A Fuzzy Cluster Based Genetic Algorithm Approach for the Aircraft Landing Scheduling Problem

Yakup ÇELİKBİLEK*

Abstract

Aircraft Landing Scheduling (ALS) problem is one of the most important part of both aviation and air traffic control. The main objective of the problem is determining the landing time of the aircrafts with minimizing the penalty cost under some constraints. Each aircraft has an optimum target landing time based on their specialties related with fuel, airspeed and cost. Deviations from landing time targets increase the penalty cost of both the aircraft and the problem. In this paper, a fuzzy cluster based genetic algorithm approach is given for the solutions of ALS problems. An ALS benchmark, which contains up to 500 aircrafts and five runways, was obtained from OR–library to execute and evaluate the algorithm. Computational results of the proposed algorithm are given in detail and compared with the best results in the literature. The algorithm results show that it is very competitive and have good results when applied to the regarding problem.

Keywords: Aircraft Landing, Scheduling, Timetabling, Fuzzy Cluster, Genetic Algorithms

Uçak İniş Probleminin Çizelgelenmesinde Bulanık Küme Temelli Bir Genetik Algoritma Yaklaşımı

Öz

Uçak İniş Planlaması (UİP) problemi hem havacılığın hem de hava trafik kontrolünün en önemli bölümlerinden birisidir. Problemin esas amacı, bazı kısıtlar altında ihlal maliyetlerinin minimize edilerek uçakların iniş zamanlarının belirlenmesidir. Problemde, uçakların her biri için yakıt, hava hızı ve maliyet ile ilgili iniş zamanlarına dayalı spefikasyonların olduğu optimum hedefler söz konusudur. İniş zamanı hedefinden sapmalar uçağın ve problemin ihlal maliyetlerinin artmasına neden olmaktadır. Bu çalışmada bulanık küme temelli bir genetik algoritma yaklaşımı UİP problemleri için verilmiştir. 500 uçağın ve 5 pistin bulunduğu bir UİP test problemi önerilen tekniğin kullanılması ve değerlendirilmesi için yöneylem araştırması kütüphanesinden elde edilmiştir. Önerilen algoritma ile elde edilen detaylı sonuçlar

Özgün Araştırma Makalesi (Original Research Article)
Geliş/Received: 16.12.2017
Kabul/Accepted: 14.01.2018
DOI: http://dx.doi.org/10.17336/igusbd.367106

* Asst. Prof., PhD, Istanbul Gelişim University, Faculty of Economics, Administrative and Social Sciences, Department of Management Information Systems, Istanbul, Turkey,
E-mail: ycelikbilek@gelisim.edu.tr ORCID ID https://orcid.org/0000-0003-0585-1085

literatürde yer alan en iyi sonuçlarla kıyaslanmıştır. Önerilen yöntem uygulandığında elde edilen algoritma sonuçları oldukça rekabetçi ve iyi sonuçlardır.

Anahtar Kelimeler: Uçak İniş, Planlama, Çizelgeleme, Bulanık Küme, Genetik Algoritma

Introduction

Aviation has an increasing importance in the globalizing world. Due to varied high cost related variables, copious of the subtle details are vital. Due to early and late landing costs of aircrafts and those engendered related costs, Aircraft Landing Scheduling (ALS) problem has an increasing importance in both aviation and Academic Community. ALS problem encompasses air traffic control with determining the best schedule for landing and departure of aircrafts.

The main purpose of the problem is to provide aircrafts landing and departure at the closest planned time. Controllers in the air route traffic control centers are responsible for scheduling these planes as an orderly and sufficiently separated sequence, and assigning each plane a landing and departure time[1]. This situation is not only important for the current airport where aircraft is but also important for the destination airport for departure scheduling. Tardiness of the departure of an aircraft leads to altering the schedule of the destination airport and this posture increases the costs of the airports for both current and destination reciprocal. This is also valid for earlier departure time than planned. At this point, aircrafts can fly as fast as possible or can hold in the air with maneuvers and cycles to land the destination airport at the scheduled time. But, aircrafts has an optimum target landing time determined based on their most fuel-efficient airspeed, referred to as the cruise speed[2]. There are a myriad of copious plans and policies with high costs which cannot be altered depending on countries and airports.

The structure of the paper is as follows. In Section 2, literature review within our knowledge is given in detail chronologically. In Section 3, mathematical notations and formulations used in the solution and the proposed algorithm are given. In Section 4, the proposed algorithm is explained and given in detail. In Section 5, the details of Benchmark and the previous results in the literature are given. In Section 6, computational results are given and compared with the best results in the literature. Finally, conclusions and discussions are given in Section 7.

[1] Ke Tang…[et al.], “A multi-objective evolutionary approach to aircraft landing scheduling problems”, IEEE World Congress on Computational Intelligence in Evolutionary Computation, CEC 2008, 2008, p. 3652.
[2] B. S. Girish, “An efficient hybrid particle swarm optimization algorithm in a rolling horizon framework for the aircraft landing problem”, Applied Soft Computing, 44, 2016, p. 200.

Literature Review

According to hardness of ALS problem, there are constraints such as departures, arrivals, other cost related factors and variables with time dependence. This kind of problems are known as NP–hard. Solutions of the problems are obtained from heuristic techniques. There are several benchmarks and problems including specific problems for ALS and lots of heuristic techniques for NP–Hard problems in the literature. In this study, an ALS benchmark derived from OR–library is used. At this point, literature review is considered with the studies used OR–library benchmark or ALS problems. Due to the limited literature about ALS problems and the gap for this problem lead us to scrutinize this problem.

ALS problem has been studied more than a decade but it becomes appealing topic for the researchers mostly in the last five years with the evolution of the information systems and their performances. The benchmark used in this study was firstly introduced in 2000 by Beasley et al.[3] for up to 50 aircrafts and four runways. The authors studied ALS problem as a static case and presented a mixed–integer zero–one formulation. They also presented a heuristic algorithm and showed all computational results. And then in 2004, Beasley et al.[4] used the benchmark up to 500 aircrafts and 5 runways. In the study, the authors considered problems as dynamic aircraft landing problems. They applied three different approaches and reported all results with a comparable table in detail. In 2006, Pinol and Beasley[5] introduced two different heuristic techniques which are scatter search and bionomic algorithm for the ALS problems. These heuristic techniques were formerly used by researchers but have not been applied to ALS. The authors applied the algorithms up to 500 aircrafts and five runways. A hybrid algorithm with Genetic Algorithms (GA) and Ant Colony Optimization (ACO) was proposed by Bencheikh et al. in 2009[6]. They applied the ALS problem as a job shop scheduling problem based on a graphical representation. They applied the hybrid method up to 50 aircrafts and four runways.

In 2011, Yu et al.[7] introduced a real–time schedule method based on cellular automation with the combination of GA. In the study, the authors used GA to improve the results obtained from cellular automation. They showed the results up to 500 aircrafts but one runway. Bencheikh et al.[8] presented a two–

---

[3] J. E. Beasley…[et al.], "Scheduling aircraft landings—the static case", Transportation Science, 34(2), 2000, p. 180-197.
[4] J. E. Beasley…[et al.], "Displacement problem and dynamically scheduling aircraft landings", Journal of the Operational Research Society, 55(1), 2004, p. 54-64.
[5] H. Pinol and J. E. Beasley, "Scatter search and bionomic algorithms for the aircraft landing problem", European Journal of Operational Research, 171(2), 2006, p. 439-462.
[6] Ghizlane Bencheikh…[et al.], "Hybrid method for aircraft landing scheduling based on a job shop formulation", International Journal of Computer Science and Network Security, 9(8), 2009, p. 78-88.
[7] Shenpeng Yu…[et al.], "A real-time schedule method for Aircraft Landing Scheduling problem based on Cellular Automation", Applied Soft Computing, 11(4), 2011, p. 3485-3493.
[8] Ghizlane Bencheikh…[et al.], "Improved ant colony algorithm to solve the aircraft landing problem", International Journal of Computer Theory and Engineering, 3(2), 2011, p. 224-233.

part solution for ALS problems. They introduced a mathematical formulation with a linear and nonlinear objective function to minimize deviations between the landing times and to maximize the objective function to use efficiently the runway. In the second part of the proposed algorithm, they introduced a new heuristic approach with ACO to solve the multiple runways cases. They applied the proposed algorithm up to 50 aircrafts and 4 runways. Dhouib[9] applied a variable neighborhood search with a multi start technique. An adaptive taboo memory was also used to avoid bad solutions and to obtain results earlier. They also applied the proposed algorithm up to 50 aircrafts and four runways.

In 2013, Salehipour et al.[10] introduced a hybrid metaheuristic with simulated annealing framework. The authors applied the proposed approach up to 500 aircrafts and five runways. They obtained the best solutions up to 100 aircrafts with the proposed hybrid approach. For more than 100 aircrafts and multiple runways, they could not obtain best results but they have converged. A polynomial algorithm was introduced by Awasthi et al.[11] to solve the ALS problems with single and multiple runways. The authors applied the polynomial algorithm up to 500 aircrafts and five runways. Their results show that the proposed algorithm get better results than the literature especially for the multiple runways. Bencheikh et al.[12] introduced four hybrid algorithms with using Tabu Search (TS) and GA. These hybrid algorithms were especially introduced for multiple runways. The introduced four hybrid algorithms were applied up to 500 aircrafts and five runways to prove the success of the algorithms. The best result for 500 aircrafts and one runway was obtained by the hybrid algorithm which is based on GA, and then applied TS to avoid the prolongation of search. Phirouzabadi et al.[13] proposed a time segment heuristic method especially for the constraint of the unavailability of runways. They solved the ALS problems with the proposed approach by separating the main problem to sub problems. The proposed approach was applied up to 50 aircrafts and four runways by obtaining the best results.

In 2014, a hybrid algorithm with differential evolution and simple descent algorithm was introduced by Sabar and Kendall[14]. Simple descent algorithm was used to accelerate the solutions of differential evolution. They

[9] Souhail Dhouib, "A multi Start adaptive Variable Neighborhood Search metaheuristic for the aircraft landing problem. 4th International Conference on Logistics (LOGISTIQUA), 197-200.

[10] Amir Salehipour…[et al.], "An efficient hybrid meta-heuristic for aircraft landing problem", Computers & Operations Research, 40(1), 2013, p. 207-213.

[11] Abhishek Awasthi…[et al.], "Aircraft landing problem: An efficient algorithm for a given landing sequence", IEEE 16th International Conference on Computational Science and Engineering (CSE), 2013, p. 20-27.

[12] Ghizlane Bencheikh…[et al.], "Hybrid Algorithms for the Multiple Runway Aircraft Landing Problem", International Journal of Computer Science and Applications, 10(2), 2013, p. 53-71.

[13] M. Mahmoudian…[et al.], "Aircraft Landing Scheduling Based On Unavailability Of Runway Constraint Through A Time Segment Heuristic Method", International Journal of Informatics and Communication Technology (IJ-ICT), 2(3), 2013, p. 175-182.

[14] Nasser R. Sabar and Graham Kendall, "Aircraft landing problem using hybrid differential evolution and simple descent algorithm", IEEE Congress on Evolutionary Computation (CEC), 2014, p. 520-527.

also claimed that proposed hybrid algorithm performs better than differential evolution without the simple descent algorithm.

In 2015, Sabar and Kendal[15] proposed an iterated local search algorithm with multiple perturbation operators to modify the solutions. Perturbation operators provide escaping from local optimum and a new solution for local search procedure. They used a variable neighborhood descent algorithm in the iterated local search procedure. Ghoniem and Farhadi[16] used a two–step solution method to solve the ALS problems in their study. First step includes mixed–integer programming with preprocessing routines and symmetry-defeating hierarchical constraints to improve its performance. In the second step, problem was reformulated as a set partitioning model and a column generation approach was used. The proposed method was applied up to 500 aircraft and five runways. Faye[17] proposed a method with time discretization. Mainstay of the method was an approximation of the separation time matrix which provides lower and upper bounds to be used in algorithm to solve the problem. The computational results of the proposed method was shown up to 500 aircrafts and five runways.

In 2016, a Modified Variable Neighborhood Search algorithm was proposed by Ng and Lee[18]. Branch and bound algorithm was also used to avoid from local optimum and to reduce the calculation times. The proposed approach was applied only up to 20 aircrafts and 4 runways. Bencheikh et al.[19] used a memetic algorithm to obtain the solutions of the dynamic multiple runways ALS problems. In the study, the authors mainly focused on dynamic version of the ALS problems. They combined the memetic algorithm with ACO and a local search. They applied the proposed combined algorithm up to 50 aircrafts and four runways. Ji et al.[20] developed a sequence searching and evaluation algorithm. In the study, the authors classified the different formulations of ALS problems by the objectives and constraints. They used only up to 44 aircraft and one runway problems to apply the developed algorithm and compare the results. A hybrid Particle Swarm Optimization (PSO) was introduced by Girish[21]. Evaluating the effectiveness of the proposed algorithm

[15] Nasser R. Sabar and Graham Kendall, "An iterated local search with multiple perturbation operators and time varying perturbation strength for the aircraft landing problem", Omega, 56, 2015, p. 88-98.

[16] Ahmed Ghoniem and Farbod Farhadi, "A column generation approach for aircraft sequencing problems: a computational study", Journal of the Operational Research Society, 66(10), 2015, p. 1717-1729.

[17] Alain Faye, "Solving the aircraft landing problem with time discretization approach", European Journal of Operational Research, 242(3), 2015, p. 1028-1038.

[18] K. K. H. Ng and C. K. M. LEE, "A modified Variable Neighborhood Search for aircraft Landing Problem", IEEE International Conference on Management of Innovation and Technology (ICMIT), 2016, p. 127-132.

[19] Ghizlane Bencheikh…[et al.], "A memetic algorithm to solve the dynamic multiple runway aircraft landing problem", Journal of King Saud University-Computer and Information Sciences, 28(1), 2016, p. 98-109.

[20] Xiao Peng Ji…[et al.], "Sequence searching and evaluation: a unified approach for aircraft arrival sequencing and scheduling problems", Memetic Computing, 8(2), 2016, p. 109-123.

[21] B.S. Girish, ibid, p. 200–221.

was executed by applying it up to 500 aircrafts and 5 runways. The results of the proposed algorithm were also compared with two other PSO algorithms adapted from the literature.

In 2017, Abdullah et al.[22] proposed a harmony search algorithm for the solutions of the multiple runways ALS problem. The proposed approach was applied up to 500 aircrafts and 5 runways to show the effectiveness of the algorithm. The results show that the proposed approach is successful especially small size problems.

Mathematical Formulation

Notations and their explanations used for mathematical formulations are given in Table 1.

Table 1: Notation and Explanations

Notations	Explanations
n	Number of Aircrafts
N	Set of Aircrafts
m	Number of Runways
M	Set of Runways
TA_i	Appearance Time of the Aircraft *i*
TS_i	Scheduled Landing Time of the Aircraft *i*
TE_i	Earliest Landing Time of the Aircraft *i*
TT_i	Target Landing Time of the Aircraft *i*
TL_i	Latest Landing Time of the Aircraft *i*
TD_{ij}	Separation Time for Landing of Aircraft *j* after Aircraft *i*
PE_i	Penalty Cost per Unit of Time for Landing Earlier than Target Time
PL_i	Penalty Cost per Unit of Time for Landing Later than Target Time
P	Total Penalty Cost
DE_i	Time Difference of Earlier Landing
DL_i	Time Difference of Later Landing
R_i	Landing Runway of Aircraft *i*
C_{ijk}	Membership of Aircraft *i* for Cluster *j* of Runway *k*
S_{i5}	Solution Matrix of the Problem (Columns: $R_i, TS_i, TE_i, TT_i, TL_i$)

The objective function of the ALS problems is the same as other NP–hard problems. The objective is to obtain the minimum penalty cost which is

[22] Omar Salim Abdullah…[et al.], "Harmony Search Algorithm for the Multiple Runways Aircraft Landing Scheduling Problem", Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(3-7), 2017, p. 59–65.

calculated by using the deviations of the target landing times in ALS problems. The penalty cost function or the objective function of the problem is given in Eq. (1).

$$\min \quad (P) = \sum_{i=1}^{n}(DE_i PE_i + DL_i PL_i) \tag{1}$$

While calculating the total penalty cost function in Eq. (1), constraints are given in Eq. (2) and Eq. (3) have to be ensured.

$$TE_i \leq TS_i \leq TL_i, \quad \forall\, i \in N \tag{2}$$

$$TS_i \geq TS_{(i-1)} + TD_{ij} \tag{3}$$

where *(i-1)* represents the aircraft landing before the aircraft *i* on the same runway. Because of the proposed algorithm design, the runway constraints are not included in the model used in this study.

The Proposed Algorithm and Foundations

In this section, the proposed fuzzy cluster based genetic algorithm (FCGA) approach and its foundations are given.

Fuzzy Set Theory

Fuzzy Set Theory (FST) was firstly introduced by Zadeh[23]. Zadeh has provided handling with the problems by the help of membership functions. Since FST was introduced, it has been implemented a wide range of fields. A basic notation of a fuzzy set is shown in Eq. (4).

$$A = \{\langle x, \mu_A(x)\rangle \ x \in [0,1]\} \tag{4}$$

where A is a fuzzy set, $\mu_A(x)$ is the membership function which describes the membership degree of x in set A. In this study, while applying the proposed GA, the fuzzy clusters are used for the time ranges. This provides improving the initial bad solutions rapidly. None of the memberships is left 0, not to let escape the better solutions, because of the different penalty costs per minute.

Generally, in the literature, triangular fuzzy numbers are used in the studies. An example of membership functions by using triangular fuzzy numbers is shown in the Figure 1.

[23] Lotfi A. Zadeh, "Fuzzy Sets", Information and Control. 8, 1965, 338-353.

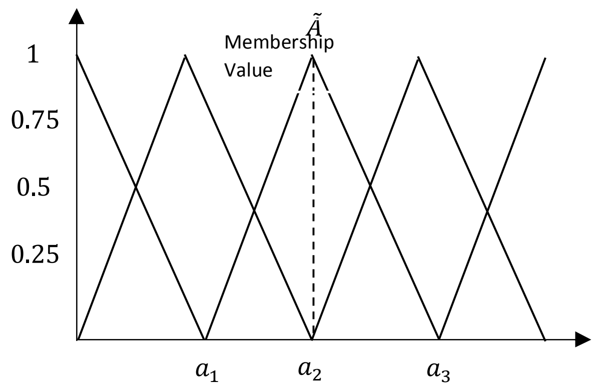

Figure 1: An Example of Membership Functions of the Triangular Fuzzy Numbers

The membership function of *x* for the set A in the Figure 1 is calculated by using Eq. (5).

$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & x \in (-\infty, a_1) \\ \dfrac{x - a_1}{a_2 - a_1} & x \in [a_1, a_2] \\ \dfrac{a_3 - x}{a_3 - a_2} & x \in [a_2, a_3] \\ 0 & x \in (a_3, +\infty) \end{cases} \tag{5}$$

Genetic Algorithm

Solutions based on GA were firstly introduced by Holland[24]. The method models the problems inspired by genetics. Variables in the problems are defined as genes and chromosomes, and solutions are obtained by using techniques such as crossover, mutation etc. like in the genetics. In the following paragraphs, basic genetic operators and definitions used in the proposed algorithm are given briefly.

Fitness Function (FF): FF is calculated by using the objective function which is penalty cost function in this study. Generally, in the literature, FF is equal to objective function in maximization problems, in contrast in this study FF used for minimizing the objective function which is given in Eq. (6).

[24] John H. Holland, Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, Ann Arbor, University of Michigan Press, 1975.

$$\boldsymbol{FF_i} = \frac{\mathbf{1}}{\boldsymbol{P_i}} \tag{6}$$

Mutation: altering the random genes by using a predefined percentage value. Mutation is mostly preferred in GA-based solutions, because it eliminates trapping to the local optimums.

Roulette Wheel Selection (RWS): RWS technique is used for the selection of parents to form the next generations. Two of the population is selected with this technique, and then a child solution is generated by using crossover the parent solutions. RWS increases the selection chance of better solutions, and worse solutions vanished faster from the population. An example of RWS is given in Eq. (7) with the penalty function which means that minimum value is better.

$$\boldsymbol{c_i} = \frac{\mathbf{1}/\boldsymbol{P_i}}{\sum_{\boldsymbol{j}}\left(\mathbf{1}/\boldsymbol{P_j}\right)} \tag{7}$$

where c_i is selection chance of the parent solution *i* and P_i is the penalty cost of parent *i*.

Crossover: Crossover techniques are used to generate child solutions from parent solutions. There are three main crossover techniques, which are one point crossover, two point crossover and uniform crossover, have been using in the literature. Examples of one point crossover and two point crossover are given in Figure 2.

Parent 1

1	0	0	1

Parent 2

0	1	1	1

Child 1

1	0	1	1

Child 2

0	1	0	1

Parent 1

1	0	0	1

Parent 2

0	1	1	1

Child 2

0	0	0	1

Child 2

1	1	1	1

Figure 2: One and Two Point Crossovers

In this study, two point crossover is used according to the cluster memberships. This means that if a crossover does not satisfy the memberships, it will not be happened.

The Proposed Algorithm

In this sub–section, proposed algorithm for the solutions of ALS problems is given. Proposed algorithm can be considered as solving problems dividing them to sub–problems. But the main approach of the algorithm is that penetrating members of each cluster to another clusters by using membership values. With this perspective, aircrafts can be controlled under the clusters in which they have higher membership value. Not to let escape the better solutions, membership values of the aircrafts for each cluster have values greater than zero which are also depending of their earliest and latest landing times. If a cluster is out of the aircraft's earliest or latest landing time interval, then the membership value for the cluster is zero. Memberships are also controlled together with the probabilities in the solution process of the algorithm.

The beginning of the algorithm is determining the number of time clusters in the solution. Number of clusters can be determined depending on the complexity of the problem. According to similar problems in the literature, larger clusters get closer the problem to main problem and smaller clusters make the calculations of the problem heavy which effects the solution times. After determining the number of time clusters, membership values of each aircraft are calculated for all clusters of runways. Then, the proposed GA is executed by including fuzzy clusters. Flowchart of the proposed FCGA is shown in Figure 3.

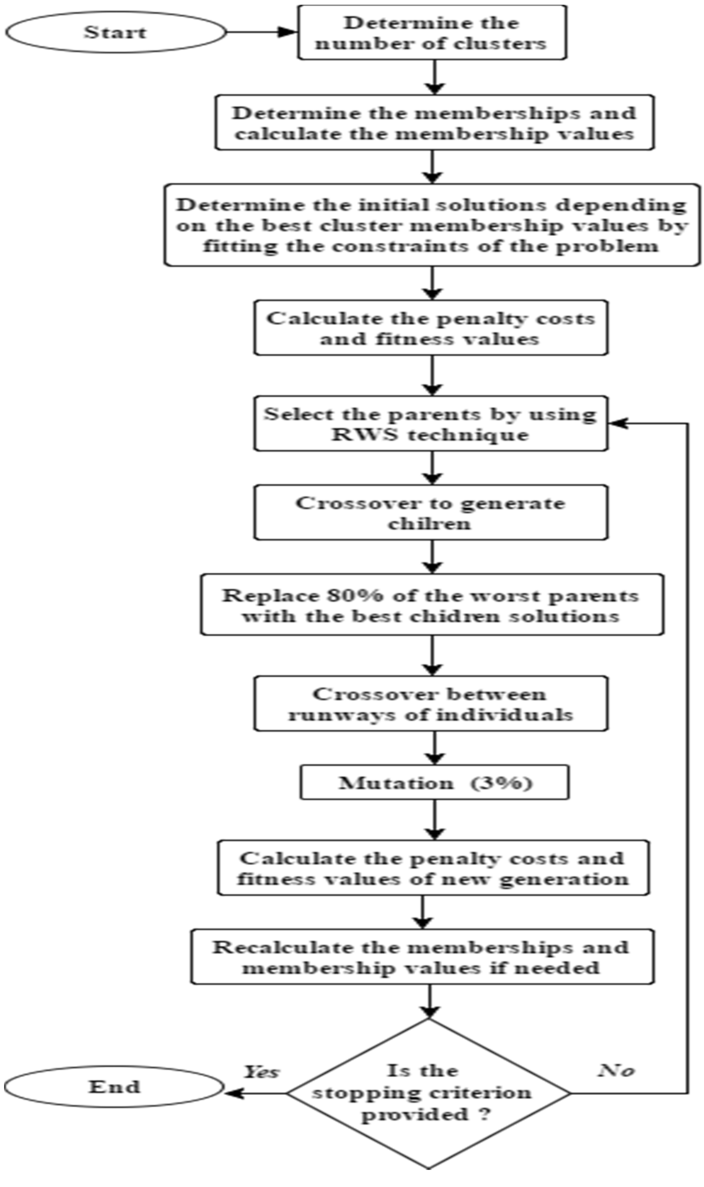

Figure 3: Flowchart of the Proposed FCGA

Table 2: Fuzzy Cluster Based Genetic Algorithm

Read $n, TE_i, TT_i, TL_i, TD_{ij}, PE_i, PL_i$ from database
Determine m
Generate S_{i5}
Determine Clusters
Calculate C_{ijk} as an initial
Calculate initial TS_i according to memberships and constraints
Calculate P and FF of initial solution
do
  for (all population)
    select parents with RWS
    crossover parents to generate children considering memberships of the clusters
  end for
  for (%80 of population)
    Replace the worst parent with the best child
  end for
  for (all population)
    select two runway of the individual randomly
    crossover runways considering memberships of the clusters
  end for
  for (all population)
    mutation with 3% possibility
  end for
  check and fix TS_i within S_{i5}
  Recalculate P and FF of all population
  Recalculate the memberships and membership values
while ($P_{new\ best} < P_{old\ best}$)
Print the best solution

In the proposed algorithm, probabilities, crossovers and mutations are restricted by fuzzy clusters. Cut values of the fuzzy cluster memberships can be determined depending on the problem or by user. If a cut value is never defined, algorithm works as a plain GA. Because, as it was reported before, not to let escape better solutions, none of the memberships left zero. The main difference of the algorithm is controlling the GA by using changing fuzzy clusters and cut values. Computational results of the proposed approach are given in detail with the comparison of the best results in Section 6.

Benchmark and Results in the Literature

This section is allocated to introduce the benchmark which is used to evaluate the proposed algorithm. Benchmark instances will be given with the results as in the literature.

Results in the Literature

The benchmark instances have been taken from the OR–library website[25]. It was firstly used in 2000 by Beasley et al.[26] up to 50 aircrafts and four runways. And then in 2004, Beasley et al.[27] used the benchmark up to 500 aircrafts and 5 runways. The benchmark has 13 problems up to 500 aircrafts and 5 runways. Details of the benchmark and the results of the benchmark problems in the literature are given in Table 3. Only comparable literature results are reported in the table. Results which are only up to 50 aircrafts are eliminated in the table.

Table 3: Airland Benchmark and the Comparative Results in the Literature

PN	n	m	Beasley et al. 2000	Beasley et al. 2004	Pinol & Beasley 2006	Salehipour et al. 2013	Awasthi et al. 2013	Bencheikh et al. 2013	Sabar & Kendall 2015	Ghoniem & Farhadi 2015	Girish 2016	Best
A1	10	1	700	740	700	720	700	700	701	700	700	700
		2	90	90	90	90	90	90	90	90	90	90
		3	0	0	0	0	0	0	0	0	0	0
A2	15	1	1480	1730	1480	1480	1480	1480	1485	1480	1480	1480
		2	210	210	210	210	210	210	211	210	210	210
		3	0	0	0	30	0	0	1	0	30	0
A3	20	1	820	940	820	820	820	820	825	820	820	820
		2	60	60	60	70	60	60	64	60	70	60
		3	0	0	0	10	0	0	1	0	10	0
A4	20	1	2520	2700	2520	2520	2520	2520	2523	2520	2520	2520
		2	640	680	640	660	640	640	643	640	660	640
		3	130	130	130	160	130	130	136	130	160	130
		4	0	0	0	30	0	0	0	0	30	0
A5	20	1	3100	3180	3100	3100	3100	3100	3102	3100	3100	3100
		2	650	680	650	650	650	650	658	650	650	650
		3	170	240	170	170	170	170	176	170	170	170
		4	0	0	0	30	0	0	1	0	30	0
A6	30	1	24442	24442	24442	24442	24442	24442	24458	24442	24442	24442
		2	554	809	554	554	554	600	564	554	554	554
		3	0	0	0	0	0	0	0	0	0	0
A7	44	1	1550	3974	1550	1550	1550	1550	1558	1550	1550	1550

[25] OR–Library, < http://people.brunel.ac.uk/~mastjjb/jeb/orlib/airlandinfo.html> Accessed 01.02.2017.
[26] J. E. Beasley…[et al.], ibid.
[27] J. E. Beasley…[et al.], ibid.

		2	0	0	0	0	0	0	1	0	0	0
A8	50	1	1950	2000	1950	1950	1995	1950	1959	1950	1950	1950
		2	135	135	135	135	135	135	143	135	135	135
		3	0	0	0	10	0	0	2	0	10	0
A9	100	1		7848.4	5611.7	6091.8	5703.5	6890	5614.0	5611.9	5611.7	5611.7
		2		573.2	452.9	450.2	444.1	453	455.2	444.1	444.1	444.1
		3		88.7	75.7	75.7	75.7	76	82.7	75.7	75.7	75.7
		4		0.0	0.0	0.0	0.0	0	2.2	0.0	0.0	0.0
A10	150	1		17726.0	12329.3	12329.3	13515.6	14230	12344.3	12310.7	12292.2	12292.2
		2		1372.2	1288.7	1219.2	1203.7	1320	1281.5	1143.7	1144.0	1143.7
		3		246.1	220.7	206.4	205.2	246	211.5	205.2	206.2	205.2
		4		34.2	34.2	35.2	34.2	42	41.2	34.2	35.2	34.2
		5		0.0	0.0	1.0	0.0	0	1.1	0.0	1.0	0.0
A11	200	1		19327.4	12418.3	12418.3	13401.5	16790	12419.5	12418.3	12418.3	12418.3
		2		1683.7	1540.8	1416.8	1400.6	1830	1416.0	1330.9	1330.9	1330.9
		3		333.5	280.8	272.9	253.1	320	276.4	253.0	253.0	253.0
		4		69.6	54.5	54.5	54.5	55	65.5	54.5	54.5	54.5
		5		0.0	0.0	0.0	0.0	0	1.2	0.0	0.0	0.0
A12	250	1		25049.2	16209.7	16209.7	17346.4	16242	16216.9	16302.2	16122.1	16122.1
		2		2204.9	1961.3	1961.3	1753.6	1980	1969.6	1695.6	1695.6	1695.6
		3		430.5	290.0	279.7	233.4	324	290.9	221.9	221.9	221.9
		4		2.8	3.4	3.4	2.4	3	7.1	2.4	2.4	2.4
		5		0.0	0.0	0.0	0.0	0	1.2	0.0	0.0	0.0
A13	500	1		58329.6	44832.3	41448.1	43052.0	44832	41391.4	37294.7	37064.1	37064.1
		2		4897.9	5501.9	5475.8	4593.7	5502	5470.1	3920.3	3921.3	3920.3
		3		821.8	1108.5	744.9	712.8	1140	1114.4	673.8	673.8	673.8
		4		123.3	188.4	100.6	89.9	188	104.8	89.9	89.9	89.9
		5		0.0	7.3	3.8	0.0	7	9.1	0.0	0.0	0.0

*PN: Problem Name, A1: Airland1, … , A13: Airland13.

An Example of Benchmark Data

The ALS problem data set for the Airland1 problem is given in Table 4. The ALS problem data sets are generally represented like in the Table 4 without special conditions of the problems, airports and companies. First column of the table, represents the aircraft numbers. Columns between 2 and 11 show the separation times between the landings of following aircrafts. Columns between 12 and 15 show the appearance time, the earliest landing time, the target landing time and the latest landing time respectively. 16th column is for the penalty cost per unit of time for landing earlier than target time and the 17th columns is for the penalty cost per unit of time for landing later than target time.

Table 4: Airland1 Data Set

n	1	2	3	4	5	6	7	8	9	10	TA_i	TE_i	TT_i	TL_i	PE_i	PL_i
1	X	3	15	15	15	15	15	15	15	15	54	129	155	559	10	10
2	3	X	15	15	15	15	15	15	15	15	120	195	258	774	10	10
3	15	15	X	8	8	8	8	8	8	8	14	89	98	510	30	30
4	15	15	8	X	8	8	8	8	8	8	21	96	106	521	30	30
5	15	15	8	8	X	8	8	8	8	8	35	110	123	555	30	30
6	15	15	8	8	8	X	8	8	8	8	45	120	135	576	30	30
7	15	15	8	8	8	8	X	8	8	8	49	124	138	577	30	30
8	15	15	8	8	8	8	8	X	8	8	51	126	140	573	30	30
9	15	15	8	8	8	8	8	8	X	8	60	135	150	591	30	30
10	15	15	8	8	8	8	8	8	8	X	85	160	180	657	30	30

Computational Results

In this section, computational results of the study are given. Calculations of the proposed algorithm were implemented in Java and executed on a computer with Intel(R) Core (TM) i7–4720HQ 2.60 GHz, 16 GB RAM and windows 8.1 professional 64 bit operating system.

Because of the size of the problems all of the results cannot be given but obtained results are given in detail as much as possible. In this section, detailed schedule results of the first two problems for one and two runways are given as examples. And then, obtained penalty costs for all problems are given with the best results in the literature as a comparable table.

Detailed Schedules of Airland1 and Airland2 for One Runway

Calculated schedules of Airland1 and Airland2 problems for one runway are given in Table 5. First column of the table is ranking of the aircrafts, second column is scheduled landing time of the aircrafts obtained by the proposed approach, third column is the target landing time given in the problem, fourth column is the difference between the scheduled landing time and the target landing time of the aircrafts, fifth column is the penalty cost per unit of time before or after the target landing time, and sixth column is total penalty cost for each aircraft. Penalty cost per unit of time before or after the target landing time is given in one column, because both of them are same in these problems. Total penalty costs of both problem are given at the bottom of the table.

Table 5: Schedules of A1 and A2 for One Runway

Airland1						Airland2					
A*	Scheduled Landing Time	Target Landing Time	Dif.*	Penalty	Total Penalty per Aircraft	A*	Scheduled Landing Time	Target Landing Time	Dif.	Penalty	Total Penalty per Aircraft
3	98	98	0	30	0	3	90	93	3	30	90
4	106	106	0	30	0	4	98	98	0	30	0
5	118	123	5	30	150	5	106	111	5	30	150
6	126	135	9	30	270	6	114	120	6	30	180
7	134	138	4	30	120	7	122	121	1	30	30
8	142	140	2	30	60	8	130	120	10	30	300
9	150	150	0	30	0	9	138	128	10	30	300
1	165	155	10	10	100	10	151	151	0	30	0
10	180	180	0	30	0	14	171	171	0	30	0
2	258	258	0	10	0	13	181	181	0	30	0
						1	196	155	41	10	410
						2	250	250	0	10	0
						12	313	313	0	10	0
						11	341	341	0	10	0
						15	344	342	2	10	20
Total Penalty Cost for Airland1					700	Total Penalty Cost for Airland2					1480

* A: Aircraft Number, Dif.: Difference between scheduled and target landing time

Detailed Schedules of Airland1 and Airland2 for Two Runways

Calculated schedules of Airland1 and Airland2 problems for two runways are given in Table 6. First column of the table is numbers of the runways and other columns of the table are the same as Table 6. Total penalty costs are given at the bottom of the table for each problem.

Table 6: Schedules of A1 and A2 for Two Runway

Airland1							Airland2						
Runway	A	SLT	TLT	D	P	TP	Runway	A	SLT	TLT	D	P	TP
	3	98	98	0	30	0		3	93	93	0	30	0
	4	106	106	0	30	0		5	111	111	0	30	0
1	7	138	138	0	30	0		6	120	120	0	30	0
	9	150	150	0	30	0	1	9	128	128	0	30	0
	10	180	180	0	30	0		10	151	151	0	30	0
	5	123	123	0	30	0		14	171	171	0	30	0
	6	132	135	3	30	90		15	342	342	0	10	0
2	8	140	140	0	30	0		4	98	98	0	30	0
	1	155	155	0	10	0		8	120	120	0	30	0
	2	258	258	0	10	0		7	128	121	7	30	210
								1	155	155	0	10	0
							2	13	181	181	0	30	0
								2	250	250	0	10	0
								12	313	313	0	10	0
								11	341	341	0	10	0
Total Penalty Cost for Airland1						90	Total Penalty Cost for Airland2						210

All Comparable Results of the Proposed Approach

Computational results of the Benchmark by using the proposed FCGA approach are given in Table 7 with the comparison of the best results in the literature. Fourth column named *'best'* is the best results obtained by Table 3. Fifth column shows the results obtained by executed the proposed FCGA approach. Last column of the table (Δ(%)) represents the difference between the obtained results and the best results.

Table 7: Computational Results

P	*n*	*m*	Best	Proposed FCGA	Δ(%)
A1	10	1	700	700	0
		2	90	90	0
		3	0	0	0
A2	15	1	1480	1480	0
		2	210	210	0
		3	0	0	0
A3	20	1	820	820	0
		2	60	60	0
		3	0	0	0

A4	20	1	2520	2520	0
		2	640	640	0
		3	130	130	0
		4	0	0	0
A5	20	1	3100	3100	0
		2	650	650	0
		3	170	170	0
		4	0	0	0
A6	30	1	24442	24442	0
		2	554	554	0
		3	0	0	0
A7	44	1	1550	1550	0
		2	0	0	0
A8	50	1	1950	1950	0
		2	135	135	0
		3	0	0	0
A9	100	1	5611.7	5687.5	0.01
		2	444.1	453.2	0.02
		3	75.7	76.2	0.01
		4	0.0	0.0	0.00
A10	150	1	12292.2	12615.8	0.03
		2	1143.7	1136.3	-0.01
		3	205.2	205.4	0.00
		4	34.2	34.2	0.00
		5	0.0	0.0	0.00
A11	200	1	12418.3	12759.1	0.03
		2	1330.9	1347.6	0.01
		3	253.0	257.3	0.02
		4	54.5	57.3	0.05
		5	0.0	0.0	0.00
A12	250	1	16122.1	16237.7	0.01
		2	1695.6	1701.7	0.00
		3	221.9	221.9	0.00
		4	2.4	2.4	0.00
		5	0.0	1.2	0.00
A13	500	1	37064.1	37183.8	0.00
		2	3920.3	4125.1	0.05
		3	673.8	703.3	0.04
		4	89.9	89.9	0.00
		5	0.0	0.0	0.00

As a result of our findings, small size problems between A1 and A8 can be solved no more than 2 seconds. However, problems with larger number of aircrafts and smaller number of runways have challenging difficulties. While number of runways are increasing, reaching the solution time decreases yet it is still challenging for large problems. During the implementation of our algorithm, problems with large number of aircrafts and small number of runways were limited to 3600 seconds. Especially, implementation of problems with larger number of aircrafts and 5 runways were carried out no more than 5 seconds like small size problems.

The best results in the literature for the problems between A1 and A8 were obtained by the proposed FCGA approach in this study. Literature review also shows that almost all of the proposed methods can find the best results for these small size problems. For the problems between A9 and A13, the FCGA approach converges the best results. Half of these problems could be obtained without any difference. Quarter of these problems could be obtained with 1% difference than the best results. And, the rest of the results shown in Table 7 and whose difference is more than 1% could be obtained in a limit of 3600 seconds. When all of the obtained results by FCGA is evaluated together, it can be stated that the proposed FCGA approach is competitive and effective.

Conclusion

The ALS problem known as NP–hard problem for heuristic optimization is an important issue for both aviation and air traffic control. The aim of the problem is to minimize penalty cost by decreasing the deviations from the landing time under some constraints. The ALS problem has two main parts which make finding the best solution of the problem really hard. First part is sorting the aircrafts. And then, the second part is that calculating the optimum landing time of the aircrafts.

In this study, the implementation of fuzzy set theory to GA was focused on. Firstly, foundations of the proposed algorithm, which are fuzzy set theory and GA, were given briefly. And then, generation of fuzzy clusters by satisfying the constraints was defined to use in the algorithm. The effectiveness and validation of the proposed algorithm were tested by executing the algorithm with a benchmark set which is up to 500 aircrafts and 5 runways. The proposed algorithm has a satisfactory performance and the results are competitive with the literature.

For further research, the proposed approach can be also implemented to other heuristic techniques or adapted with different operators to improve. The proposed FCGA approach can be implemented to various scheduling problems from different fields with small modifications depending on problem types and constraints.

REFERENCES

ABDULLAH, O. S., ABDULLAH, S., & SARIM, H. M., "Harmony Search Algorithm for the Multiple Runways Aircraft Landing Scheduling Problem", **Journal of Telecommunication, Electronic and Computer Engineering (JTEC)**, 9(3-7), 2017, 59–65.

AWASTHI, A., Kramer, O., & LASSIG, J., "Aircraft landing problem: An efficient algorithm for a given landing sequence", **IEEE 16th International Conference on Computational Science and Engineering (CSE),** 2013–December, 20-27.

BEASLEY, J. E., Krishnamoorthy, M., Sharaiha, Y. M., & Abramson, D., "Scheduling aircraft landings—the static case", **Transportation Science**, 34(2), 2000, 180-197.

BEASLEY, J. E., KRISHNAMOORTHY, M., SHARAIHA, Y. M., & ABRAMSON, D., "Displacement problem and dynamically scheduling aircraft landings", **Journal of the Operational Research Society**, 55(1), 2004, 54-64.

BENCHEIKH, G., BOUKACHOUR, J., ALAOUI, A. E. H., & KHOUKHI, F. E., "Hybrid method for aircraft landing scheduling based on a job shop formulation", **International Journal of Computer Science and Network Security**, 9(8), 2009, 78-88.

BENCHEIKH, G., BOUKACHOUR, J., & ALAOUI, A. E. H., "Improved ant colony algorithm to solve the aircraft landing problem", **International Journal of Computer Theory and Engineering**, 3(2), 2011, 224-233.

BENCHEIKH, G., EL KHOUKHI, F., BACCOUCHE, M., BOUDEBOUS, D., BELKADI, A., & OUAHMAN, A. A., "Hybrid Algorithms for the Multiple Runway Aircraft Landing Problem", **International Journal of Computer Science and Applications**, 10(2), 2013, 53-71.

BENCHEIKH, G., BOUKACHOUR, J., & ALAOUI, A. E. H., "A memetic algorithm to solve the dynamic multiple runway aircraft landing problem", **Journal of King Saud University-Computer and Information Sciences**, 28(1), 2016, 98-109.

DHOUIB, S., "A multi Start adaptive Variable Neighborhood Search metaheuristic for the aircraft landing problem", **4th International Conference on Logistics (LOGISTIQUA)**, 2011–May, 197-200.

FAYE, A., "Solving the aircraft landing problem with time discretization approach", **European Journal of Operational Research**, 242(3), 2015, 1028-1038.

GHONIEM, A., & FARHADI, F., "A column generation approach for aircraft sequencing problems: a computational study", **Journal of the Operational Research Society**, 66(10), 2015, 1717-1729.

GIRISH, B. S., An efficient hybrid particle swarm optimization algorithm in a rolling horizon framework for the aircraft landing problem, **Applied Soft Computing**, 44, 2016, 200-221.

HOLLAND, J. H., Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, Ann Arbor, **University of Michigan Press**, 1975.

JI, X. P., CAO, X. B., & TANG, K., "Sequence searching and evaluation: a unified approach for aircraft arrival sequencing and scheduling problems", **Memetic Computing**, 8(2), 2016, 109-123.

MAHMOUDIAN, M., AMINNAYERI, M., & MIRZADEH, A., "Aircraft Landing Scheduling Based On Unavailability Of Runway Constraint Through A Time

Segment Heuristic Method”, **International Journal of Informatics and Communication Technology (IJ-ICT)**, 2(3), 2013, 175-182.

NG, K. K. H., & LEE, C. K. M., “A modified Variable Neighborhood Search for aircraft Landing Problem”, **IEEE International Conference on Management of Innovation and Technology (ICMIT)**, 2016–September, 127-132.

OR–Library, http://people.brunel.ac.uk/~mastjjb/jeb/orlib/air landinfo.html, Accessed 01.02.2017.

PINOL, H., & BEASLEY, J. E., “Scatter search and bionomic algorithms for the aircraft landing problem”, **European Journal of Operational Research**, 171(2), 2006, 439-462.

SABAR, N. R., & KENDALL, G., “Aircraft landing problem using hybrid differential evolution and simple descent algorithm”, **IEEE Congress on Evolutionary Computation (CEC)**, 2014–July, 520-527.

SABAR, N. R., & KENDALL, G., “An iterated local search with multiple perturbation operators and time varying perturbation strength for the aircraft landing problem”, **Omega**, 56, 2015, 88-98.

SALEHIPOUR, A., MODARRES, M., & NAENI, L. M., “An efficient hybrid meta-heuristic for aircraft landing problem”, **Computers & Operations Research**, 40(1), 2013, 207-213.

TANG, K., WANG, Z., CAO, X., & ZHANG, J., “A multi-objective evolutionary approach to aircraft landing scheduling problems”, **IEEE World Congress on Computational Intelligence in Evolutionary Computation, CEC 2008**, 2008–June, 3650-3656.

YU, S. P., CAO, X. B., & ZHANG, J., “A real-time schedule method for Aircraft Landing Scheduling problem based on Cellular Automation”, **Applied Soft Computing**, 11(4), 2011, 3485-3493.

ZADEH, L. A., Fuzzy sets, **Information and Control**, 8, 1965, 338-353.

Özet

Uçak iniş planlaması(UİP) problemi hem havacılığın hemde hava trafik kontrolünün en önemli bölümlerinden birisidir. Problemin esas amacı, bazı kısıtlar altında ihlal maliyetlerinin minimize edilerek uçakların iniş zamanlarının belirlenmesidir. Problemde, uçakların her biri için yakıt, hava hızı ve maliyet ile ilgili iniş zamanlarına dayalı spefikasyonların olduğu optimum hedefler söz konusudur. İniş zamanı hedefinden sapmalar uçağın ve problemin ihlal maliyetlerinin artmasına neden olmaktadır. Burada özetlenmiş olan amaç doğrultusunda da bu çalışmada, planlanan iniş zamanlarından sapmaları en aza indirgeyerek uçakların iniş zamanlarının planlanması üzerinde durulmuştur.

Bunun için öncelikle detaylı bir literatür çalışması gerçekleştirilmiştir. Fakat mevcut literatürde fazla makale yer almaması dolayısıyla da erişilebilmiş olan makaleler olabildiğince tüm yönleriyle detaylı bir şekilde incelenmeye çalışılmıştır. Literatür taramasının aktarılmasının ardından, önerilmiş olan çözüm yönteminin ve algoritmanın daha anlaşılır bir hale gelmesi adına yöntemlerde ve algoritmalarda kullanılan

matematiksel notasyonlar ve formüller açıklamalarıyla birlikte üçüncü bölümde aktarılmıştır. Çalışmanın dördüncü bölümünde, bulanık küme teorisi ve genetik algoritmalar, yöntemlerde yer alan önemli formüllerle de birlikte özet bir şekilde aktarılmıştır. Aynı bölümün devamında daha sonra, önerilmekte olan yöntem akış diyagramı ve kaba kodları verilerek olabildiğince detaylı bir şekilde aktarılmıştır. Önerilmekte olan yöntemin test problemiyle uygulanmasına geçmeden önce literatürde yer alan, problemlerden elde edilmiş sonuçlar karşılaştırmalı olarak tablo halinde verilmiştir. Altıncı bölümde öncelikle, veri setinin ve sonuçların anlaşılabilirliğini de arttırmak adına bir ve iki pist için ilk iki test probleminin çizelgeleme sonuçları tablo halinde detaylı olarak verilmiştir. Son olarak da, tüm test problemlerinin uygulama sonuçlarına ait elde edilmiş olan en iyi sonuçlar, literatürde yer alan en iyi sonuçlarla karşılaştırmalı olarak tablo halinde verilmiş ve elde edilen tüm sonuçlar ayrıntılı olarak değerlendirilmiştir. Çalışmada gerçekleştirilen uygulamalar, 500 uçağa ve 5 piste kadar 13 adet farklı problemin yer aldığı UİP test problemlerinde bulanık küme temelli bir genetik algoritma yaklaşımıyla gerçekleştirilmiştir. Kullanılmış olan test problemi Yöneylem Araştırması Kütüphanesinden (OR–library) elde edilmiştir. Önerilen algoritma ile elde edilen detaylı sonuçlar literatürede yer alan en iyi sonuçlarla karşılaştırıldığında, elde edilen algoritma sonuçlarının oldukça rekabetçi ve iyi sonuçlar ürettiği gözlemlenmiştir.