.

In the following sections of this paper we first introduce the related work and the GA & NN methods. Then, we present the research problem and the proposed solutions in detail. Finally, the simulation tests and their results are discussed.

Manuscript received July 12, 2014; revised September 15, 2014. This work was supported in part by the Atilim University.

Halil Savuran is with the Graduate School of Natural and Applied Sciences of Atilim University, Ankara, Turkey (e-mail: savuran.halil@student.atilim.edu.tr).

Murat Karakaya is with the Department of Computer Engineering of Atilim University, Ankara, Turkey (e-mail: murat.karakaya@atilim.edu.tr).

II. Related Work

The problem mentioned above is related with the well-known Vehicle Routing Problem (VRP). The VRP can be defined as designing an optimal route from a given depot to *n* number of customers, in a manner that every customer is visited once and only once, and the route ends in the starting depot [9]. As such, a VRP problem is basically a TSP problem and a classic case of combinatorial optimization problems.

There are a wide variety of VRP problems focusing on different constraints [5] and many exact and heuristic approaches have been proposed to solve them [10], [11]. As the solution space of such a problem includes *n!* different combinations, a new customer to be introduced in the problem may increase the size of the solution space exponentially. Therefore, VRP problems are defined as non-deterministic polynomial problems (NP-Hard) [12]. This situation makes the problem attractive for the heuristic and meta-heuristic methods, such as Tabu Search [13], Ant Systems [14], Artificial Bee Colony [15], Particle Swarm Optimization [16], Simulated Annealing [17] and Genetic Algorithm [18]-[21].

We have observed that most of the study in the field consider static depot in the problem and even though there are some studies focusing on problems where multiple depots are involved, from the perspective of solution design and handling a dynamically position switching depot (UAV carrier in this case), the subject problem of this paper addresses rather an emerging area of research for the time being.

III. Genetic Algorithm

In the literature, Genetic Algorithm is principally deemed a heuristic search method that mimics the nature from the perspective of biological evolution. Several researchers studied evolution inspired algorithms in 1950s and 1960s, to solve specific problems [22]. However, GA is accepted to be invented by John Holland in 1960s. The difference of his study from that of others was, that he formally studied the phenomenon of adaptation as it occurs in nature and tried to develop ways to import the mechanism of adaptation into computer systems [23]. He also extensively studied the foundational theory of GA as why and how it works [24]. His 1975 book *"Adaptation in Natural and Artificial Systems"* presents a theoretical framework for adaptation under the GA [22], [23].

A. Implementation

GA depends on the basic rule of Darwinian Theory of Evolution: "Survival of the Fittest" [25], [26]. In practice, GA treats a set of possible solutions in the search space as if they are a population of individuals, applies crossover operation on pairs to produce offsprings, applies mutation operation on individuals to preserve genetic diversity, calculates the fitness of these individuals by the given objective function and uses their fitness values to determine which ones to let survive. An optional GA operation is *elitism*, which suggests cloning the most fit individual in every generation by-passing the genetic operations into the next generation, to make sure potential adverse effects of genetic operations do not cause an evolutionary fall back. One cycle of this set of operations is called a generation, such cycles are repeated until a stop criteria is satisfied [22], [23].

Fig. 1 represents the terminological connection between genealogy and GA.

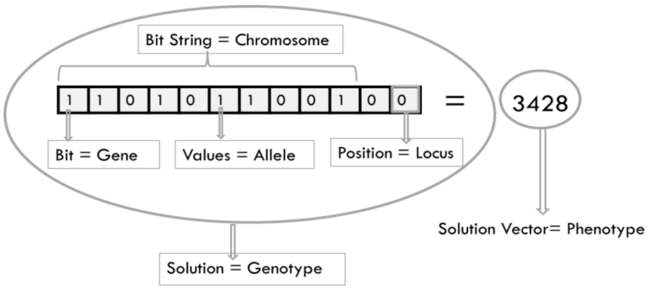


Fig. 1. Representation of terminological connection between genealogy and GA.

B. Applying GA on VRP Problems

Some problems may need an extra process to be made suitable for GA operations. Permutation based problems such as VRP fall under this type.

1) Representation of VRP in GA

Potential solutions of a VRP problem are represented as chromosomes in which each of the customer location is represented as a gene. The order the genes are placed corresponds to the actual order each customer will be visited.

As a VRP problem is classified as a combinatorial problem, the constraint of non-repetitiveness of genes in a chromosome is introduced. This dictates the GA operators such as crossover and mutation to respect this fact. There are studies performed on the problems of such nature.

2) GA operations suitable for VRP

As simple GA operations will most probably result in mal-formed chromosomes that have repetitive and missing genes, a common approach is to attache a repairing operation to such operations when such results occur. Larranaga makes a comprehensive review of representations and operations of GA for TSP [20]. Another study with similar purpose is Whitleys' [27]. A comprehensive list of permutation respecting crossover operations is made by Üçoluk [28] where he also proposes his own method that avoids the need for using a special crossover or mutation operator. Another crossover operation proposed by Ahmed promotes the advantage of consciously selecting the better genes from parents during the crossover [19]. Larranaga explains some of the popular crossover and mutation operators and their procedures in detail [20].

3) Objective function of a VRP chromosome

Fitness value of a VRP chromosome is the total distance to be covered while visiting all the customers in the order of genes in that chromosome, including the distance between the last customer and the starting node (depot). Since the objective of a route optimization problem is minimization of the tour length, smaller tour lengths mean greater fitness values.

There are two approaches adopted for tour length calculation, in one approach a cost matrix is used where all the distances between each city is predefined. This is usually the case for TSP problems where the distance of two nodes is dependent on how the path is laid between them. In the other approach, an euclidean distance calculation is performed on geographical locations of customers sequantially. Latter one is adopted in our study, as it fits well for an air vehicle where the shortest path between two nodes is also the shortest distance between them. Thus, a basic objective function for a VRP chromosome where the tour starts and ends in the same depot is the total Tour Length (TL) of the solution string it contains, which is acquired by a sequence of euclidean distance calculations as given below (1), where x and y parameters are the x and y coordinates of a node, l is the length of the chromosome and the index 0 holds the location of the depot.

$$TL = \sum_{i=0}^{l-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} + \sqrt{(x_{l-1} - x_0)^2 + (y_{l-1} - y_0)^2} \tag{1}$$

4) 2-Opt method

2-Opt is a local search method that was developed by Croes [12] to rearrange a route in a way that intersecting sub-tours to not to intersect (see Fig. 2) and is widely used on TSP based problems [29].

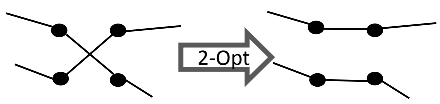


Fig. 2. Representation of a 2-Opt operation on an intersecting sub-tour.

On a solution string (tour) that was provided as input by heuristic methods, 2-Opt iteratively reverses one of the two adjacent sub tours and if the new fitness is better than the original one the tour is rearranged with this order.

IV. Implementation of NN for Route Optimization for a Carrier Launched UAV

Nearest Neighbor (NN) heuristic is one of the earliest methods proposed for TSP problems and adopts the principle of always jumping to the next nearest node until all nodes have been visited [30]. It runs fast but producing the optimal tour highly depends on the layout of the nodes within the problem.

We have adapted NN for the subject use case to evaluate the performance of our proposed GA solution.

In the implementation of the NN heuristic to the given use case, after covering the last node, algorithm calculates the landing point and includes it in the tour.

V. Implementation of GA for Route Optimization for a Carrier Launched UAV

A. *Graphical Representation of a Solution*

The potential take-off points throughout the carriers' route are defined as *steps*. These steps should be in such a frequency that the distance between two successive steps are ignorable in terms of affecting the design of the route to be traveled by the UAV. Fig. 3 represents a sample route design in this manner.

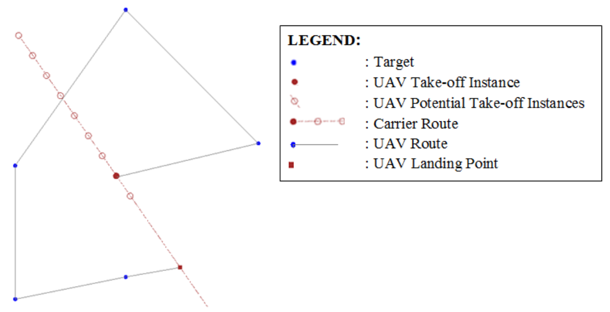


Fig. 3. A sample route design.

B. *Genetic Operations*

One cycle of genetic operations is called a generation. Stopping criteria to finalize evolution could be a fixed number of elapsed generations, reaching a desired fitness value or number of generations where no improvement takes place.

A high level flow of our algorithm is given in Fig. 4.

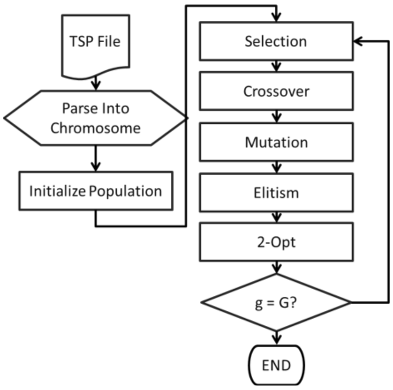


Fig. 4. A high level flow chart of the algorithm.

Initialization: After parsing a TSP file into a chromosome, the gene order of this chromosome is randomized to generate a new chromosome. Geographically closest take-off instance to the first gene of this chromosome is assigned as the take-off instance of this chromosome. Chromosomes created this way are added to the population. New chromosomes are kept on generated this way until the given population number (N) is reached.

Selection: Selection operation eliminates the worst half of the population and clones the fittest individual for elitism.

Crossover: Crossover process uses the Partially Mapped Crossover (PMX) operator [20] for reproduction of the population. This operator, after performing a simple two-point crossover between two parents, swaps the repetitive genes between siblings according to the mapping between exchanged parts (Fig. 5).

Fig. 5. PMX crossover.

Mutation: One of the Exchange, Displacement, Insertion and Inversion mutations is used randomly, within the given mutation probability.

Elitism: Elitism is applied as explained in para. III.A.

2-Opt: At the end of the 2-Opt operation, each re-ordered chromosome is re-assigned an updated take-off instance, as explained in para. V.C.

C. *Calculation of the Take-off Instance*

Since the take-off instance of a tour is a parameter in the objective function (See para. V.E), it needs to be assigned before the selection operation. Therefore, take-off instances are calculated once in initialization and at the end of every generation. For this, NN heuristic is employed in a minimal dose, in the sense that geographically closest take-off instance to the first gene of a chromosome is assigned as its take-off instance.

D. *Calculation of the Landing Point*

As the carrier is on the move while the UAV is executing its mission of visiting the targets, the route design should dynamically include the meeting point of the UAV and the carrier, which is actually the landing point of the UAV. On the other hand, the location of the landing point keeps changing depending on the length of the tour the UAV has to cover. To handle this issue, a trigonometric representation is utilized as described below.

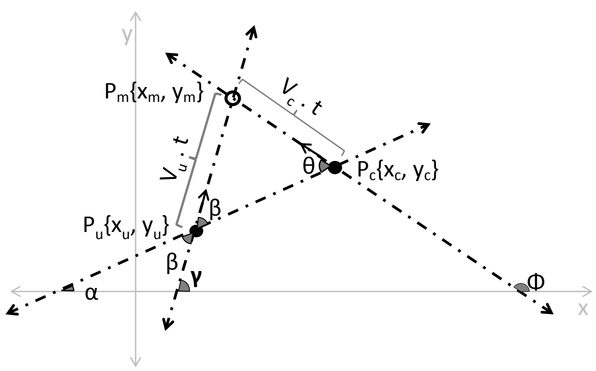


Fig. 6. Trigonometric representation of the carrier and the UAV meeting point.

Known: UAV Speed (V_u), Carrier Speed (V_c), Carrier Heading (θ) and the current traveled distances of the UAV and the carrier.

Assumption: Both UAV and carrier are assumed to have constant speed.

Unknown: Meeting point of the UAV and the carrier.

As t_0 being the moment the UAV completed the visit of the last target and ready to return back to the carrier, P_c , P_u and P_m respectively represent the carrier, UAV and meeting points.

As the meeting point of the UAV and the carrier will be the intersection of their axes of movement, P_m can be expressed as the solution set of the lines d_u and d_c as seen in Fig. 6.

Then from the linear equation of point-slope form:

$$(y_m - y_0 = m(x_m - x_0)):$$

$$d_c\!: y_m = \tan(\Phi) \,.\, (x_m - x_c) + y_c$$

$$d_u\!: y_m = \tan(\gamma) \,.\, (x_m - x_u) + y_u$$

From triangle properties:

$$\gamma = \alpha + \beta$$

From the linear equation of point-slope form:

$$(m = \tan(\alpha) = (x - x_0) / (y - y_0)):$$

$$\alpha = \arctan\left((y_c - y_u) / (x_c - x_u) \right)$$

From the law of sines (a / sin A = b / sin B = c / sin C):

$$\beta = \arcsin\left(V_c t / V_u t \right) . \sin(\theta)$$

Finally $P_m\{x_m, y_m\}$, from linear equation system:

$$\begin{bmatrix} \tan(\Phi) & -1 \\ \tan(\gamma) & -1 \end{bmatrix} \begin{bmatrix} x_m \\ y_m \end{bmatrix} = \begin{bmatrix} \tan(\Phi).\ x_c - y_c \\ \tan(\gamma).x_u - y_u \end{bmatrix}$$

And with the inclusion of the distance already covered by the carrier at time t_0 in the formulation, landing point of the UAV is predicted.

E. Objective Function

Unlike a generic VRP in which a route ends in where it starts off, objective function for a carrier launched UAV must calculate take-off and land-on points separately. Therefore, as x and y being the x and y coordinates of a node, l being the length of a chromosome, i = 0 being the take-off instance of the UAV, i = 1 .. l-1 being the nodes to be visitied and i=l being the land-on location of the UAV, tour length of a chromosome is calculated as in (2).

$$TL = \sum_{i=0}^{l-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \qquad (2)$$

VI. Simulation Results

We have implemented both GA and NN solutions with C# programming language and executed various tests using some of the well-known benchmark data files in the TSP Library [31]. These problems are listed in Table II.

TABLE II: Problems Tested in Simulation

Problem	Nodes#	Known Optimal Tour Length
Berlin52	52	7542
Eil76	76	538
Bier127	127	118282
Ch130	130	6110
Ch150	150	6528
KroB200	200	29437

Note that known optimal tour lengths for these problems are set for static depots and solutions form a closed circuit. Test results for the same problems involving mobile depots are explained below.

As stated before, the UAV and the Carrier are assumed to have constant speeds. Problem parameters used in the simulation is given in Table III.

TABLE III: Problem Parameters

UAV Speed (Units)	Carrier Speed (Units)	Number of Take-Off Instances
300	40	24

GA parameters used in simulation are given in Table IV.

TABLE IV: GA Parameters

Problem	Population Size (N)	Generations (G)	Crossover Probability (P_c)	Mutation Probability (P_m)
Berlin52	120	50	1.0	0.5
Eil76	120	80	1.0	0.5
Bier127	120	130	1.0	0.5
Ch130	120	130	1.0	0.5
Ch150	120	150	1.0	0.5
KroB200	120	200	1.0	0.5

Since GA is a stochastic method, we have run 30 tests with same parameters for each problem. As for NN, because of the deterministic nature of this method, we have run the algorithm for each of the take-off instances (24 in this case) once and selected the best result.

We have observed the success of our GA operations over the convergence of the of tour lengths (fitness-f) over the generation (g). Convergence graphs of Berlin52 and Bier127 are illustrated respectively in Fig. 7 and Fig. 8 for sampling.

Note: Tour length values of solution strings are negated in representation to reflect fitness values of chromosomes.

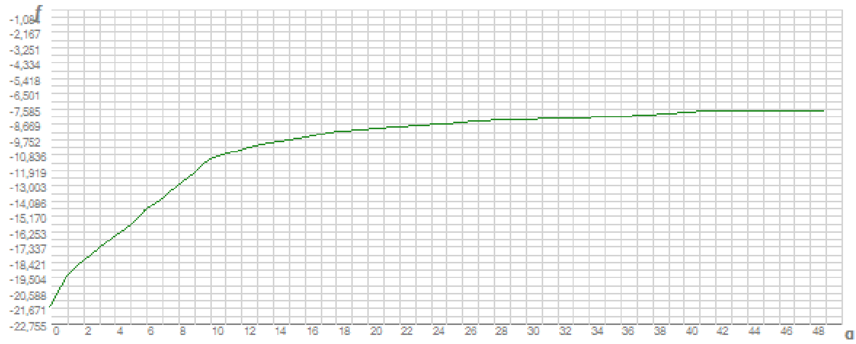


Fig. 7. Convergence graph for Berlin52.

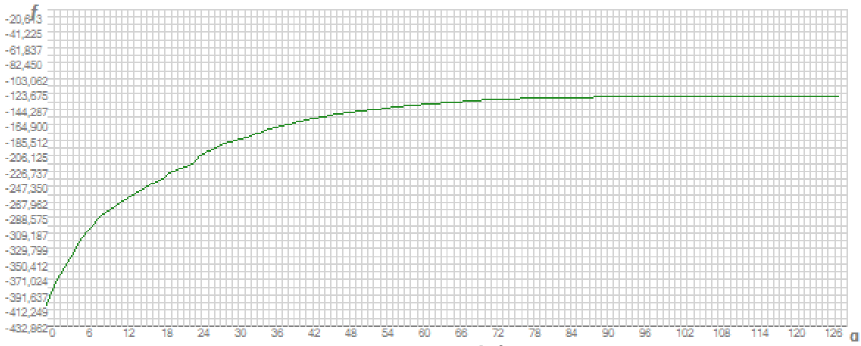
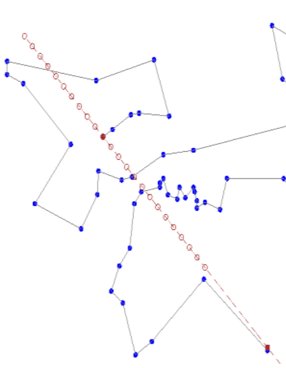


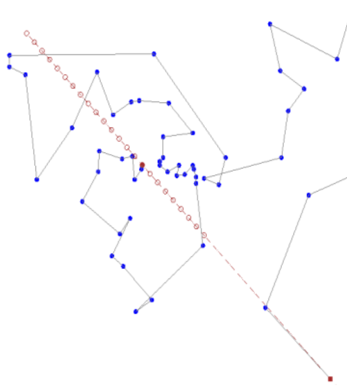
Fig. 8. Convergence graph for Bier127.

Sample route designs produced by both of these methods for sample problems of Berlin52 and Bier127 are given in Fig. 9 to Fig. 12. In these figures, we observe that the routes planned by GA are shorter and more effective compare to those of NN.



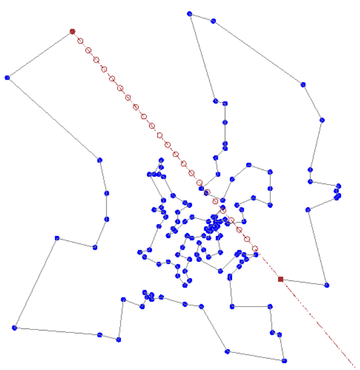
T.L.: 7533.01

Fig. 9. Route design generated by GA for the Berlin52 benchmark file.



T.L.: 8774.01

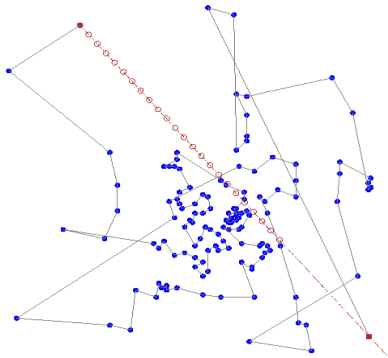
Fig. 10. Route design generated by NN for the Berlin52 benchmark file.



T.L.: 119234.3

Fig. 11. Route design generated by GA for the Bier127 benchmark file.

Best, average and worst length values of the routes generated by the GA are compared to the best results of the NN method for six different benchmark problems in Table V. We have run each settings of the simulation 30 times and presented the mean values in Table V.



T.L.: 151489.71

Fig. 12. Route design generated by NN for the Bier127 benchmark file.

TABLE V: COMPARISON OF TOUR LENGTHS GENERATED BY GA AND NN

Problem	GA(Average)	NN
Berlin52	7815.635975	8774.01
Eil76	595.8412477	655.09
Bier127	125322.0481	151489.71
Ch130	6672.368576	7344.68
Ch150	7288.101526	7645.36
KroB200	32918.41938	39984.14

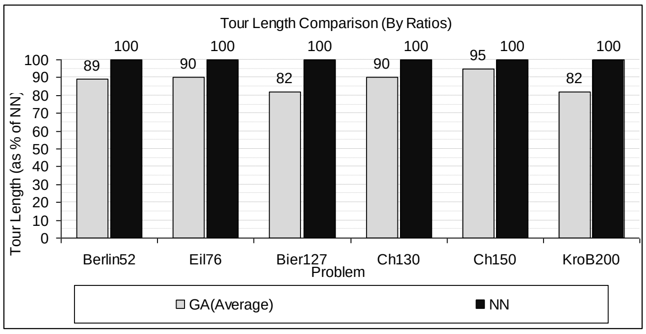


Fig. 13. Comparison of tour lengths for solutions generated by GA and NN.

Graphical visualization of these results is given in Fig. 13. Because the problems have different scales of distances, comparison metric is selected as the percentage of the GA results over the NN results.

As seen in Fig. 13, for the entire benchmark files we tested the GA method manages to create shorter routes compared to the results of the NN method. As the benchmark problem sizes and topologies are different, one can argue that the GA method is robust against these important parameter changes. The GA method can save up to 18% of the route length that the NN method generates. This saving is crucial for the operational usage and maintenance of UAVs and operators.

VII. Conclusion

In this paper we have developed a GA based solution for route plan optimization for a UAV launched from a moving carrier. One of the main aspects of our study is dynamically including the take-off and land-on points of the UAV in the objective function and reflecting their effects in fitness calculation this way.

We then compared the results of our proposed method against the results of the NN heuristic. The test results showed that the GA approach outperformed the NN method by generating shorter tours with the average of %88.5.

In line with the practical trend, our intention for the future is to extend this study to handle similar problems where multiple UAVs are available for mission and focus on maximum target coverage where endurance constraint is involved.

Acknowledgment

Authors would like to thank colleague Omer Ciftcibasi for his valuable contribution on trigonometric formulations and coding for calculation of the landing point.

Reference

[1] M. Agostino, S. Mammone, M. Nelson, M. Zhou, and T Arjomandi, "Classification of unmanned aerial vehicles," The University of Adelaide, Australia, 2006.

[2] M. Antoine and R. A. Dewolfe, "Promising outlook for navy's unmanned aviation," *National Defense*, pp. 36-37, July 2012.

[3] P. Cragin, "The way ahead for maritime UAVs," Newport RI Joint Military Operations Dept., Newport, 0704-0188, 2006.

[4] M. J. Sullivan *et al*., "Defense acquisitions: Navy strategy for unmanned carrier-based aircraft system defers key oversight mechanisms," Government Accountability Office, Washington DC, Congressional Report ADA586381, 2013.

[5] C. Ercan and C. Gencer, "Literature review of dynamic unmanned aerial system routing problems and proposals for future studies of UASs," *Pamukkale University Journal of Engineering Sciences*, pp. 104-111, 2013.

[6] M. Karakaya, "UAV route planning for maximum target coverage," *Computer Science & Engineering: An International Journal (CSEIJ)*, vol. 4, no. 1, February 2014.

[7] H. Ergezer and K. Leblebicioglu, "3D path planning for multiple UAVs for maximum information collection," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, pp. 737-762, January 2014.

[8] H. Ergezer and K. Leblebicioglu, "Path planning for UAVs for maximum information collection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 1, pp. 502-520, Jan. 2013.

[9] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80-91, 1959.

[10] T. Weise, A. Podlich, and C. Gorldt, "Solving real-world vehicle routing problems with evolutionary algorithms," *Natural Intelligence for Scheduling, Planning and Packing Problems*, vol. 250, pp. 29-53, 2009.

[11] G. Laporte, "The traveling salesman problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, pp. 231-247, 1992.

[12] G. A. Croes, "A method for solving traveling — Salesman problems," *Operations Research*, pp. 791–812, 1958.

[13] A. Amberg, W. Domschke, and V. Stefan, "A tabu search algorithm using capacitated trees," *European Journal of Operational Research*, vol. 124, no. 2, pp. 360-376., 2000.

[14] B. Bullnheimer, R. F. Hartl, and C. Strauss, "An improved ant system algorithm for the vehicle routing problem," *Annals of Operations Research*, vol. 89, pp. 319-328, 1999.

[15] W. Y. Szeto, Y.-Z. Wu, and C. H. Sin, "An artificial bee colony algorithm for the capacitated vehicle routing problem," *European Journal of Operational Research*, vol. 215, no. 1, pp. 126-135, 2011.

[16] F. Pinar Goksal, I. Karaoglan, and F. Altiparmak, "A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery," *Computers & Industrial Engineering*, vol. 65, no. 1, pp. 39-53, May 2013.

[17] W. C. Chiang and R. A. Russell, "Simulated annealing metaheuristics for the vehicle routing problem with time windows," *Annals of Operations Research*, vol. 63, no. 1, pp. 3-27, 1996.

[18] B. K. Ambati, "Heuristic combinatorial optimization by simulated Darwinian evolution: A polynomial time algorithm for the traveling salesman problem," *Biological Cybernetics*, pp. 31-35, 1991.

[19] Z. H. Ahmed, "Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator," *International Journal of Biometrics & Bioinformatics*, vol. 3, no. 6, pp. 96-106, 2010.

[20] P. Larranaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the traveling salesman problem: A review of representations and operators," *Artificial Intelligence Review*, no. 13, pp. 129-170, 1999.

[21] F. B. Pereira, "GVR: A new genetic representation for the vehicle routing problem," in *Artificial Intelligence and Cognitive Science*, Michael O'Neill, Ed., vol. 2464, Limerick, Ireland: Springer, 2002, pp. 95-102.

[22] D. A. Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers*, Singapore: World Scientific, 2010.

[23] M. Mitchell, *Introduction to Genetic Algorithms*, London: The MIT Press, 1996.

[24] H. John Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, U Michigan Press, 1975.

[25] C. Darwin, *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*, London, United Kingdom: John Murray, 1859.

[26] H. Braun, "On solving travelling salesman problems by genetic algorithms," in *Parallel Problem Solving from Nature*, H. P. Schwefel and R. Männer, eds., Kalsruhe, Germany: Universitat Kalsruhe, 1991, vol. 496, pp. 129-133.

[27] D. Whitley and N. W. Yoo, "Modeling Simple genetic algorithms for permutation problems," *Foundations of Genetic Algorithms*, Estes Park, 2006, pp. 163-184.

[28] G. Ucoluk, "Genetic algorithm solution of the TSP avoiding special crossover and mutation," *Intelligent Automation & Soft Computing*, vol. 8, no. 3, pp. 265-272, Mar. 2002.

[29] M. Englert, H. Röglin, and B. Vöcking, "Worst case and probabilistic analysis of the 2-Opt algorithm for the TSP," in *Proc. the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Aachen, 2007, pp. 1295-1304.

[30] E. L. Lawler, J. K. Lenstra, R. A. Kan, and D. B. Shmoys, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, New York: Wiley, 1985.

[31] University of Heidelberg. (Jan. 1995). TSP Library. [Online]. Available: http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/XML-TSPLIB/instances/

Halil Savuran was born in Izmir, Turkey in 1980. He received his B.S. degree from the Department of Computer Engineering at Turkish Air Force Academy, Istanbul, Turkey in 2002. He worked as an IT systems manager, project officer and software developer/maintainer in various Air Force projects between 2002 and 2014. He has been doing his M.Sc. at the Department of Software Engineering, Atilim University, Ankara, Turkey, since 2012. Currently he works as a data handling systems analyst for NATO AWACS base in Geilenkirchen, Germany, on behalf of Turkish Air Force.

Murat Karakaya received his B.S.E.E. degree in 1991 from the Turkish Military Academy (KHO), Ankara, Turkey, and the M.S. and Ph. D. degrees in computer engineering from the Bilkent University, Ankara, Turkey in 2000 and 2008, respectively. From 1992 to 2000, he worked as an engineer at different units in the Turkish Land Forces (KKK), Ankara, Turkey. From 2000 to 2005, he worked as an instructor and software engineer at the Turkish Military Academy (KHO), Ankara, Turkey. Then, during 2005-2008 he worked as an IT project manager in the North Atlantic Treaty Organization (NATO) Brussels, Belgium. From 2008 to 2012, he worked as an instructor and software engineer at the Turkish Military School of Electronics, Communications and Information Systems (MEBS) and Turkish Military Academy (KHO), Ankara, Turkey. He joined the Faculty of Atilim University in 2012 and he is currently an asst. professor in the Department of Computer Engineering, Ankara, Turkey. His research interests are natural computing, sensor networks, peer-to-peer networks, route optimization, and communications protocol design.