ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

Aircraft Parking Optimization Using Genetic Algorithm

M.Sc. THESIS

Burak GÜLER

Department of Mathematical Engineering

Mathematical Engineering Programme

Thesis Advisor: Dr. Eti MİZRAHİ

DECEMBER 2017

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

Aircraft Parking Optimization Using Genetic Algorithm

M.Sc. THESIS

Burak GÜLER
(509131052)

Department of Mathematical Engineering

Mathematical Engineering Programme

Thesis Advisor: Dr. Eti MİZRAHİ

DECEMBER 2017

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

Genetik Algoritma Kullanarak Uçak Park Yeri Optimizasyonu

YÜKSEK LİSANS TEZİ

Burak GÜLER
(509131052)

Matematik Mühendisliği Anabilim Dalı

Matematik Mühendisliği Programı

Tez Danışmanı: Dr. Eti MİZRAHİ

ARALIK 2017

Burak GÜLER, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 509131052 successfully defended the thesis entitled "Aircraft Parking Optimization Using Genetic Algorithm", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Dr. Eti MİZRAHİ** Istanbul Technical University	.....
Jury Members :	**Assoc. Prof. Dr. Ali ERCENGİZ** Istanbul Technical University	.....
	Prof. Dr. Ayşe Hümeyra BİLGE Kadir Has University	.....
		.....

Date of Submission : **17 November 2017**
Date of Defense : **12 December 2017**

To my mom and lecturer Dr. Eti Mizrahi,

FOREWORD

I would like to thank my advisor Dr. Eti Mizrahi for sparing her precious time to guide and support me splendidly on my path to write this thesis. I would like to thank Burak Durukan as well for providing technical support by writing a PHP code for this thesis in particular.

December 2017 Burak GÜLER

TABLE OF CONTENTS

Page

FOREWORD ..... **ix**
TABLE OF CONTENTS ..... **xi**
ABBREVIATIONS ..... **xiii**
LIST OF TABLES ..... **xv**
LIST OF FIGURES ..... **xvii**
SUMMARY ..... **xix**
ÖZET ..... **xxi**
1. INTRODUCTION **1**
1.1 Motivation of Thesis 2
1.2 Literature Review 2
2. Genetic Algorithm **5**
2.1 Fundamental of the GA 7
2.2 Main Factor of Genetic Algorithm 8
2.2.1 Representation of the Solution 9
2.2.1.1 Binary Code 9
2.2.1.2 Real-Valued Code 9
2.2.1.3 Permutation Code 9
2.3 Genetic Operators 10
2.3.1 Selection Operator 10
2.3.1.1 Fitness Proportional Selection 10
2.3.1.2 Tournament Method 11
2.3.1.3 Steady - State Selection 12
2.3.1.4 Ranking Method 12
2.4 Crossing Over Operator 13
2.4.1 1-Point Cross-over 15
2.4.2 K-point Cross-over 15
2.4.3 Ordered Crossover (OX) 15
2.4.4 Partially Mapped Crossover (PMX) 16
2.5 Mutation 18
2.6 Termination Condition 19
3. APPLICATION **21**
3.1 Purpose 21
3.2 Termination of the Program 22
3.3 Parameters 22
3.4 Program 26
4. NUMERICAL ANALYSIS **29**

5. CONCLUSION AND RECOMMENDATIONS 31
5.1 Conclusion 31
5.2 Recommendations 31
REFERENCES 33
CURRICULUM VITAE 35

ABBREVIATIONS

GA	**:** Genetic Algorithm
POP	**:** Population
GP	**:** Genetic Programming
GAP	**:** Gate Assignment Problem
FV	**:** Fitness Value
CO	**:** Crossing-Over
PMX	**:** Partially Mapped Crossover
OX	**:** Ordered Crossover
TSP	**:** Traveling Salesman Problem
GUI	**:** Graphic User Interface

LIST OF TABLES

Page

Table 2.1 : Binary Code For Individuals 9
Table 2.2 : Real-Valued Code For Individuals 9
Table 2.3 : Permutation Code For Individuals 10
Table 2.4 : Fitness Proportional Selection For Individuals 10
Table 2.5 : Ranking Method Selection For Individuals..... 12
Table 4.1 : Stagnant Step Optimization..... 29
Table 4.2 : Mutation Rate Optimization..... 30
Table 4.3 : Elitism Number Optimization 30

LIST OF FIGURES

Page

Figure 2.1 : Proportional Selection 11
Figure 2.2 : Tournament Selection 11
Figure 2.3 : Ranking Selection 12
Figure 2.4 : CO Tree (Umbarkar and Sheth, 2015) [1] 14
Figure 2.5 : CO with binary code 15
Figure 2.6 : Ordered CO 16
Figure 2.7 : Ordered CO with null value added 16
Figure 2.8 : PMX 17
Figure 3.1 : Data Input Screen 25
Figure 3.2 : Run Algorithm Screen 25
Figure 3.3 : Algorithm Diagram 27

Aircraft Parking Optimization
Using
Genetic Algorithm

SUMMARY

The growth of aviation industry created new problems. One of these problems is the GAP (Gate Assignment Problem). Increasing aircraft traffic at large airports requires the effective use of gates. Airplane parking positions have affected the operational costs as well as the ability of transit passengers and their baggage to reach the next flight. The goal of GAP is to park aircrafts at the appropriate locations on the ground to ensure that airport parking area is used in a most efficient way. In literature, first the total walking distance of transit passengers, then the distance from the parking position to the baggage area and the total distance to the next flights of transit passengers are used as the target function.

In 1980s, Babic et al. [2] did some research on this problem by using integer-programming and it got investigated by many others. In the following years, statistical models were preferred instead of deterministic models. Various simulation models were proposed by Yu Cheng [3]. Bolat [4] took operational costs as a target function instead of transit passengers or baggage distance. For the first time in 1998, GA [5] was used for GAP. The calculation of this complicated problem (GAP) is shortened using GA. In 2001, multiple targeted problems have been proposed [6]. Linkage and uniform gene exchange methods were used for GAP by Paolo [7].

In this thesis, GAP is solved by GA method. Total transit passenger walking distance is used as an objective function. This method aims to minimize airline cost and maximize customer satisfaction. This will also minimize the time delay and the number of delayed planes. The GA has been developed properly for GAP without any use of GA program package, and it has been introduced in the interface of a web page 'http://genetik1.netf13.com/home'.

This paper aims the optimization of planes allocation in order to minimize the operation cost, especially in hub airports. Gate assignment problem (GAP) is a process which affects hub airports scheduling if it is not handling with a great attention. Even there are similar studies, this paper's application runs on an interactive and open to public web page. In this thesis, it is emphasized that GA gets inspiration from natural selection. These type of algorithms are successful in problem-solving because they use the nature's rules of survival. A genetic algorithm is an evolutionary algorithm. So there must be appropriate parameters of the algorithm to work properly. These parameters are chosen in accordance with GAP and the size of the problem.

While coding the program, a modular structure was constructed. This program can be developed easily because changes in a specific class are only effective within its class. As a result of this, changes do not affect the general working protocol.

Enhanced Permutation Code is appropriate for this study because normal permutation code is insufficient for null values in representation. OX and Partially Mapped CO (PMX) are appropriate for permutation representation but OX is better for permutation with null values.

In this application, PHP is used for the algorithm, HTML is used for GUI and MySQL4 is used for the database. As this program is a GUI, it allows other people to go to the web-site and interact. While coding the programme, a modular structure was constructed. The user can choose any parameter and start the program. However, by using their preferred parameters, they can get top 5 results, analyze and examine them. $'data\_set\_gate'$ table is a square matrix that indicates the distance between the gates. $'data\_set\_plane'$ table is a square matrix that indicates the number of transit passengers on a plane. Furthermore, 'open' table is for the easy use of the program with memorizing and accessing usable parameter sets. In the end, parameters and results are kept in the "stat" table. On the background, communication with the database, the start of the algorithm and parameter checks and setup is provided from PHP. This object oriented PHP system is coded by Burak Durukan and algorithm is designed by Burak Güler. The program and the code in the thesis is stored in a CD. Running parameters and results can be seen 3.1 and 3.2 sections.

During the trials, when the final step is used to get a result, the outcome turns out to be random. The algorithm does not give palpable results in the final step. The algorithm should work like an artificial intelligent so stagnant step is preferable parameter for it. Using stagnant step gives a chance for algorithm to find a solution before the final step which is determined earlier but in big numbers. The stagnant step forces algorithm to stop after getting the best result. This study shows that the best stagnant step is between 1/5 and 1/9 of the final step.
Another usable parameter that affects run time of the algorithm is mutation rate. Studies show that the best mutation rate for this algorithm is between 0,06 and 0,11. Due to direct changes in solutions, higher mutation rate is unnecessary and negative for the result. Another parameter behind the best result is elitism.
There have been similar studies about GAP before. However, what makes this application unique is that it can be run on a web-page and is open to public. This program is open to further development as changes in a class are only effective within its class. Changes do not affect the general working protocol.

Genetik Algoritma
Kullanarak
Uçak Park Yeri Optimizasyonu

ÖZET

Havacılığın büyümesi bir çok yeni probleme neden oldu. Bu problemlerden bir tanesi de GAP (Kapı Atama Problemi [Gate Assignment Problem]) dir. Uçak sayısındaki artışın sebep olduğu yoğunluk havalimanlarının verimli şekilde kullanılmasını zorunlu kılmaktadır. Uçakların park pozisyonları operasyonel maliyetleri etkilediği gibi transit yolcuların ve yolculara ait bagajların bir sonraki uçağa yetişebilmesini de etkilemektedir. GAP in amacı uçakları en uygun şekilde park ettirerek havalimanının verimli kullanılmasını sağlamaktır. Literatürde, ilk olarak toplam yürüme mesafesi hedef fonksiyon olarak, ardından toplam yürüme mesafesi ve bagajların bir sonraki uçuşa olan uzaklıkları hedef fonksiyon olarak kullanıldı.

GAP, Babic ve diğerleri [2] tarafından 1980'lerde tam-sayılı program kullanılarak araştırılmaya başlandı ve bu konuda yapılan araştırmalar sürdü İlerleyen yıllarda, deterministik model yerine istatistiksel modeller tercih edildi. Çeşitli simulasyon modelleri Yu Cheng tarafından problemin çözümü olarak önerildi [3]. Bolat [4] ise hedef fonksiyonunu transit yolcu ya da bagaj uzaklığı almayıp operasyonel maliyetleri hedef fonksiyonu seçmiştir. İlk defa 1998'de GA (Genetik Algoritma [Genetic Algorithm]) , GAP için kullanıldı [5]. Bu karmaşık problemin hesaplanması GA ile kısaltılmıştır. 2001 yılında çoklu hedef fonksiyonu önerildi [6]. Bağlantı (Linkage) ve düzenli dağılım metodu GAP için Paolo [7] tarafından kullanıldı.

Bu tezde, GAP, GA yöntemi kullanılarak çözüldü. Toplam transit yolcu yürüme mesafesi hedef fonksiyon olarak seçildi. Bu metod havacılık maliyetlerini düşürmeyi ve müşteri memnuniyetini yükseltmeyi hedeflemektedir. Bu da gecikmeleri ve geciken uçak sayısını azaltmaktadır. GA uygulaması herhangi bir hazır program kullanmadan GAP için, web arayüzü kullanılarak "http://genetik1.netf13.com/home" üzerinde geliştirildi.

Bu tezde özellikle aktarma merkezi olan havalimanları için operasyon maliyetlerini minimize edecek şekilde uçak yerleştirme optimizasyonu yapıldı. GAP'in uçak yerleştirme süreci yeterince dikkatli takip edilmezse havalimanlarının programları etkilenebilmektedir. Benzer çalışmalar daha önce yapılmış olsa da bu çalışma uygulama web arayüzü ile etkileşimli bir programdır. Bu çalışmada, GA'nın doğadan ilham aldığı vurgulanmaktadır. Bu tarz programlar doğanın doğal seçim kurallarından etkilenerek çalışırlar. Genetik algoritma evrimsel bir algoritma olup uygun parametreler ile çalışmaktadır. Bu parametreler GAP'e uygun ve problemin veri hacmine göre seçilmiştir.

Geliştirilmiş Permütasyon kodlama bu çalışma için uygundur çünkü normal permütasyon kodlama boş değerlerin gösteriminde yetersiz kalmaktadır. OX (Düzenli parça değişimi [Ordered Crossing-over]) ve PMX (kısmi eşleştirmeli parça değişimi

[Partially Mapped Crossing-over]) permütasyon kodlama için uygundur fakat OX tekniği boş değerler içeren permütasyon kodlama için daha iyidir.

Bu çalışmada PHP programı algoritma için, HTML ise GUI (kullanıcı arayüzü [Graphic User Interface]) için ve MySQL4 ise veritabanı için kullanılmıştır. Bu programda GUI olduğundan kullanıcılar web sitesini ekileşimli şekilde kullanabilir. Kullacı istediği parametreleri kullanarak, en iyi 5 çözüme ulaşabilir, bu çözumleri analiz edebilir ve inceleyebilir.

Tablo, $'data\_set\_gate'$ bir kare matris olup kapılar arasındaki uzaklığı göstermektedir. $'data\_set\_plane'$ matrisi kare matris olup uçaklar arasındaki transit yolcu sayısını göstermektedir. Buna ek olarak "açık" tablo programın kolay kullanılması ve parametrelerin hafızada tutulması ve küme olarak kullanılabilmesi içindir. Son olarak, parametreler ve sonuçlar "stat" tablosunda tutulmaktadır. Program arka planda veritabanı ile haberleşmekte, algoritmayı çalıştırmaya başlamadan parametreleri kontrol etmekte ve son olarak PHP ile çalıştırılmaktadır. PHP kodları Burak Durukan tarafından, algoritma tasarımı ise Burak Güler tarafından yapılmıştır.

Program testlerinde, en son adım metodu sonuç bulmak için kullanıldığında sonuçlar büyük bir rassallık göstermektedir. Algoritma çok net sonuçlar vermemektedir. Algoritma'nın yapay zeka gibi çalışıyor olması için durgun adım (stagnant step) parametresi kullamalıdır. Durgun adım parametresi kullanıldığında algoritma, çok büyük adımlara gelmeden de sonuç bulabilmektedir. Durgun adımlar parametresi ile en iyi sonuçları erken adımlarda bulabiliyoruz. Bu çalışma durgun adım parametresinin, final adım sayısının 1/5 ile 1/9 arasında olduğunu gösterdi. Diğer sonuç süresini etkileyen, kullanılabilir parametre ise mutasyon oranıdır. En iyi mutasyon oranı ise 0,06 ile 0,11 arasındadır. Daha yüksek mutasyon oranı sonuca olumsuz yansır ve çözüm süresini uzatır. Sonucun iyileşmesini sağlayan bir diğer parametre ise elitizmdir.

Sonuç olarak benzer çalışmalar yapılmış olsa da bu çalışma halka açık ve web sitesi üzerinde yürütülebilmektedir. Bu program kolayca geliştirilebilirdir çünkü bir sınıfta yapılan değişik sadece ilgili kısmı etkilemekte, programın geri kalanını etkilememektedir.

1. INTRODUCTION

The importance of aviation started with World War II, where the growth of aviation industry has been astonishing. From those days nowadays, there has been a great increase in the number of aircrafts, and airports. This growth leads to new problems as flight routes, slots, capacity of airports, distance of gates etc..

Recently, one of the biggest issues of airports is minimizing walking distance of transit passengers. All over the world, there are millions of transit passengers who are traveling by plane. They use at least two airports to reach their final destination. Our main purpose is to solve transit passengers problem. The problem is; they have to walk a lot to reach their transit flights in an airport facility. In this project, we will use genetic algorithm and try to minimize walking distance.

In an airport, most of the operational process can be controlled by the help of technology. There are so many airway companies which provide services. They can control every process except walking distances of transit passengers. In huge airports like Istanbul Atatürk, it is one of the biggest issues. For instance, companies can not control the gates which are given by the ramp tower. It is important because when an aircraft landed on ground, operation starts. From that moment, they have to deal with fueling, catering, and of course with passengers. Last one is the most random of all. They have a plane of passengers including transfer ones. They have to think about their customers best interests. Because of this, minimizing the distance between incoming flight and outgoing flight is very crucial. To solve this operational problem, a genetical algorithm can be used.

A genetical algorithm is; the way to solve problems by using natural selection methods. In an algorithm like this, people try to find the best solution in a complex problem. It uses the way of biological evolution. The algorithm asks the same question over and over and it only stops when it finds the one of best answer of all or the answer step which is predicted. Choosing process works by using fitness values. High rated options have better chances to be chose. The ones which are not selected disappears.

There are so many methods to find best answers. Roulette Wheel Selection Method, Tournament Selection Method and Elitism are the most common ones. We will give more information about these methods in section 3.

Organization of this thesis is as follows: introduction section is devoted to general information about aviation and the importance of GAP in the hub airports. The motivation of the thesis and literature review about GAP are given in the same section. GA's process is explained and a literature review is given in the second section.
In section three, all the steps regarding the application of GA to the GAP are given. All the parameters used for this application are explained one by one and their effects to the solution are emphasized. The programmes used for the design of the web page and the algorithm are explained in the same section. The resulting values obtained after several trials and runs are given in Numerical Analysis Section. The last section is dedicated to the conclusion, comments, and recommendations. In order to make the algorithm better and to be applicable at an airport, the parts that need to be improved are highlighted and the good and bad sides are mentioned.

1.1 Motivation of Thesis

In this thesis, we investigate the effect of aircraft parking position on transit passenger walking distance. Airline companies may take precautions, make developments or rectifications for their inner process, malfunctions and unpredictable events. However there is no way for an airline company to take precautions for delay of transit passengers.
Although, architecture of airports consider walking distance of transit passenger, still it is an important process which need to be managed. Delays may cause penalty fee for the airline company.

1.2 Literature Review

Genetic Algorithms (GAs) were found around middle 60's but before Holland's book "Adaptation in Natural and Artificial Systems", GAs were not accepted for solving problems. GA were seen as theoretical algorithms such that not applicable to any kind of problem. Holland's work gathered in this book, explaining biological fundamental of the evolution and gives theoretical ideas about adaptation and crossing over and it is

suggesting that using chromosomes, genes as a way of bits of computer codes so this method is able to be used as an algorithm [8].

In those years, computers calculation capacity was not enough to use such an algorithm and more important parallel programming was introduced later. Aircraft Gate assignment problem has been researching from 1980 to nowadays. This problem stated mathematically and solutions were proposed by linear 0 - 1 Integer Programming. Transfer passengers were not evaluated and scalability of a solution was also another concern. Branch and bound technique (Backtracking version) was proposed in 1983 by Babic and et al for this purpose [2].

In 1985, a paper similar to previous work aside from transfer passenger consideration is published. Linear programming and heuristic programming are used to optimize total walking distance, where the greedy heuristic solution can be used as an initial solution and deterministic program can be used afterward. Thus, total run time can be shortened [9].

Chang and Schonfeld, their approach was using flight sequencing in an airport. Their motivation was "sequencing bigger aircraft last in and first out" (BLIFO) or "smaller aircraft first in and first out" (SFIFO) the key factor of the efficient gate assignment problem. Choosing one of the methods by looking at the density and size of the airport can be found out the optimal solution [10]. Haghani and Chen made improvements on Haghauni's heuristic solution by adding time parameter to the solution of the GAP [11].

Simulation and simulation models are also proposed. Yan S. And Chang C.M.'s model is a multi-commodity network-flow model which is based on Lagrangian relaxation. In 1998, a simulation model for GAP is suggested by Cheng Y. And also in this year a rule-based reactive model for the simulation is also published. Rule-based reactive model is more applicable to object-oriented languages. [3] [12] [13]

Bolat's Gate assignment approach was different from rest of the work because the ground operation is rarely objective of the gate assignment problem but in his paper objective is the cost of the ground operation [4].

Gu and Chung used GAs for a delayed flight for a limited time period neglecting time parameter. Their formulation is for solving gate manager's problem for delayed flights assignment in a short time [5].

In Yan's and Huo's paper multiple objective models - which are the waiting time and the passenger walking distance - is used for a Chiang Kai-Shek Airport. The model multi-objective and zero-one integer program by using the simplex method and the branch-and-bound technique and also the column generation approach for large-scale problems [6].

For congested time and the number flights exceed the number of gates, scheduling and optimization is the topic for the paper of Ding, Lim, Rodrigues, and Zhu. In this optimization, greedy algorithm, tabu search heuristic and tabu short-term memory methods are investigated [14].

Genetic algorithm suggested for the GAP is also developed in algorithms. Their papers propose a novel matrix representation of the GAP. This matrix leads GA to uniform crossover and linkage. That is to say, decrease the chance of infeasible solution after CO. [7] [15]

General happiness of the passengers are defined by three metrics by Clarke and Maruoli which are passenger transit time, aircraft taxi time and robustness of the gate assignments. If these metrics would be balanced the airport performance can be accepted as good. [16].

2. Genetic Algorithm

Genetic algorithms aim to choose the most suitable solution way to a problem. While doing this process,the genetic algorithm follows up the natural selection which decreases poor genes by extracting them and pick good genes to produce more. A genetic algorithm does not scan all possible solutions for a problem, it provides a faster way to get the solution as it recognizes the specific part of the solution. That is why genetic algorithms outshine other traditional methods which are insufficient for problems of today's complex world and expending a lot of time for a solution.

First, ever genetic algorithm approach was in the early 70s. However, actual evolution of genetic algorithm was after John Holland's machine proposition which was the successful transmission of the evolutionary process to a computer.

Holland gave information about how genetic algorithm process works in his work that was published in 1975 called "Adaptation in Natural and Artificial Systems". In this work, Holland took genetic algorithm as an abstract model of evolution. Furthermore, he aimed to copy the evolution model's one and only mechanic structure that contains learning, reproducing, changing and adapting ability to computers, in order to get new and better products from solutions.

Holland built his genetic algorithm by sequencing "1"s and "0"s, simply using "bit"s on a computer to substitute natural selection's crossing over, mutation and inversion. According to this method, each chromosome in other words bit, composed of specific alelles and selection operator would supervise which chromosome is going to pass to further generations. Crossing over is the part exchange of the different chromosomes. Mutation is the random changes in some alelles' of a chromosome. Tough this method was only used in theoretical works till the early 80s, Hollstien and Jong's studies carried genetic algorithm to the more utilizable method in many fields. In 1989 a student of Holland, Goldberg, published a book named "Genetic algorithms in Search Optimization and Machine Learning" and carry the value of genetic algorithm in

international academy by using the genetic algorithm on gas pipeline supervision and unearth possible work fields with the genetic algorithm. [17]

In 1992, John R. Koza used the genetic algorithm on genetic programming his studies and in the following process perspective on genetic algorithm has changed in a positive way. Moreover, performance process of genetic algorithm improved in following studies which led genetic algorithm to have more work area. John R. Koza, Genetic Programming, On the Programming of the computers by means of natural means of the natural selection

In 2014, Bouras and his associates worked on GAP in details. They made a different approach to GAP by looking through all possible algorithms that can solve the problem such as heuristic and meta-heuristic algorithms.Their paper solutes the GAP in deterministic ways. [18] Nowadays genetic algorithm actively used in areas such as

- Finance and marketing
- Auto-programming and information systems
- Mechanic learning
- Optimization problems
- Math problems
- Social systems
- Game programming
- Genetic of the population. [19]

Purpose of an algorithm in a problem's solution is the best and the fastest way to solve the problem. If the problem is a polynomial equation, the solution can be found in short time with ease. However, if the problem is a non-deterministic polynomial equation, solutions may not give the results or solution will take longer time than expected. To get a solution for a non-deterministic polynomial equation, a researcher can use GA to get the closest result as it works with intuitive way.

GAs differ from traditional search techniques in several ways ;

- First, GAs optimize the trade-off between exploring new points in the search space and exploiting the information discovered thus far.
- Second, GAs have the property of implicit parallelism. Implicit parallelism means that the GA's effect is equivalent to an extensive search of hyperplanes of the given space, without directly testing all of the hyperplane values.

- Third, GAs are randomized algorithms, in that they use operators whose results are governed by probability. The results for such operations are based on the value of a random number.
- Fourth, Gas operate on several solutions simultaneously, gathering information from current search points to direct subsequent search. Their ability to maintain multiple solutions concurrently makes Gas less suspectable to the problem of local maxima and noise. [20]

2.1 Fundamental of the GA

In this section, we investigate theory underneath the GA. Holland explained in his book, how GA search the highest fitness valued individuals by using Schema Theory. Schemata (plural of schema) is like subseries in algebra. A subset of the solution set is called schema. For example; a schema can be defined by using the element of $\{a, b, *\}$ and $*$ is able stands for a or b.
$X = [*aab]$
$'*'$ asterisk can define a or b so $X1 = [aaab]$ and $X1 = [baab]$. For this H set, there is only two possible solutions because solution number is a number of set values over the number of the asterisk in set. We need to make two definitions.

Order of the schemata O(X) : Constant elements of the string, in our example, 3. Length of the schemata L(X) : Distance between the first constant element of the string and last constant elements of the string. Building Blocks : Low order and shortly defined schemata are called Buildings blocks.

Hypothesis 1 : *(Building Block Hypothesis) A genetic algorithm seeks near-optimal performance through the juxtaposition of short, low-order, high-performance schemata, called building block.* [21]

Fundamental GA theorem : *Schemata Theorem: Short, low-order, above-average schemata receive exponentially increasing trials in subsequent generations of a genetic algorithm.* [21]

That is to say after some recursive computation solutions are better. The Algorithm straightens solution generation by generation. In each generation, the algorithm tries to find an optimal solution by the information taken from an ancestor of the generation,

therefore, GA does not need to scan all of the solution universe. In brief, shorter scan times and more chance to find solution differ from the random algorithm.

2.2 Main Factor of Genetic Algorithm

Two main factors of the GA are problem encoding and the evaluation function. The problem encoding means the representation of the problem. [22]
Usually, binary codes are used to represent a solution. Coding from normal solution to binary codes and reverse, encoding from binary codes are maintained in the algorithm. In this binary, representation, sometimes unused values may occur. For example, if our solution universe made consist of 2000 elements, our binary code must be at least 2^{10} which is equal to 2048 so at least 48 elements are same or infeasible.
Using symbol or numbers to represent solution is another way to represent solutions. In this way, the algorithm does not have empty values but it is hard to decode solution strings considering binary coding. In our paper, we used symbols provided that it is a unique solution if the way of approaching the problem, our coding is able to be useless.

The evaluation function is for understanding how far to the optimal solution. The evaluation function is also called the fitness function (we used as Fitness Function).The fitness function is also for if the algorithm works fine or not. We do not know the optimal solution in GA and also we would not be sure for where to stop searching the Solution set. Consequently, Fitness function can be a good indicator resolve the stop point of the algorithm.

Two components of the GA is usually enough for the solution of any problem. First is the representation of the solution, other is for determining how well our solution. Other components are more static for any GA problems called Genetic Operators.

2.2.1 Representation of the Solution

2.2.1.1 Binary Code

Hollands works mostly uses binary coding so this is the first representation of the GA codings. In this type of coding, "1" and "0"s are used. In this coding, there is not any additional mutation and CO operator optimization. Because this coding hide the real value in it.

Table 2.1 : Binary Code For Individuals

Individual 1	101101010101000000000
Individual 2	110101110001010110101

2.2.1.2 Real-Valued Code

In some problem, real-valued coding of the solution is more appropriate than binary coding. There is not any precise decision about performance of the real-valued or binary code. In the real-valued coding system, coding and encoding of the individuals may be required, this can increase the number of the calculation but on the other hand, this process can decrease feasibility calculations.

Table 2.2 : Real-Valued Code For Individuals

Individual 1	AB12CD34X45AA4
Individual 2	12332221834
Individual 4	UP DOWN UP UP UP DOWN

2.2.1.3 Permutation Code

Job Scheduling and Traveling Salesman Problem is appropriate for permutation coding. This is because every individual shows an array. In Traveling Salesman Problem, salesman has to visit every town once so in our problem, we assume that every gate has been assigned to only one airplane. difficulty in this coding is CO operator optimization because when CO happens some elements occurs twice in the solution that makes solution infeasible. For avoiding this doubling problem there are different kinds of CO method.

Table 2.3 : Permutation Code For Individuals

Individual 1	1	5	8	9	2	11	10	3	6	4	7
Individual 2	9	2	1	3	4	6	11	5	8	10	7

2.3 Genetic Operators

Genetic Algorithms, use GA operators to reproduce chromosome and maintain the diversity of the population. These processes are : Selection, Crossover, mutation.

2.3.1 Selection Operator

Reproduction Operator: This operator is also called selection operator. The solution called individual is copied for their the fitness values. As we mentioned, fundamental theorem of GA states that in every population individuals must be selected through fitness based process. Paired or matching individuals give better results. Linkage and uniform crossover techniques study for selecting matching pair with various techniques.

2.3.1.1 Fitness Proportional Selection

Fitness Proportional Selection (Rulet-Wheel Method): In this method, individuals are marked on the wheel proportional to their fitness like in rulet wheel. Thereby fitter individuals have the higher area on the wheel so fitter individuals have higher chance to be picked and breed next generations. In each, turn an individual is chosen for next generation.

Table 2.4 : Fitness Proportional Selection For Individuals

	Fitness Value	Chance Of Selection
Individual 1	100	20.88 %
Individual 2	90	23.20 %
Individual 2	80	26.10 %
Individual 4	70	29.83 %
TOTAL	340	100 %

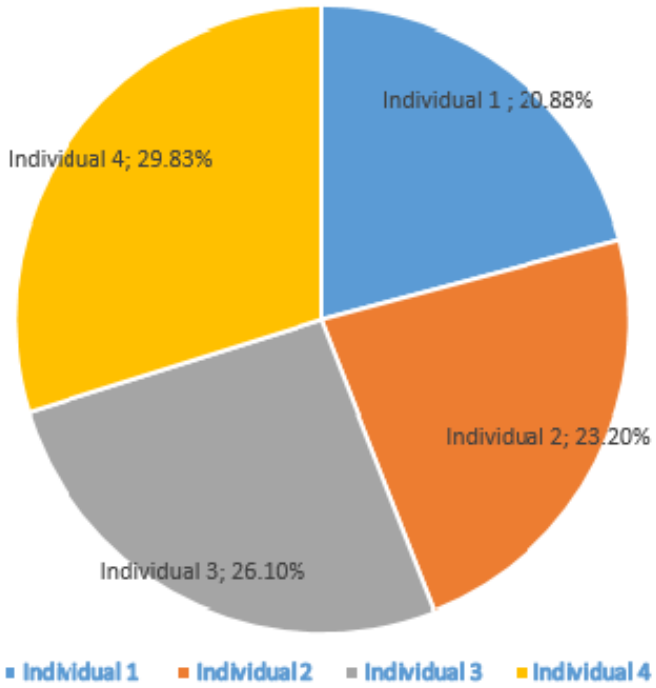

Figure 2.1 : Proportional Selection

2.3.1.2 Tournament Method

Tournament Method: The name comes from tournaments are held for selection next generations. Two random individuals are selected. IF the tournament size increases, (times of comparison between individuals) lower the chance of weak individuals are selected. Tournament selection is easy to code and implement on both parallel computing and normal computing. Also, it is usable noisy fitness (ambiguous function which sometimes can give more or fewer values) function.

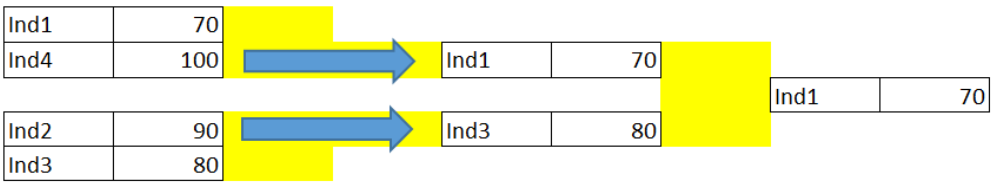

Figure 2.2 : Tournament Selection

2.3.1.3 Steady - State Selection

Steady - State Selection: The idea behind this method is to sort fitness values of the individuals for selection take place in the next generation. In the next generation, individuals have sorted again and bad breeds are removed. For instance, an algorithm has the population size of 15 takes more than 15 individuals for selection. It can depend of the problem, let's say in each generation 30 individuals and eliminate half of the individuals for new generations.

2.3.1.4 Ranking Method

Ranking Method : This method is similar to Rulet Wheel method but instead of proportional distribution, high fitness valued individual makes a surplus number of copy to the population.

Table 2.5 : Ranking Method Selection For Individuals

	Fitness Value	Calculated Column	Chance Of Selection
Individual 1	100	4	12.00 %
Individual 2	90	3	16.00 %
Individual 2	80	2	24.00 %
Individual 4	70	1	48.00 %
TOTAL	340	10	100 %

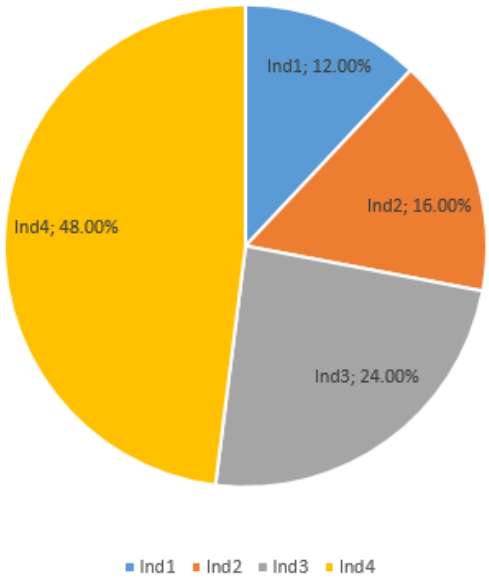

Figure 2.3 : Ranking Selection

2.4 Crossing Over Operator

Crossing Over Operator : It is the main operator to refine individuals for better breeds in the next population. This operator mix to string or solution in various ways. By doing this new population can take good genes from their ancestors. We use many crossing over methods depends on the specific problem and coding/decoding of the individuals.

Crossing-Over is the fundamental application of evolution for all living species. It lets functioning genes to produce in the offsprings that allow further offsprings to have better genes for survival. Therefore, CO effect cannot be underestimated in Genetic algorithms due to its huge impact. In the figure 2.4 , Umbarkar and Sheth are divide CO into three part but it is not very-useful since every binary CO is included in standart CO.

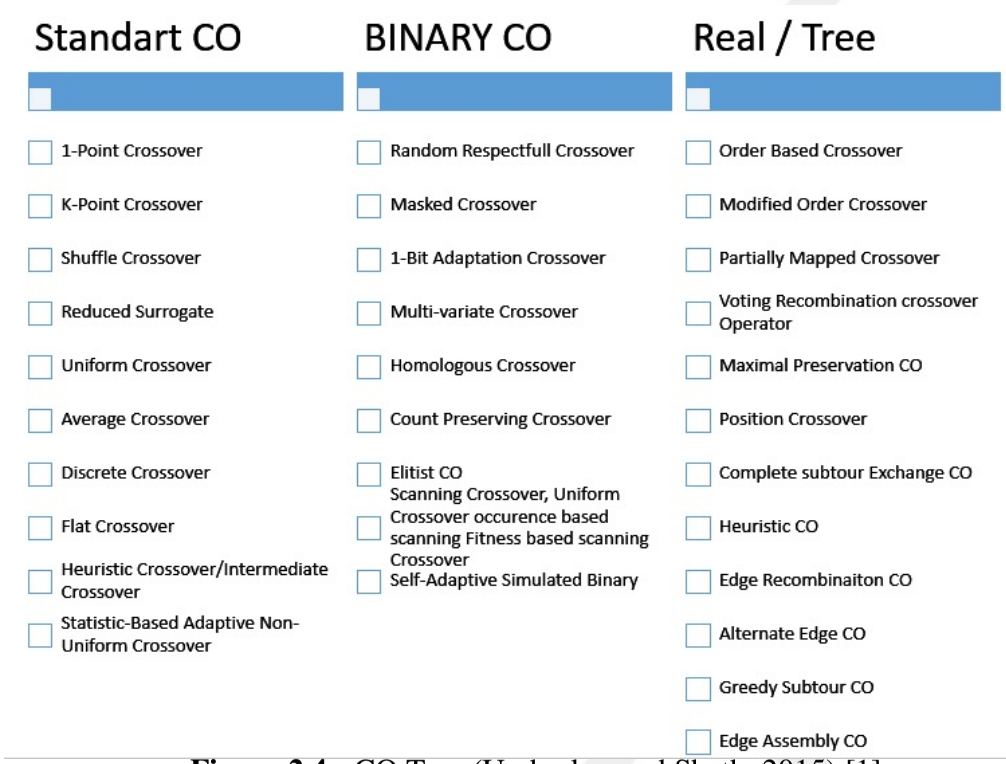

Figure 2.4 : CO Tree (Umbarkar and Sheth, 2015) [1]

2.4.1 1-Point Cross-over

1-Point Cross-over is the most basic method amongst the crossing over methods. Two individuals are split in the same point and they swap their genes between each other. This operator may not be enough for diversity. As well as the cut point can be manual or randomly chosen.

2.4.2 K-point Cross-over

K-point Cross-over, individuals are split over K point then change their part.This method differs from 1-point crossover by taking k number of the cut point. So enough diversity for the next generation can be granted.

- In Uniform Cross-over, two children are generated from two different individuals and they get equal number genes from their ancestors

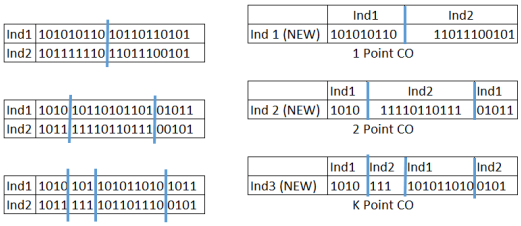

Figure 2.5 : CO with binary code

In binary coding, there is no need extra effort because of the feasibility reason. In our problem, we mostly focus on permutation coding CO techniques. A plane can park only a gate and also a gate is occupied by only a plane. Real-valued coding need to be controlled for feasibility.

2.4.3 Ordered Crossover (OX)

Ordered CO is used for mostly TSP because doubled genes are avoided in this method. In figure 2.6, CO starts with taking some random gene to transmit with each other. In our sample, 2., 3., 4. positioned genes are chosen randomly and marked. In next step, for the Parent1, "8 4 2" are marked then Parent2 order of are found "2 4 8". Order of elements "8 4 2" changed to "2 4 8". For the Parent2, marked genes in the Parent1 "7 6 5" are found after that -same as Child1- order of the elements are changed like the Parent1. Other genes, (non-marked) preserved like their parents.

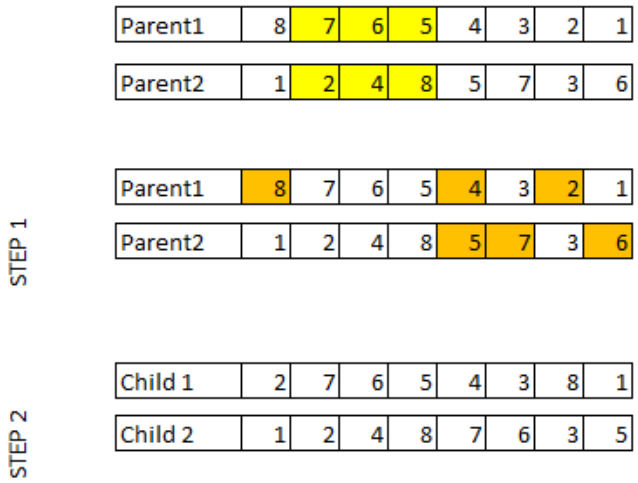

Figure 2.6 : Ordered CO

In our thesis, we used the improved version of the OX. In GAP, some gate may not be used so in this case individuals have "NULL" values in the code.

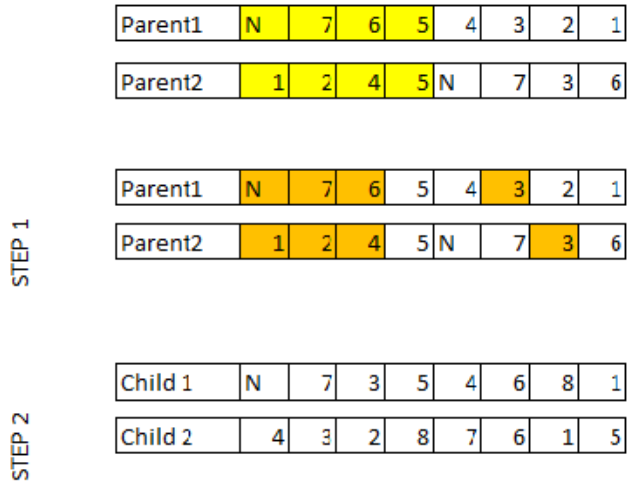

Figure 2.7 : Ordered CO with null value added

2.4.4 Partially Mapped Crossover (PMX)

Partially Mapped CO is also another method for TSP and permutation coded individuals. In figure 2.8, random number of genes are taken place for CO. In this example, 4., 5., and 6. genes are exchanging from Parent1 to Parent2. After this step, some double genes may be seen in the individuals. To avoid doubled genes, every doubled genes are fixed by change a gene in this doubled pair. For example, in first step, "1" and "8" is changed, then "2" and "5" and finally "4" and "5". After there

is not any doubled genes, permutation coded individual is granted so new individuals (children) are ready.

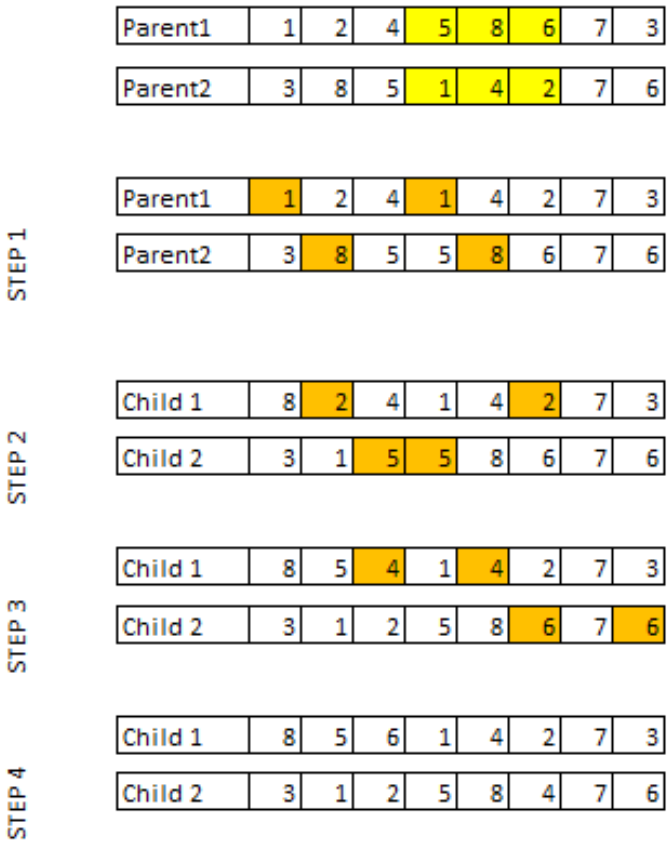

Figure 2.8 : PMX

2.5 Mutation

Mutation Operator: Basically, mutation is the deformation of in the individual to form new individuals. This process is done because of the current population is not enough for global minimum or maximum. Sometimes solutions are restricted to a limited set so changing individuals element results from different fitness value and different combination in the next generation.

1. Bit Flip Mutation / Random resetting : one gene is selected and its value changes. In binary coded individuals, 0 to 1 or inverse but normal coded individuals reset a random gene in the individuals. In the sample below, second gene has mutated by changing its value from 0 to 1.

$Ind1 = [10010]$

$Ind1(new) = [11010]$

2. Swap Mutation : Two random genes are selected and interchange these two genes. In the sample below, the second and the fourth gene has changed its position in the individual.

$Ind1 = [14025]$

$Ind1(new) = [12045]$

3. Scramble Mutation: A set of genes are taken into consideration and genes are swapped their position or inverse their positions. In the sample below, genes has mutated from second position to fourth position and changed their position in the inner set.

$Ind1 = [12345]$

$Ind1(new) = [13425]$

2.6 Termination Condition

GA is not a deterministic algorithm so there is an exact point to stop the calculations. There are some methods to restrict search process.

1. Calculation Time Criteria : After the predefined time passes search process stops and give the best individual for the answer.
2. After n Recursive Step : After n times of generations, search process stops.
3. Optimization Target Criteria: If the target values are known, the process stops when the fitness values reach its target.
4. Minimum Increase in Best Fitness Value: GA works better at the start after some steps increase in fitness values decreases gradually. After a while, decreases reach zero therefore, searching algorithm can stop at this point. Because there would not be any progress.

3. APPLICATION

3.1 Purpose

Nowadays, the GAP is one of the biggest problem in hub airports. The importance of this problem due to customer satisfaction, airlines costs etc. This problem aims to minimize airlines costs and maximize customer satisfaction by minimizing transit passenger walking distance between gates. This will also minimize time delay and the number of delayed planes. In literature, GA is one of the methods applied to GAP. In this thesis have been developed and introduced in the interface of a web page "http://genetik1.netf13.com/home". The aim is to developed open public web page which can be evolved by adding new constraints or new feature. In the 3.1 and 3.2 one can see running parameters and results.

One of the biggest problems is as we mentioned in previous section transit passengers' catch up their planes. GA is used in this problem, we use real-valued code to implement individuals.

Problem: The number of gates and number of transit passengers is given. In GUI, the user can enter a matrix to represent the distances between gates and a number of the transit passenger.

Our approach to GAP is like Traveling Sales Problem. There are two constraints;

- Each gate is assigned by a plane
- Each plane can park at a gate.

In our problem, number of gates and number of passenger are given to minimize

$X_{(i,j)}$	Number of Passenger from Plane-i to Plane-j
$D_{(i,j)}$	distance from ith position to jth position

The Total Passenger Walking Distance Function

Minimize $\sum_{(i,j)} = X_{(i,j)} * D_{(i,j)}$

3.2 Termination of the Program

Termination : the upper limits are needed for termination point, as a result of GA is not a deterministic algorithm that it is not expected to search all of the solution space. So we need some criteria to finish the program.

1. **Time Constraint:** There is a time limit which is 15 min for any bug.
2. **Maximum Step Size:** GA get better fitness value slowly after some step and also it is possible that there may not be an advance in FV. Therefore the best way for the user is to enter a maximum step limit for GA. In our GUI of GA, the user is able to define max number of step
3. **Stagnant Step Size:** Repeated fitness valued steps are a sign of there may not be an advance in FV. In our GUI of GA, the user is able to define max number of step

3.3 Parameters

Individual(chromosome) : We used real-valued coding(string) to represent our solution. Individual is a solution of the problem.

For example an individual;

$IND = [PL2,\ PL3,\ PL4,\ PL5,\ PL1]$

Population : Individual pool is used for the selection, CO and the mutation and called population. Population size affect speed of the solution and convergence of the global minimum/maximum. If the population size is big then crossing over and the mutation has good results after application of the operators but the consequence is the greater time of calculation. If the population size decreases then calculation time decreases but the chance of finding global max/min value. In our GUI of GA, the user is able to determine the number of the population. For further research, we will investigate the affect of a number of POP with respect to time.

For example nth generation of POP

$POP[n] = [IND1,\ IND2,\ IND3,\ IND4,\ IND5,\ IND6,\ IND7]$

Initialization of the population: Initializing is one of the key factors in the success of the algorithm. By cause of a greater number of population is able to inquiry the space of solution. Next Greater initial population is easily restricting place of the extremum points. As in the population size also initialization must be set for the performance of the program. In our GUI of GA, the user is able to determine the multiplier of the initial population size.

Selection: Wheel roulette is mainly used for our GA. User can choose different methods to determine which methods are better or which methods are useful for big sized Individuals or greater population

Crossing over: The main operator of the diversity of the individuals is crossing over. CO method is chosen based on the coding of the individual. For instance, in binary coding, one point, two points or uniform CO can be used but in real-valued coding may not give the best performance. In our problem, order, and replication of the planes are not desired so OX is used. Some improvements are done for specification of the problem. In our GUI of GA, the user is able to choose a way of the number of the population.

Mutation: This operator is essentially for stagnant generations. When generations of the population stay constant in in terms of fitness value mutation can change individuals so this variety advance pool and make the contribution to current and the next generation. The higher rate of mutation induce much new genetic materials then these new genetic materials can change characteristics of the population rapidly. Due to rapid change in structure means loss of higher fitness valued individuals. The lower rate of mutation, stagnation may occur in the search process. In our GUI of GA, the user is able to choose a way of the probability of the mutation.

Random Individual Method: Given individuals consist of random different numbers. Selections are made from alternative good/bad solutions. Feasible method: This feasible method checks out if there is a plane assigned to two gates or a gate with more than one assigned plane. In this study, a plane cannot be assigned to two gates because of representation method. However, this option fall into disuse to make this study improve easily.

Initial Step Size: With this parameter, the algorithm chooses what times of normal population it should get and make a new population. The existence of the extra individual in initial population let algorithm to work in larger solution space that makes

a scan in a larger portion.At beginning larger scan lead algorithm to find local/global extremum points faster. However, if this parameter is high algorithm may work slower.

Fitness Method: In this problem, it takes a product of gates that planes parked and the transit passengers from those planes and walking distance for all transit passengers tried to be minimized.

Crossing Over Rate: Crossing over is the most important parameter for natural selection. Without crossing over, new offsprings cannot continue to transfer new genes to further offsprings. That is why in this algorithm crossing over rate is certain but it determines the probability rate of crossing over to preserve good solutions.

Crossing Over Point Rate: This parameter determines which point algorithm going to cut the solution.

Elitism: In this parameter, the best way for the solution is picked and preserved from mutation and crossing over.

Mutation Rate: This lets algorithm to make sharp changes in solutions and increases chances of diversity. For the best result mutation rate should be between $0.6 - 0.11$ for current algorithm.

Pool Size: This parameter determines the number of data that is going to be in problem. If the pool size is high, duration time is most likely to be high but the solution is more accurate. That is why it should be close to our data.

Stagnant Step: Stagnant step prevents algorithm to work like other deterministic algorithms. It cuts the process if there is a solution before the final step.

Final Step: Last step is determined with this parameter. It put a limit on the algorithm so that algorithm does not work like a deterministic algorithm.

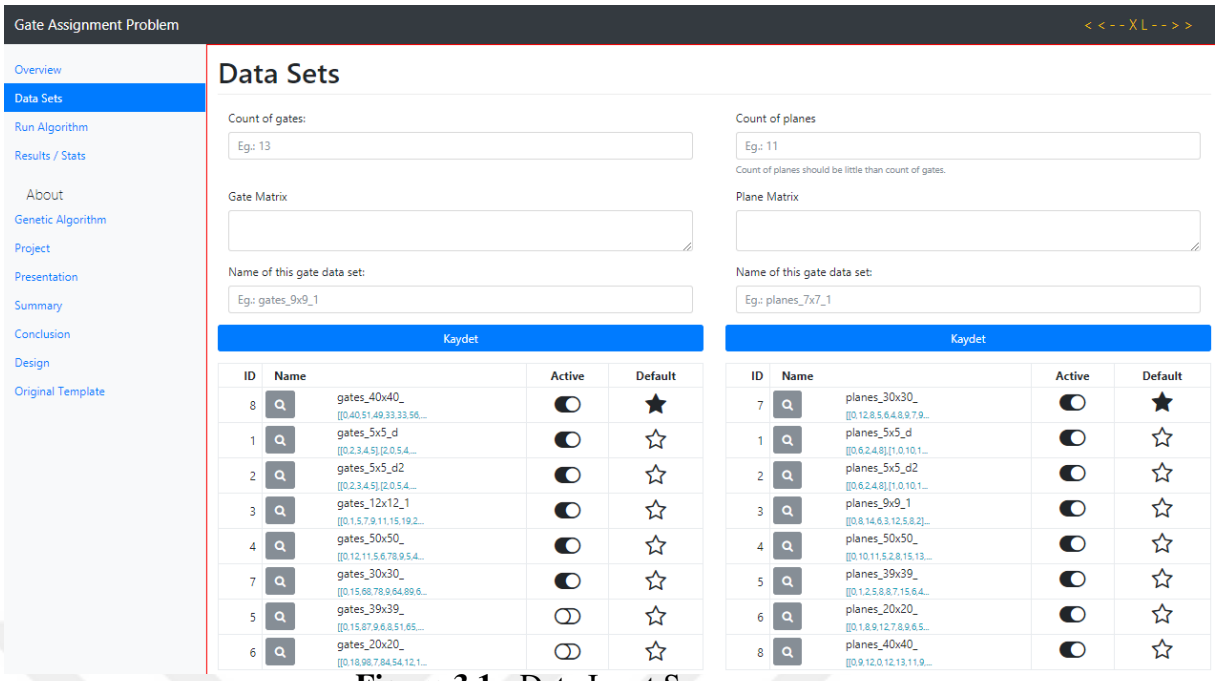

Figure 3.1 : Data Input Screen

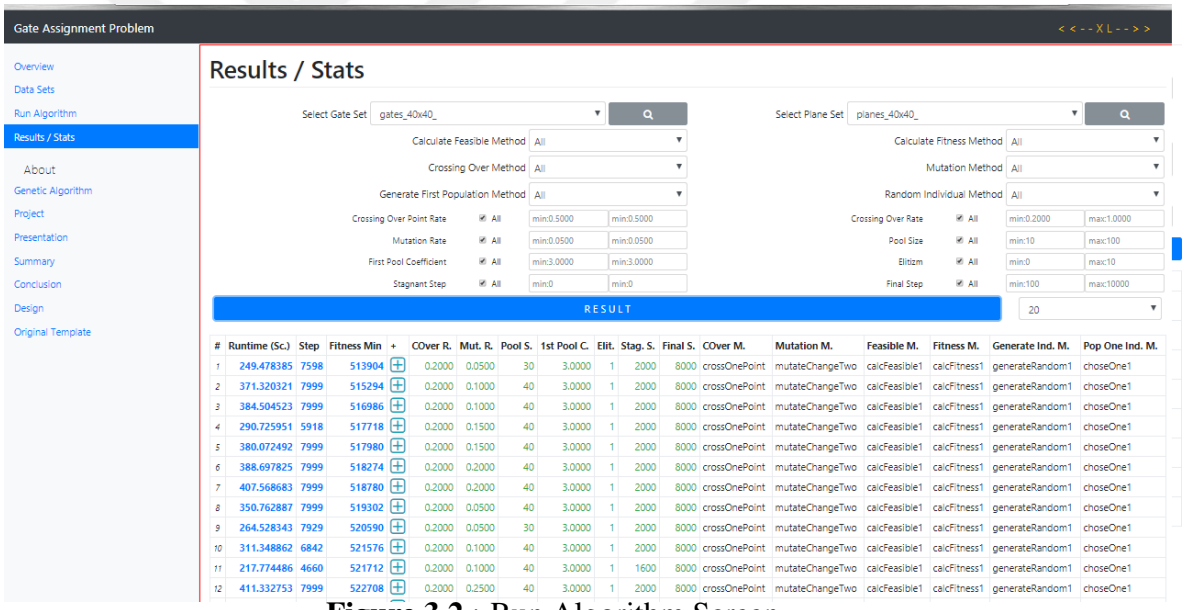

Figure 3.2 : Run Algorithm Screen

3.4 Program

In this thesis, the GA has been developed properly for GAP without any use of package program. This algorithm has been introduced in an interface of a web page which can be evolved and transformed to open public usage. Data sets, parameter sets and study results preserved in MySQL database. $'data\_set\_gate'$ table is a square matrix that indicates the distance between the gates. $'data\_set\_plane'$ table is a square matrix that indicates how many passengers are transit on a plane. Furthermore, 'open' table is for the easy use of the program with memorizing and accessing usable parameter sets. Finally, parameters and results kept in the "stat" table. Data and parameter set savings, stating of algorithm and observation of results were provided by the interface. Website's interface was coded by HTML and Javascript. Used Javascript, CSS libraries and versions: AngularJS(1.6.6), Bootstrap(4.0.0 beta), Font Awesome (4.7.0), jQuery(3.2.1), Sweetalert2(6.9.1) In the background, communication with the database, starting the algorithm and checking of parameters and setup is provided from PHP. PHP system was coded object oriented by Burak Durukan.

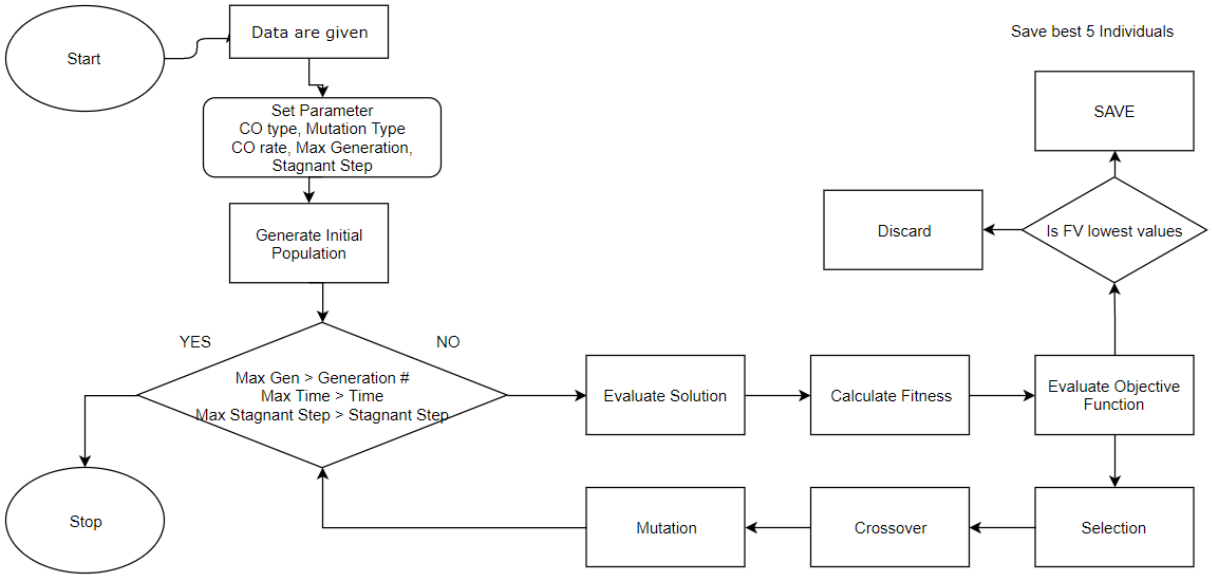

Figure 3.3 : Algorithm Diagram

4. NUMERICAL ANALYSIS

At the beginning of trials, the final step was the limit for the algorithm. However, results show that taking final step lower and not take attention on stagnant step creates a huge randomity in solutions. The algorithm does not give palpable results in final step based results. Because genetic algorithm should be working like an artificial intelligence while getting the result. That's why usage of the stagnant step was the preferable parameter for this algorithm. Using stagnant step creates a chance for the algorithm to find a solution before final step which determined earlier but in big numbers.

The stagnant step force algorithm to stop the solution process after getting the best solution. It repeats this process for five times and gives the best one as the result. Data that has been worked on this study show that best stagnant step is between 1/5 and 1/9 of the final step.

Table 4.1 : Stagnant Step Optimization

Stagnant step	Duration	Fitness Min.
ALL	40,847	272603
1200	87,232	263741
600	39,279	266464
200	14,674	270136
100	9,103	273253
50	4,272	277292
20	0,838	284580
10	0,275	285610
2	109,722	264421
1	108,986	265762

Mutations are changes in the genetic sequence, and they are the second cause of diversity among organisms. These changes happen in different levels, and they can have different consequences. For example, some mutations affect organism badly like mutation may lead a person to have tumor cell by changing the genetic sequence of the significant chromosome that is responsible for the control mechanism of a cell.

Yet, some mutations are beneficial and the only reason for a bacteria to have resistant offsprings. For this algorithm optimization of mutation rate was done by trials and observed data Show that the best mutation rate is between $0.06 - 0.11$ for current algorithm. Due to direct changes in solutions higher mutation rate is unnecessary and negative for the result.

Table 4.2 : Mutation Rate Optimization

Mutation Rate	Duration	Fitness Min.
ALL	133,395	285685
0,25	165,4677	357769
0,20	148,924	318102
0,15	118,943	278553
0,10	88,337	234789
0,05	69,011	209342

Elitism preserve the best way for the solution from mutation and crossing-over. In natural selection some features do not change like all vertebrate have a skull in order to protect the brain.

Table 4.3 : Elitism Number Optimization

Elitism Duration	Fitness Value ALL	51.410.310	265828
1		61.273.001	267077
2		56.557.825	265466
5		42.181.294	265955
10		45.629.119	264816

5. CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This paper aims to make the optimization of planes allocation in order to minimize the operation cost, especially in hub airports. The main purpose is minimizing the total walking distance of the transit passengers. Gate assignment problem(GAP) is a process which affects the hub airports scheduling if it is not handling with a great attention. Even there have been similar studies, this paper's application runs on a web page which is interactive and open to the public. While coding the program, a modular structure was constructed. This program is easily developable because changes in a specific class affect only that class. As a result of this, changes do not affect the general working protocol. In this thesis, it is emphasized that GA gets inspiration from natural selection. This type of algorithms that are successful in problem-solving with using nature's survival rules. A genetic algorithm is an evolutionary algorithm so there must be appropriate parameters to the algorithm to work properly. Parameters are chosen relevant to GAP and occasionally size of the problem. In this application PHP is used for the algorithm, HTML is used for GUI and MySQL4 is used for the database. As this program is a GUI, it let other people get on the web-site and become interactive. The user can choose any parameter and start the program however he/she like and get 5 of the best results. She/he can also analyze and observe the data from the web-site after trials.

5.2 Recommendations

This study shows that GA is feasible for GAP. However, every airport has its own restrictions. For example, two big planes can land in the same hour etc. In this study, airport and restrictions due to special occasions are out of subject so there are no studies on airports and airport's special restrictions. This algorithm was constructed on molecular structure so that it is possible to add tiny extensions.

In literature, GA does not work with the time dimension. Gate-Tail model is suggested. In this study, the time dimension is not used but left open for further studies. One can develop this algorithm adding the time dimension.

REFERENCES

[1] **Umbarkar, A. and Sheth, P.** (2015). Crossover Operators In Genetic Algorithms: A Review, *ICTACT Journal on Soft Computing*, *6*(1).

[2] **Babic, O.**, **Teodorovic, D. and Tosic, V.** (1984). Aircraft Stand Assignment to Minimize Walking, *Airport Transport Division*, 55 : 66.

[3] **Cheng, Y.** (1998). A Rule-Based Reactive Model For The Simulation of Aircraft on Airport Gates, *Knowledge-Based Systems*, *10*, 225:236.

[4] **Bolat, A.** (1998). Procedures for Providing Robust Gate Assignments For Arriving Aircrafts, *European Journal Of Operation Research*, *120*, 63 : 80.

[5] **Gu, Y. and Chung, A.C.** (1999). Genetic Algorithm Approach To Aircraft Gate Reassignment Problem, *Journal of Transportation Engineering*, *125*, 384 : 389.

[6] **Yan, H. and Huo, C.M.** (2001). Optimization of Multiple Objective Gate Assignment, *Transportation Research Part A*, *35*, 413 : 432.

[7] **Paolo, D.E. and Hu, X.B.** (2008). Genetic Algorithms For The Airport Gate Assignment: Linkage, Representation and Uniform Crossover, *Springer*, *157*, 361 : 367.

[8] **Holland, J.** (1975). *Adaptation in Natural and Artificial systems.*, The University of Michigan Press, New York.

[9] **Mangoubi, R. and Mathaisel, D.** (1985). Optimizing Gate Assignment At Airport Terminals, *Transportation Science*, 173 : 185.

[10] **Chang, C. and Schonfeld, P.** (1995). Flight Sequencing in Airport Hub Operations, *Transportation Research Record*.

[11] **Haghani, A. and Chen, M.** (1997). Optimizing Gate Assignment At Airport Terminals, *Transportation Research*, *32*(6), 437 : 454.

[12] **Cheng, Y.** (1998). Network-Based Simulation of Aircraft At Gates In Airport Terminals, *Journal of Transportation Engineering*, *124*(2), 188 : 196.

[13] **Cheng, Y.** (1998). A Rule-Based Reactive Model For The Simulation of Aircraft At Gates, *Elsevier*, *10*, 225 : 236.

[14] **Ding, H.**, **Lim, A.**, **Rodrigues, B. and Zhu, Y.** (2004). Aircraft and Gate Scheduling Optimization at Airport, *Proceedings of the 37th Hawaii International Conference on System Science*, IEEE, Hawaii.

[15] **Paolo, D.E. and Hu, X.B.** (2009). An Efficient Genetic Algorithm With Uniform Crossover For the Multi-Objective Airport GAP, *Springer*, *171*, 71 : 89.

[16] **Kim, H.S.**, **Feron, E. and Clarke, J.P.** (2013). Airport Gate Scheduling for Passenger, Aircraft and Operations, *Tenth USA/Europe Air Traffic Management Research and Development Seminar*.

[17] **Golberg, J.H.** (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Pearson Education.

[18] **Bouras, A.**, **Ghaleb, A.M.**, **Suryahatmaja, U. and Selam, A.M.** (2014). The Airport Gate Assignment Problem: Survey, *The Scientific World Journal*, *2014*.

[19] **Koza, J.R.** (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press.

[20] **Buckles, B.P. and Petry, F.E.** (1994). An overview of genetic algorithms and their applications, *IEEE Computer Society Press, Piscataway*, *171*, 1 : 4.

[21] **Michalewicz, Z.** (1996). *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Science & Business Media, 3. edition.

[22] **Whitley, D.** (1994). A genetic algorithm tutorial, *Statistics and computing*, *4*(2), 65 : 85.

CURRICULUM VITAE

Name Surname: Burak GÜLER

Place and Date of Birth: ÇANAKKALE 04.05.1988

E-Mail: burak.guler.itu@gmail.com

EDUCATION:

- **B.Sc.:** Istanbul Technical University, Mathematical Engineering, Mathematical Engineering Programme
- **M.Sc.:** Istanbul Technical University, Mathematical Engineering, Science, Engineering and Technology