Multi Objective Micro Genetic Algorithm for Combine and Reroute Problem

Soottipoom Yaowiwat, Manoj Lohatepanont, and Proadpran Punyabukkana

Abstract—Several approaches such as linear programming, network modeling, greedy heuristic and decision support system are well-known approaches in solving irregular airline operation problem. This paper presents an alternative approach based on Multi Objective Micro Genetic Algorithm. The aim of this research is to introduce the concept of Multi Objective Micro Genetic Algorithm as a tool to solve irregular airline operation, combine and reroute problem. The experiment result indicated that the model could obtain optimal solutions within a few second.

Keywords—Irregular Airline Operation, Combine and Reroute Routine, Genetic Algorithm, Micro Genetic Algorithm, Multi Objective Optimization, Evolutionary Algorithm.

I. Introduction

THE planned flight schedule is disrupted when an *Irregular Airline Operation* (*IAO*) situation occurs. Causes of disruption range from bad weather, labor strike, airport or aircraft repairing, communication device failure, etc. Solutions to the IAO situations can be formulated in many fashions, such as allowing aircraft swaps, combining flights, rerouting flight sector, or some combinations thereof. We consider a specific case in which flight sector can be simultaneously combined and rerouted. Several costs associated with this change span from passengers' compensation fees, transportation fees, accommodation fees, or loss of revenue to other airlines.

As there are several criteria, costs and options available, it demands complex computation in order to reach an optimal solution. Several approaches were introduced to minimize these costs. We propose the use of *Multi Objective Micro Genetic Algorithm* to solve this "combine and reroute" problem based on Thai Airways domestic case.

II. Background

"Flight" in airline operation means a trip of an aircraft traveling from one place to another place. The trip information comprises the name of the city and the time that the aircraft depart from and arrive to. An example of a flight detail is as follow:

TG001 BKK URT 0130 0330

where TG001 is flight number, BKK is the abbreviation of the departure city which is Bangkok, URT is the abbreviation of the arrival city which is Surattani, the string 0130 means the aircraft departs BKK at 1.30 am and the string 0330 means the aircraft arrives URT at 3.30 am. The flight information always appears in the flight schedule.

In aircraft rotation table used by Operation officer, the flight information mentioned earlier is encoded into the form shown in Fig. 1.

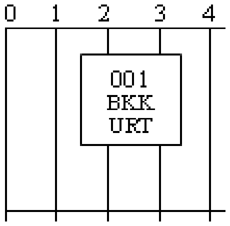


Fig. 1 A flight in aircraft rotation table

"Sector" refers to the combination of more than two flights which start and end at the same city. The example of a sector is as follow.

BKK URT BKK

The above sector comprises the flight BKK URT and URT BKK together which means the aircraft departs BKK to URT then rests for some period of time which is equal to ground time. Then it departs from URT to BKK. Sector does not appear in flight schedule but instead it appears in aircraft rotation table used by Operation officer. Fig. 2 shows a sector in aircraft rotation table.

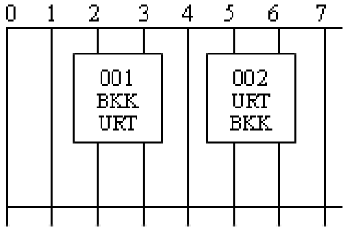


Fig. 2 A sector in aircraft rotation table

The gap between each flight is called ground time. Ground time is the rest period of an aircraft before flying the next flight. Minimum ground time must be maintained all the time through aircraft operation.

Soottipoom Yaowiwat is with the Chulalongkorn University, 254, Pathumwan, Bangkok, Thailand, 10330. He is now a graduate student at the Department of computer engineering, Faculty of Engineering, Chulalongkorn University (e-mail: tummy_navy@hotmail.com)

Manoj Lohatepanont is currently a faculty member at the Department of Civil Engineering, Chulalongkorn University, 254, Pathumwan, Bangkok, Thailand, 10330. He is an expert in optimization models for airline planning and operation (e-mail: manoj.l@chula.ac.th).

Proadpran Punyabukkana is an assistant Dean of Engineering, and an assistant professor in Computer Engineering, Chulalongkorn University, 254, Pathumwan, Bangkok, 10330, Thailand (e-mail: proadpran.p@chula.ac.th).

"Irregular Airline Operation (IAO)" is an operation done when the daily operations of regularly scheduled airline carriers are prone to unexpected irregularities which develop from several factors ranging from severe weather conditions to the unavailability of eligible flight crew. In many cases, these factors can have a significant impact on an airline's operations, resulting in substantial deviation from the planned schedule of services. According to the US department of transportation recorded on irregularities in domestic flights, the causes of irregularities can be shown as follow [4].

1. Weather – Wind, fog, thunderstorm, low cloud ceiling
2. Equipment – Air traffic radar/ computer outage
3. Runway – Unavailable because of construction, surface repair, disabled aircraft
4. Volume – Aircraft movement rate exceeds capacity of the airport at a given time
5. Other – Anything excluding the above factors

Possible solutions for irregularities based on Thai Airways might be:

A. *Swap the Aircraft in the Same Fleet Type*

Replace the irregular aircraft with a new one, which has the same type as the irregular one that is available during the operational period of the irregular one.

B. *Swap the Aircraft between Different Fleet Type*

The method is the same as 1 but done on different fleet type of replaced aircraft.

C. *Cancel and Combine*

Cancel that flight and transfer all of the passengers to another scheduled flight. The combination may be done in one aircraft, if the capacity allows, or in several aircrafts.

D. *Delay the Flight*

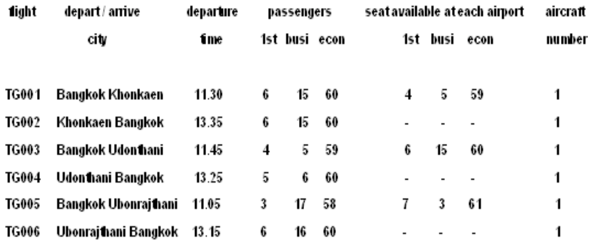
Delay the flights until the aircraft is ready.

E. *Combine and Reroute*

Combine the passengers of the sector that have the same departure and arrival city together in one aircraft. An example is combining sector BKK URT BKK with BKK UDT BKK, where BKK is the departure and arrival city of each sector. The resulting combination will be BKK URT UDT BKK. The combination means an aircraft firstly departs BKK and arrives URT. After waiting for some period of time which is equal to ground time, the aircraft then picks the passengers up and departs URT to UDT. After that the aircraft waits again, according to the ground time, and leaves UDT to BKK.

III. Problem Domain

The focus of our work is to find best possible solution for the "combine and reroute" problem. The manual approach for combining and rerouting used by Thai Airways is illustrated in Fig. 3.



flight	depart / arrive city	departure time	passengers			seat available at each airport			aircraft number
			1st	busi	econ	1st	busi	econ	
TG001	Bangkok Khonkaen	11.30	6	15	60	4	5	59	1
TG002	Khonkaen Bangkok	13.35	6	15	60	-	-	-	1
TG003	Bangkok Udonthani	11.45	4	5	59	6	15	60	1
TG004	Udonthani Bangkok	13.25	5	6	60	-	-	-	1
TG005	Bangkok Ubonrajthani	11.05	3	17	58	7	3	61	1
TG006	Ubonrajthani Bangkok	13.15	6	16	60	-	-	-	1

Fig. 3 An example flight schedule

Suppose that flight TG001 which departs from Bangkok and arrives at Khonkaen is interrupted (an irregular flight) for some reason that makes it unable to operate, the following steps are to be taken.

Step 1: Search throughout flight schedule for flights departing from the same city as the irregular flight, in this case two candidate flights is found. Those candidate flights are TG003 and TG005.

Step 2: Compute the delay between the irregular flight, TG001, and each candidate flight, TG003 and TG005. The delay can be calculated by finding the time difference between a candidate flight departure time and the irregular flight departure time. For example, the delay between TG001 and TG003 is fifteen minute whereas there is no delay between TG001 and TG005, because TG005 departs before TG001. At this step TG005 is dismissed from the candidate set because it has already departed. Then, check whether the aircraft of candidate flight can land on the airport of irregular flight, if it can not land on the airport, the candidate flight will be removed from candidate set.

Step 3: Calculate the number of excess passengers after transferring the passengers from the irregular flight to the candidate flight. The example is transferring passengers from TG001 to TG003. The number of excess passengers can be calculated as follow.

$$EP1_c = CS1_c - IRP1_c \quad (1)$$

Where, $EP1_c$ = the number of excess passenger in each class, First, business and economy class.

$CS1_c$ = the number of available seats in each class of candidate flight.

$IRP1_c$ = the number of passengers in each class of irregular flight.

Subscript c denotes each class of the seat or passenger, First, business and economy class.

For example, the number of excess passenger after combining the passengers from TG001 to TG003 is 0 because the number of passengers in each class of TG001 is equal to the number of seats available of its own class in TG003. If $EP1_c$ is more than or equal to -3, then do step 4. Otherwise If

$EP1_c$ is less than -3 then terminates the routine and go to step 8.

Step 4: Search through the flight schedule the flight whose departure city is the same as the arrival city of the irregular flight, for example TG002. Then check whether the aircraft of candidate flight can land on the airport of irregular flight, if it can not land on the airport, the candidate flight will be removed from candidate set. After that, calculate the number of excess passengers after transferring the passengers from the flight searched to the candidate flight.

Step 5: Calculate the number of excess passengers for each class at the arrival city of the irregular flight, after transferring passengers from TG002 to TG003. The calculation can be done using the following equation.

$$EP2_c = CS1_c - IRP2_c \qquad (2)$$

Where, $EP2_c$= the number of excess passengers in each class after transferring the passengers from the flight searched, in step 4, to the candidate flight.

$CS1_c$ = the number of available seats in each class of candidate flight derived from step 3.

$IRP2_c$ = the number of passengers in each class of the flight obtained from step 4.

If $EP2_c$ is more than or equal to -3, then do step 6. Otherwise If $EP2_c$ is less than -3 then terminates the routine and go to step 8.

Step 6: Search through the flight schedule for the flight whose departure city is the same as the arrival city of the candidate flight, for example TG004. Then calculate the number of excess passengers after transferring the passengers from the flight searched to the candidate flight, for example from TG004 to TG003.

Step 7: Calculate the number of excess passengers for each class at the arrival city of the candidate flight, after transferring passengers from TG004 to TG003. The calculation can be done using the following equation.

$$EP3_c = CP2_c - CP3_c \qquad (3)$$

Where, $EP3_c$= the number of excess passengers in each class after transferring the passengers from the flight searched, in step 6, to the candidate flight.

$CP2_c$ = the number of passengers in each class of the candidate flight, TG003.

$CP3_c$ = the number of passengers in each class of the flight obtained from step 6.

If $EP3_c$ is more than or equal to -3, then the selected candidate flight is added to the solution set. Otherwise If $EP3_c$ is less than -3, terminates the routine and go to step 8.

Step 8: apply step 2 - 7 to another candidate flight.

Even though the routine is commonly used, some significant short comings of the routine have been found. The first short coming is that the total delay of the schedule, after combining and rerouting the flight, is usually not taken into the routine because it takes too much time for human to compute to solve IAO. The total delay of the schedule is illustrated in Fig. 4 (a) and 4 (b).

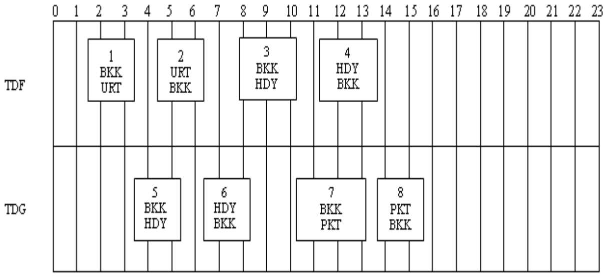


Fig. 4 (a) The original aircraft rotation table

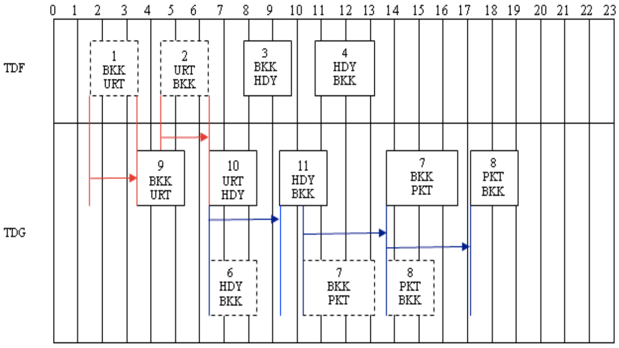


Fig. 4 (b) A new aircraft rotation table after combining and rerouting the flight

Fig. 4 (a) & (b) indicate the simulated aircraft rotation table based on Thai Airways domestic event. The row of each table represents an aircraft operation, string TDF and TDG are aircraft tail number or their ID, whereas the column represents each hour of a day. Fig. 4 (a) is the original schedule before the aircraft TDF was disrupted. Fig 4 (b) is a new schedule after combining and rerouting the flights. The original flights that have been delayed are shown in the dashed boxes.

For illustration purpose, suppose the aircraft TDF was disrupted at 0130. For this reason flight number 1, BKK-URT, and flight number 2, URT-BKK, could not be flown. Note that, in IAO, if any flights in a sector can not be flown by a particular aircraft, then the whole sector will not be flown by that aircraft. A solution to the disrupted event is to combine the sector BKK-URT-BKK flown by the aircraft TDF with the sector BKK-HDY-BKK flown by the aircraft TDG. The resulting combination will be BKK-URT-HDY-BKK which was flown by the aircraft TDG.

A consequence of combining and rerouting the flights that usually occurs is passengers' delay. The red arrows in the table indicate the delay of the original flights flown by the aircraft TDF. For example flight number 1 was delayed from 0130 to 0330, which is the departure time of the flight number 9. The blue arrows indicate the delay of the original flights flown by the aircraft TDG. For example flight number 6 was delayed from 0620 to 0910, which is the depart time of the flight number 11. The total delay of the schedule is the sum of the delay of all flights in the schedule, after combining and rerouting the flights.

The second one is the limitation of searching, through the candidate space, imposed on human capability. For this reason, the routine does not guarantee optimal solution. To overcome the mentioned shortcoming, computer based application based on the following techniques are used in the irregular airline operation.

IV. Literature Review

Several methodologies for solving IAO problem have been used for decades. Those approaches are Linear Programming, Branch and Price, Modeling technique, and Decision Support System (DSS). In an early period of the past decade, the decision support systems are quite popular in many airlines such as DSS framework for airline flight cancellations and delays at United Airlines. The solution procedure behind the application is minimum-cost network flow problem [3].

The second one is the application of the integration of computer science and operations research in decision support system for airline system operations control. The application integrates real-time flight following, aircraft routing, maintenance, crew management, gate assignment and flight planning with dynamic aircraft rescheduling and fleet re-routing algorithms for irregular operations, the algorithms involve LP problem solving. The system developed on distributed desktop UNIX workstation, networked through Ethernet and X Windows Motif graphical user interface [3].

The third one is a decision support framework for handling schedule perturbations which incorporate concepts published by United Airlines. The framework is based on a basic schedule perturbation model constructed as a dynamic network from which several perturbed network models are established for scheduling following irregularities. The authors formulate pure network flow problems and solved them using network simplex method and Lagrangian relaxation with subgradient methods [3].

The fourth one is the Inconvenienced Passenger Rebooking System, developed by Delta Airline that allows the airline to notify passengers of cancellations or delays and aid in passenger flow recommendations. Resource Management Operation Control (ROC) database/graphical display system developed by Garuda Airlines, Indonesia. The system is used for monitoring actual operations [4].

After the boom of DSS in airline operation, the modeling techniques are becoming popular. For example, the three multi-commodity network-type models for determining a recovery schedule for all aircraft operated by a large carrier following a hub closure. The first is a pure network with side constraints, the second is a generalized network, and the third is a pure network with side constraints in which the time horizon is discretized [2] and a model for projecting flight delays during irregular operation conditions to support on-time performance of airlines schedule. The model can be derived from transforming the flight scheduled to a network which each node represents the various states of the aircraft such as flight departure, wheel-off, wheel-on, flight arrival and aircraft ready. Each node is attached with the time that each event is occurred. The arcs represent the activity between each state such as the activity between flight departures [5].

Even the DSS and the modeling techniques are well-known, they still have an important limitation. The limitation is its capability to solve complex optimization problem. The DSSs and modeling techniques are constructed to ease human controller's decision making but they can not obtain the optimal solution for the IAO problem. For this reason a Branch and Price approach is proposed to enhance the shortcoming of the previous systems and techniques implemented in an early of the decade. An example of Branch and Price approach is a column-generation scheme to solve operational aircraft maintenance by optimizing LP relaxation problem called the restricted master problem and using branch and bound search tree to obtain the integer solution. The column generation scheme then is employed at each node of the tree [5].

Even LP, Branch and Price, Modeling technique and decision support system are quite popular, there are some limitation posted behind those methods. Those limitations comprise firstly a difficulty to formulate a correct objective and constraints function in LP for IAO problem and a difficulty in solving them, both manually and electronically. Also solving the wrong objective and constraints function subjects to incorrect solution. Secondly most of the modeling techniques provide only the network model for human controllers without solution deriving.

According to some difficulty posted by LP-based approaches and the limitation of network modeling technique, Multi Objective Evolutionary Algorithm (MOEA) method was introduced. MOEA overcomes traditional linear programming in various issues. For example it has minimum requirements regarding the problem formulation; objectives can be easily added, removed, or modified. Also MOEA can produce more optimal solutions that LP does. Furthermore MOEA has been demonstrated in various applications that evolutionary algorithms are able to tackle highly complex problems and therefore they can be seen as an approach complementary to traditional methods such as integer linear programming [6].

The first attempt, so far, of applying MOEA to IAO has been observed since 2006. Tung proposed the Applications of MOEA to Airline Disruption Management. The model performed quite well on real data set. However some limitations have been found. Firstly their combination of two objective functions, delay and swap cost, and misconnection penalty into one single objective function can not be used to observe tradeoff between delay and swap costs of the solutions. Hence some good quality solution may be discarded. Secondly the best convergence curve did not seem to converge within a given time. This may due to the lost of optimal solution during evolution [7].

Due to the fact that IAO problem need to be solved real time, MOEA should be designed to support this issue. Even the previous attempt could solve IAO problem effectively, there are chances to reduce the computational time. Those chances are to eliminate fitness assignment and ranking of large amount of population, which consume expensive computational time, before doing the selection process. That means the population size will be reduced to a small number and the fitness assignment will be done immediately before the selection process whereas fitness ranking is neglected. Those concepts described earlier can be accomplished by utilizing micro GA. Also applying bias to genetic operators such as crossover and mutation and utilize constraint violation search to reduce population size may help solving the problem better. Those concepts were implemented in MOMGA.

V. Genetic Algorithms

Genetic Algorithms (GAs) provide a learning method motivated by an analogy to biological evolution. Rather than search from general-to-specific hypotheses, or from simple-to-complex, GAs generate successor hypotheses by repeatedly mutating and recombining parts of the best currently known hypotheses. At each step, a collection of hypotheses called the current population is updated by replacing some fraction of the population by offspring of the most fits current hypotheses. The process forms a generate-and-test beam-search of hypotheses, in which variants of the best current hypotheses are most likely to be considered next. The popularity of GAs is motivated by a number of factors including [11]:

1. Evolution is known to be a successful, robust method for adaptation within biological systems.
2. GAs can search spaces of hypotheses containing complex interacting parts, where the impact of each part on overall hypothesis fitness may be difficult to model.
3. Genetic algorithms are easily parallelized and can take advantage of the decreasing costs of powerful computer hardware.

The problem addressed by GAs is to search a space of candidate hypotheses to identify the best hypothesis. In GAs the “best hypothesis” is defined as the one that optimizes a predefined numerical measure for the problem at hand, called the hypothesis fitness. For example, if the learning task is the problem of approximating an unknown function given training examples of its input and output, then fitness could be defined as the accuracy of the hypothesis over this training data. Hypotheses in GAs are often represented by bit strings, so that they can be easily manipulated by genetic operators such as mutation and crossover [11].

The population of GAs, such as hypotheses, can be evolved to an optimal one through the genetic operators such as mutation and crossover. It is expected that the quality of the current population can be improved through the selection and genetic operators. The basic crossover operators are Single-point crossover, Two-point crossover and Uniform crossover. The following diagram illustrated those basic crossover and mutation operators [11].

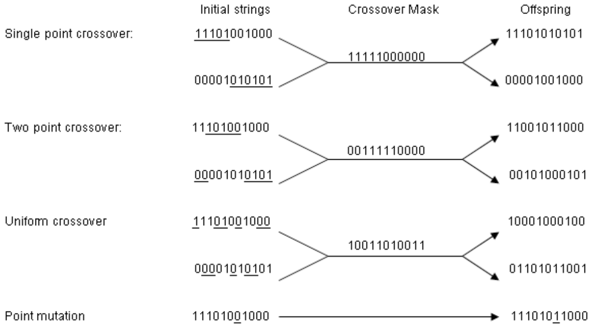


Fig. 5 Genetic operators

The following diagram shows a basic GA.

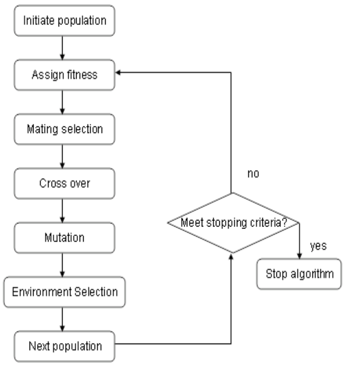


Fig. 6 Genetic Algorithms

In real world, GA has been applied to many optimization problems of very large space such as circuit layout, robot control process, learning Artificial Neural Network, job-shop scheduling, etc.

VI. Micro Genetic Algorithms

The term micro-genetic algorithm (micro-GA) refers to a small-population genetic algorithm with reinitialization. The concept was proposed by some theoretical results obtained by Goldberg, according to which a population size of 3 was sufficient to converge, regardless of the chromosomic length. The process introduced by Goldberg was to start with a small randomly generated population, then apply to it the genetic operators until reaching nominal convergence (e.g.,when all the individuals have their genotypes either identical or very similar), and then to generate a new population by transferring the best individuals of the converged population to the new one. The remaining individuals would be randomly generated [9].

VII. Multiobjective Optimization

Multiobjective optimization (also called multicriteria optimization, multiperformance or vector optimization) can be defined as the problem of finding a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term “optimize” means finding such a solution which would give the values of all the objective functions acceptable to the designer. Formally, we can state it as follows [12].

Find the vector $x = [x_1, x_2,\ldots, x_n]^T$ which will satisfy the m inequality constrints: $g_i(x) \geq 0$, $i = 1,2,\ldots,m$. The p equality constraints $h_i(x) = 0$, $i = 1,2,\ldots,p$ and optimizes the vector function $f(x) = [f_1(x), f_2(x),\ldots, f_k(x)]^T$ where $X = [x_1, x_2,\ldots, x_n]^T$ is the vector of decision variables. The set of optimal solutions in the decision space X is in general denoted as the Pareto set (X^*) and we denote its image in objective space as Pareto front ($Y^*= f(X^*)$).

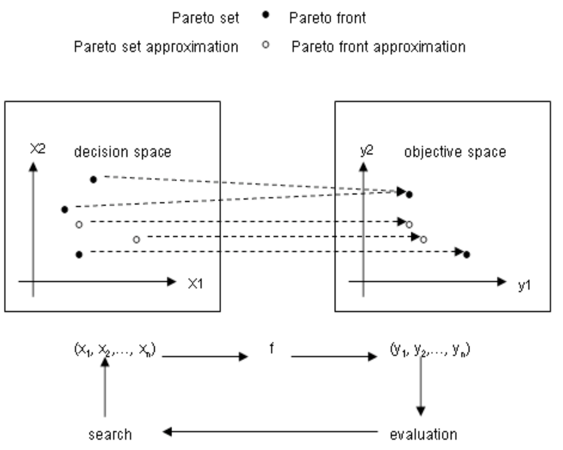


Fig. 7 Mapping from decision space to objective space

VIII. Multi Objective Evolutionary Algorithm

Generating the Pareto set can be computationally expensive and is often infeasible, because the complexity of the underlying application prevents exact methods from being applicable. For this reason, a number of stochastic search strategies such as evolutionary algorithms, tabu search, simulated annealing, and ant colony optimization have been developed: they usually do not guarantee to identify optimal trade-offs but try to find a good approximation, i.e., a set of solutions whose objective vectors are (hopefully) not too far away from the optimal objective vectors [13].

Roughly speaking, a general stochastic search algorithm consists of three parts: i) a working memory that contains the currently considered solution candidates, ii) a selection module, and iii) a variation module. A selection module consists of mating and environmental part. Mating selection aims at picking promising solutions for variation and usually is performed in a randomized fashion. In contrast, environmental selection determines which of the previously stored solutions and the newly created ones are kept in the internal memory. The variation module takes a set of solutions and systematically or randomly modifies these solutions in order to generate potentially better solutions. In summary, one iteration of a stochastic optimizer includes the consecutive steps mating selection, variation, and environmental selection; this cycle may be repeated until a certain stopping criterion is fulfilled [13].

Many stochastic search strategies have been originally designed for single-objective optimization and therefore consider only one solution at a time, i.e., the working memory contains just a single solution. As a consequence, no mating selection is necessary and variation is performed by modifying the current solution candidate. In contrast, an evolutionary algorithm is characterized by three features which are;

1. A set of solution candidates is maintained.
2. A mating selection process is performed on this set.
3. Several solutions may be combined in terms of recombination to generate new solutions.

By analogy to natural evolution, the solution candidates are called individuals and the set of solution candidates is called the population. Each individual represents a possible solution, i.e., a decision vector, to the problem at hand; however, an individual is not a decision vector but rather encodes it based on an appropriate representation [13].

Basically MOEA has very similar algorithms to the traditional GA, see Fig. 3. However, unlike GA, MOEA has more than one objective function. For this reason its fitness assignment scheme is different from the traditional GA. There are three ways of assigning fitness to each individual in MOEA. Those methods are aggregation base, criterion base and dominance base.

1. Aggregation base

   This approach is built on the traditional techniques for generating trade-off surfaces by aggregating the objectives into a single parameterized objective function. The parameters of this function are systematically varied during the optimization run in order to find a set of nondominated solutions instead of a single trade-off solution [13].

2. Criterion base

   Criterion-based methods switch between the objectives during the selection phase. Each time an individual is chosen for reproduction, potentially a different objective will decide which member of the population will be copied into the mating pool [13].

3. Dominance base

   This method calculates an individual’s fitness on the basis of Pareto dominance [10], and different ways of exploiting the partial order on the population. The following diagram indicates tradeoff surface for two objective functions in dominance base fitness assignment. The detail of the trade off surface will be described later on in MOMGA architecture section.

IX. Multi Objective Micro Genetic Algorithm for Combine and Reroute Problem

The flight combine and reroute problem, based on Thai Airways domestic case, can be viewed as an optimization problem whose formulation is as follow.

Minimize

$$TD = \sum_{i=1}^{m} \sum_{j=1}^{n} (Y_{i,j} - X_{i,j}) \qquad (4)$$

$$EP = \sum_{i=2}^{3} (PF_{i-1} - SF_i) + \sum_{i=2}^{3} (PB_{i-1} - SB_i) + \sum_{i=2}^{3} (PE_{i-1} - SE_i) \qquad (5)$$

Subject to

$$\sum_{k=1}^{o} \sum_{l=1}^{p} (AP_k - AC_l) \neq 0 \qquad (6)$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} (Y_{i,j} - X_{i,j}) > 0 \qquad (7)$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} (Y_{i,j} - X_{i,j}) < 600 \qquad (8)$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n-1} C_{i,j} = 0 \qquad (9)$$

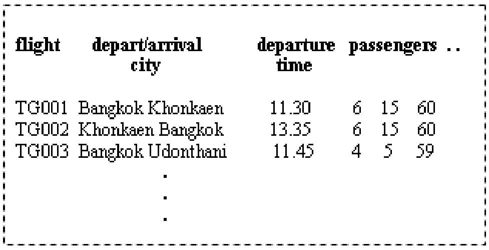
$$\sum_{i=2}^{3} (PF_{i-1} - SF_i) + \sum_{i=2}^{3} (PB_{i-1} - SB_i) + \sum_{i=2}^{3} (PE_{i-1} - SE_i) < 16 \qquad (10)$$

Where

TD = Total delay of the schedule.
$Y_{i,j}$ = New departure time of a flight.
$X_{i,j}$ = Original departure time of a flight.
EP = Excess passenger.
PF_{i-1} = First class passenger of flight i-1.
SF_i = Available first class seat of flight i.
PB_{i-1} = Business class passenger of flight i-1.
SB_i = Available business class seat of flight i.
PE_{i-1} = Economic class passenger of flight i-1.
SE_i = Available Economic class seat of flight i.
$C_{i,j}$ = Misconnected flight cost.
AP_k = Airport Code of airport k.
AC_l = Aircraft Code of aircraft l.
m = Amount of aircraft.
n = The number of flights in every courses.
o = The number of irregular airports.
p = The number of candidate aircrafts.
subscript i = a specific aircraft.
subscript j = a specific flight.
subscript k = airport code at airport k.
subscript l = aircraft code of aircraft l.

The MOMGA model was designed to find new routes and flight combinations whose total delay to the schedule, objective function (1), and excess passengers, objective function (2), are minimized whereas various constraints are maintained. Notice that the calculation of the objective function (1) is done based on Fig. 2 whereas the objective function (2) is calculated based on the combine and reroute routine section. Constraint (3) ensures that a candidate aircraft can be landed on the irregular airport .Constraint (4) insures that the solution candidate flights can not depart earlier than irregular flight. Constraint (5) ensures that solutions obtained cause total delay to the schedule less than 600 minutes. Constraint (6) assures that misconnection flights will never occur. Constraint (7) certifies that the number of total excess passengers of a solution flight will not exceed sixteen passengers. The following section describes MOMGA model in details.

A. Chromosome Representation



flight	depart/arrival city	departure time	passengers ..
TG001	Bangkok Khonkaen	11.30	6 15 60
TG002	Khonkaen Bangkok	13.35	6 15 60
TG003	Bangkok Udonthani	11.45	4 5 59
	.		
	.		
	.		

Fig. 8 The synthetic flight schedule

Fig. 8 indicates a synthetic flight schedule based on real schedule obtained from Thai airways, it is used for the experiment in this research. The schedule contains 16 fields of data which are flight number, sector (the name of depart and arrival city), departed time, arrival time, the number of first, business and economic class passengers, the name of departed airport, airport code, the number of seats available in first, business and economic class, aircraft code, aircraft number, region code and city code. Note that aircraft code and airport code is used to find the restriction between irregular airport and candidate aircrafts. The reason that the simulated schedule was used is that the real schedule contains insufficient data for combine and reroute problem solving. For example when experts solve the combine and reroute problem, they always take the number of booked passengers into an account in order to calculate the number of excess passengers, who can not be carried by the candidate flight. Unfortunately the real schedule does not contain the number of booked passengers. Therefore the missing information was simulated.

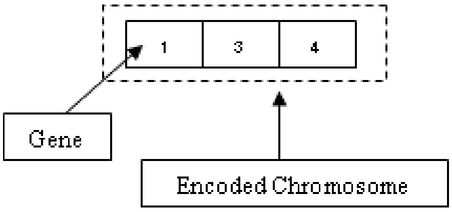


Fig. 9 The chromosome representation

Fig. 9 shows the encoded chromosome used by the MOMGA model. Each gene represents a code number of a particular flight of the schedule in Fig. 6. The first gene corresponds to the code number of the irregular flight whereas the second and third genes contain the code number of candidate flights. The chromosome can be simply encoded back into the aircraft rotation table style for total delay calculation.

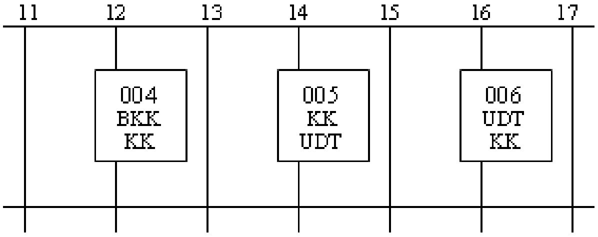


Fig. 10 The aircraft rotation form of the chromosome in Fig. 9

B. MOMGA Architecture

MOMGA architecture has been adopted from the model proposed by Carlos, A.[8]. However the extensions from his work are violation search, micro GA population control, bias crossover and goal base selection. MOMGA architecture comprises three important types of memories, Replaceable, Non-replaceable and External memory and six processes. Replaceable and Non-replaceable memory act as population pool used to provide initial population to micro GA for each iteration of the GA. The population residing in Replaceable memory can be replaced by non-dominated solution over time in order to converge to global convergence whereas the Non-replaceable memory can not be changed over time in order to provide diversity for the initial population of the micro GA. The External memory is used as a memory to store non-dominated solution at each nominal convergence.

The algorithm starts with random generating population to replaceable and non-replaceable memory. After that four individuals randomly selected from those memories are initialized as micro GA initial population. Then Violation Search process is employed to find out constraint violated solution. The solutions violating the constraints (3) to (7) are diminished immediately from population set. The number of survived individuals is then checked against micro GA population size, which is 4. If the number less than 4 then the algorithm reinitializes more initial population for micro GA. This routine continues until the initial population size of micro GA is more than or equal to 4. If the number is more than 4, best four solutions are selected, base on non-dominance criteria, and used as the initial population for micro GA. After that those initialized population will be tournament selected and assigned crossover and mutation operators respectively. After each operation violation breaking is also checked and solutions violating the constraints will be removed from population set.

Those processes mentioned earlier continue running until the nominal convergence, the number of iteration in micro GA, is reached. Since the nominal convergence is reached, a best solution is greedily selected, base on non-dominance criteria, and added to External and Replaceable memory. If the External memory is overloaded, the algorithm will create density region and each individual will be added to a particular region. Then every region is assigned a density value, which is the number of population in that region. The region whose density is less crowded than the threshold is more preferable for an individual entering the External memory. The density region is illustrated as follow.

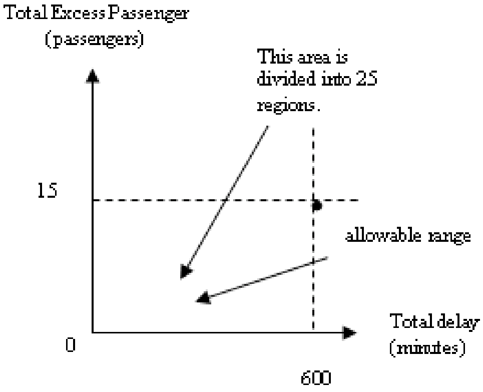


Fig. 11 The density region

The algorithm described earlier is re-executed until a termination criterion is reached. The following diagram shows the architecture of MOMGA.

C. Genetic Operator

Two types of genetic operators were applied in this research. They are one point crossover and mutation. The algorithm uses tournament selection to select two chromosomes and assign the crossover operator to the dominated ones. After crossover, a randomly selected chromosome is selected and mutated. The following picture indicates the crossover operator.

Notice that there is bias in the above crossover operator. Only the second and third genes are allowed in crossover operation. The crossover can occur only with the genes having the same column. The reason that why the first gene will never participate in crossover operation is that if the first gene is crossed over and it becomes another flight rather than irregular flight, then the chance of converge to an optimal solution is reduced. The same reason is valid for crossing over on the same column in second and third gene.

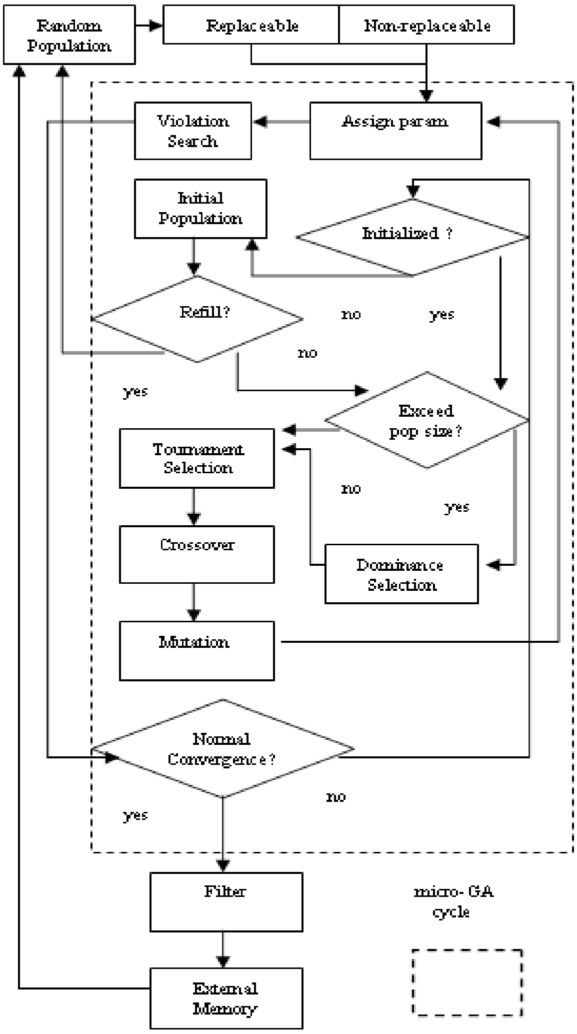


Fig. 12 The MOMGA architecture

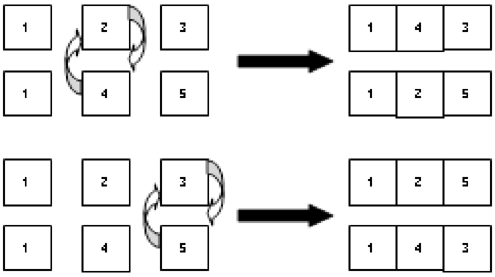


Fig. 13 Crossover operator

If they are not crossed within the same column the chance of misconnection is increased. Hence the chance of converging to optimal solution will decrease. Furthermore the random point mutation is applied to the parents, the second and third genes of the parents are mutated to any random flight code.

D. Constraint Violation Search

Every candidate solutions are checked whether they break constraints (4) – (7). Any solutions contravening those constraints will be eliminated from solution sets. The benefit of constraint violation search is that the constraint violated solutions can be eliminated before selection for the best solution occurs. Therefore computational time on constraint violated solution in the selection process is reduced. Another advantage of constraint violation search is that it helps the algorithm to avoid fitness evaluation and ranking scheme, again reducing computational time consuming.

E. Dominance Selection

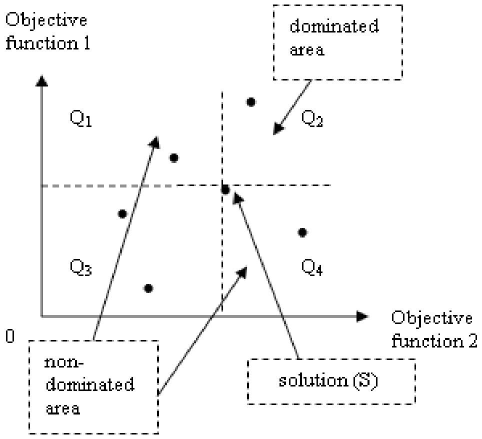


Fig. 14 Trade off surface

Fig. 14 denotes the trade-off surface used by decision maker to find out desirable solutions. In multi-objective minimization problem base on Pareto dominance, solution in Q_1, Quadrant 1, and Q_4 are equally preferable to a solution S, non-dominated by S, whereas solution in Q_2 is less preferable than S because both of its objective function value is more than S, dominated by S. On the other hand, solutions in Q_3 dominate S because both of their objective function values is less than S [10].

However a goal was introduced in order to improve the selection process. By proposing the goal, the coordinate (0,0), when two non-dominated solutions are taken into the selection process, the solution which has less Euclidean distance between its position and the goal on the trade-off surface will be selected. The MOMGA model utilizes the mentioned scheme to select individuals for next generation by setting the objective function 1 to be the function of total excess passengers (1) and the objective function 2 to be the function of total delays of the schedule after combining and rerouting the flights (2). MOMGA applies greedy selection to find best solution based on Pareto dominant. By using this method, fitness ranking is not necessary. This approach is also very efficient because the population size is very small, 4 individuals.

F. Elitism

MOMGA model applies dominance based selection to choose the best two solutions for next generation. A best solution in every two epochs is added into an external file and replaceable memory, if it is not dominated by any solution in them. The approach ensures that non-dominated solution will survive to the next generation [8].

G. Convergence

By utilizing elitism scheme in 4.6, as the time runs on, the solution will be converged to true Pareto front, optimal solution [8].

H. Diversity Preservation

To assure that the non-dominated solutions obtained are diverse, duplicate detection and density check methods are applied in MOMGA. Any non-dominated solution entering the external memory which duplicate with the one in the memory is not allowed to save in the memory. Furthermore when the memory size is full the tradeoff surface will be divided into 25 regions. A solution just entering the memory is allocated a region. The solution whose region has density exceeding the threshold is diminished from the region. By applying those two methods, solutions obtained are diverse [8].

X. Experiment

The experiment was conducted in order to observe the convergence behavior and running time until global convergence is reached. Test data is a simulated flight schedule, see Fig 7, containing 100 flights and 8 aircrafts. The experiment was implemented in Java application which runs on 1.47 GHz CPU with 1GB RAM PC.

The parameters used by the micro GA for this experiment were: size of replaceable memory = 500 individuals, size of non-replaceable memory = 100 individuals, size of external memory = 50 individuals, number of iterations = 200, number of iterations of the micro GA (to achieve nominal convergence) = 10, number of subdivisions of the density region = 25, crossover rate = 0.5, mutation rate = 0.25, micro GA population size = 4 individuals.

The following table indicates the total delay of the schedule and the total excess passenger s of the non-dominated solutions in the MOMGA run. As the number of GA round increase, the non-dominated solutions are developed to the pareto front which are (40,9) and (90,3). Notice that the non-dominated solutions had been approached the goal (0,0), as set in the model, from the top row to the bottom row.

TABLE I
THE VALUE OF EACH NON-DOMINATED SOLUTION ON THE OBJECTIVE SPACE

Total delay of the schedule	Total excess passengers
390	4
370	13
115	10
85	9
105	3
40	9
90	3

Fig. 15 illustrates the non-dominated solution, produced by MOMGA, recorded in the above table on the objective space.

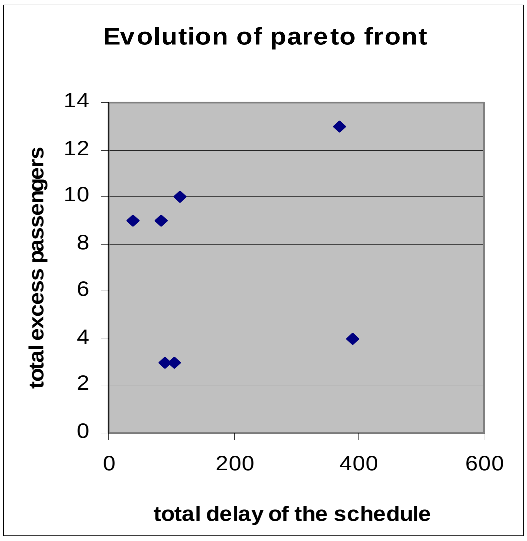


Fig. 15 The non-dominated solution produced by MOMGA during the application run.

The following diagram shows the average total delay of non-dominated solution kept in External memory of each generation. As more generation is produced, the member of External memory tends to converge to particular non-dominated solutions which yield the same average total delay, even the next generation was produced. Those solutions have found to be a set of global optimal solution since their total delay and the numbers of excess passengers are optimal in manual calculation. The average time of converging to optimal solution is about 5 second.

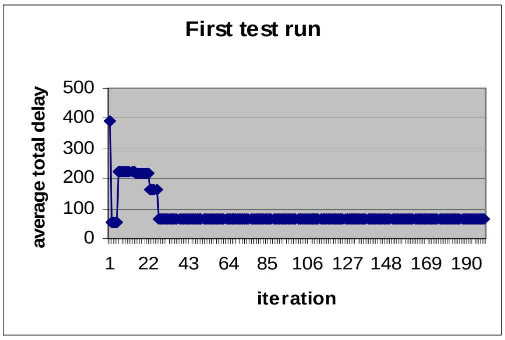


Fig. 16 The average total delay of the first 200 epochs

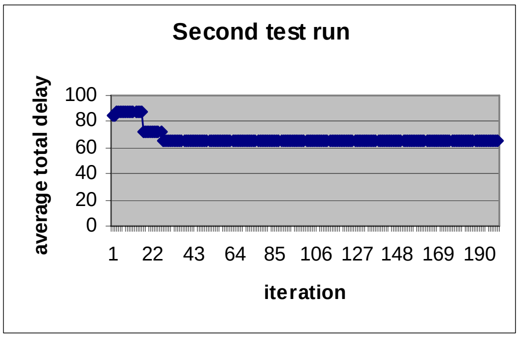


Fig. 17 The average total delay of the second 200 epochs

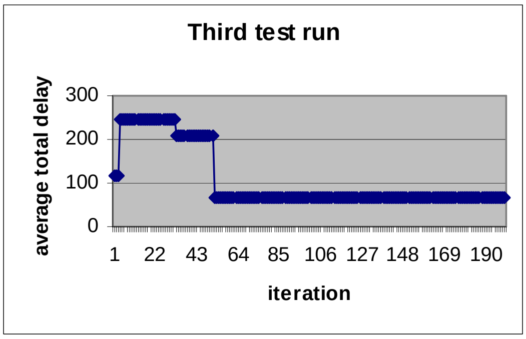


Fig. 18 The average total delay of the third 200 epochs

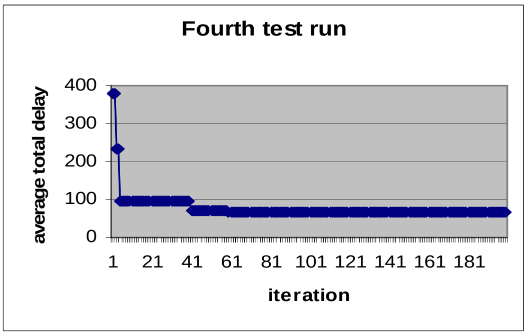


Fig. 19 The average total delay of the fourth 200 epochs

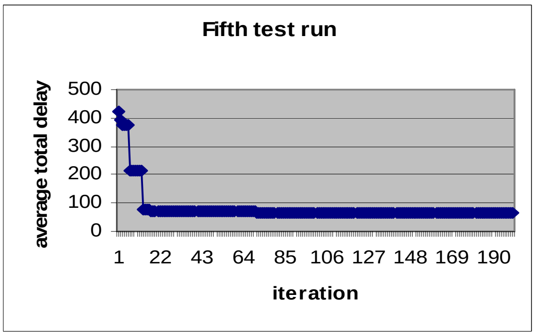


Fig. 20 The average total delay of the fifth 200 epochs

XI. Conclusion

The experiment result has shown that the model could obtain the optimal solutions and converge to the Pareto front. However it still needs improvement for more complex task, which is combining and rerouting international flights. Moreover the synthetic part of the schedule should be replaced by real data in order to support more realistic environment.

Acknowledgment

The authors wish to thank Mr. Komsan Chotikateerakul, the Vice Chief of Thai Airways Operational Flight Center, who has contributed a lot of time to teach the authors the basic understanding and problem solving for Irregular Airline Operation.

References

[1] S. Abdulkadir, B. Rajan, R. Christopher, “A branch-and-price approach for operational aircraft maintenance routing”, *European Journal of Operational Research,* vol 175, pp. 1850 – 1869, 2006.

[2] G. Benjamin, Y. Gang, B. Jonathan, “Multiple fleet aircraft schedule recovery following hub closures”, *Transport research part A*¸ vol 35, pp. 289-308, 2001.

[3] M. Dennis, “Decision support for airline system operations control and irregular operations”, *Computer Operational Research,* vol 23, no 11, pp. 1083 – 1098, 1996.

[4] D. Michael, C. Delano, “Irregular airline operations: a review of the state-of-the-practice in airline operations control centers”, *Journal of Air Transport Management*, vol4, pp. 67-76, 1998.

[5] F. Khaled, S. Sharmila, R. Sidhartha, A. Ahmed, “A model for projecting flight delays during irregular operation conditions” *Journal of Air Transport Management*, vol10, pp. 395-394, 2004.

[6] T. Back, U. Hammel, E. Schwefel, “Evolutionary Computation: Comments on the history and current state”, *IEEE Transactions on Evolutionary Compution*, vol1, pp. 3-17, 1997.

[7] L. Tung-Kuan, J. Chi-Ruey, L. Yu-Ting, T. Jia-Ying,, “Applications of Multi- objective Evolutionary Algorithm to Airline Disruption Management”, *IEEE*, pp. 4130 – 4135, 2006.

[8] A. Carlos, P. Gregorio, “A Micro-Genetic Algorithm for Multi-objective Optimization”, *EMO*, pp. 127-139, 2001.

[9] G. David, “Sizing Populations for Serial and Parallel Genetic Algorithms”, in Proc. 3rd International Conference on Genetic Algorithms, San Mateo, California, pp. 70-79, 1989.

[10] G. David, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, Massachusetts.

[11] T. Mitchell, Machine Learning¸ International Edition, USA, McGrawHill, 1997.

[12] O. Andrzej¸ Multicriteria optimization for engineering design, Academic Press, 1985.

[13] Z. Eckart, Evolutionary Algorithms for Multiobjective Optomization: Methods and Applications, Ph.D. thesis, Shaker Verlag, Germany, 1999.