Solving Airport Gate Assignment Problem Using Genetic Algorithms Approach

Conference Paper · May 2015

DOI: 10.1109/ICAdLT.2015.7136615

3 authors:


Hammadi Ghazouani
Ecole Nationale des Sciences de l'Informatique


Moez Hammami


Ouajdi Korbaa
University of Sousse

Solving Airport Gate assignment problem using Genetic Algorithms approach

Hammadi GHAZOUANI
MARS/RU,
University of Monastir
Monastir, Tunisia
(hammadighazouani@gmail.com)

Moez HAMMAMI
COSMOS/ISG
University of Tunis
Tunis Tunisia
(moez.hammami@isg.rnu.tn)

Ouajdi KORBAA
MARS/RU,
University of Monastir
Monastir, Tunisia
(Ouajdi.Korbaa@Centraliens-Lille.org)

***Abstract*—**Because of the rapid growth of air traffic, optimizing airport management is becoming necessary in order to increase airport's capacity and better align its resources to the received traffic. In this paper we study the assignment of the available gates to the arriving aircrafts based on the fixed daily schedule. The gate assignment problem (GAP) addresses the issue of maximizing the use of gates equipped with aerobridges. We introduce a new approach based on Genetic Algorithms (GA) to solve the GAP. The encoding strategy consists in representing the chromosome by a vector of integers. The index of each gene represents the flight number and its value represents the number of the gate to which the flight will be assigned. The method used to generate the initial population is based on three different heuristics and a random sorting of the gates. The selection method is the "In fitness proportionate selection" known as "roulette wheel selection". In addition to one point and two point Crossover operators, we designed a Greedy procedure Crossover (GPX) operator. The experimentation is based on the use of fictive scenarios generated in accordance with the physical characteristics of Tunis Carthage airport and using different flight schedules. The comparison between deterministic approach, simple heuristics and the GA has shown the efficiency of the last approach in term of quality of the solution when we aim to solve the problems of large size. In order to determine the best configuration of the GA, we compared the different crossover operators and we noticed that the use of GPX improves the speed of convergence of the algorithm towards better solutions.

Keywords- ground optimization; gate assignment; meta-heuristics; Genetic Algorithms; Greedy Partition Crossover (GPX)

1. INTRODUCTION

In order to ensure an efficient management of an airport, the manager has to identify the critical resources and to optimize their utilization. Taking into account the objectives of ensuring the aircraft safety, reducing the costs and minimizing the delays, the optimization of gate assignment problem (GAP) can be considered as one of the principal problems that should be treated by the airport manager. In this paper we propose to study the GAP in order to maximize the use of gates equipped with aerobridges. After introducing a literature review, we present our approach based on Genetic Algorithms. In order to verify the effectiveness of the proposed approach and decide about the best configuration of the genetic algorithm, we simulate different test cases using probability rules and compare the proposed approach with previous work.

2. LITERATURE REVIEW

The gate assignment problem is one of the first ground problems studied in order to optimize the airport revenue and offer to each aircraft the maximum of services needed. It was shown to be NP-hard since 1979 [1]. According to the complexity of each studied problem and simplifying assumptions, solving approaches used in the literature vary from the deterministic approaches to the meta-heuristics. In this section we try to cover most the approaches used to solve the GAP.

2.1 Mathematical programming techniques: according to the objective function, the constraints and the assumptions taken in consideration, different mathematical formulations were used in the literature. One of the most common objective functions is the minimization of the passenger walking distance. R. A. Bihr [2] has proposed to deal with the minimization of the total passenger walking distance and introduced a conceptual solution using 0,1 linear programming model in which he treated the single assignment constraints only without taking into account operational constraints. G. vanderstraeten [3] treated the objective of maximizing the number of aircrafts assigned to services area while respecting operational constraints. The author introduced a mathematical model that was solved using a specialized heuristic based on the implicit enumeration. The solving approach does not consider delays or flight cancellations. A similar objective that consists on minimizing the intra-terminal travel was studied in [4] and the main contribution consists on the consideration of the costs of delays. There are also recent studies that treat real time gates reassignment problems such as the network model proposed in [5] that aims at minimizing the fuel burn cost of aircraft taxi by type and expect passenger discomfort for "tight" connections as a function of inter-gate distance and connection time. Another mathematical multi-objective model that aims at minimizing the number of unassigned aircrafts and maximizing the number of flights assigned to gates equipped with aerobridges was introduced in [6]. Despite the variety of mathematical models, the disadvantage of deterministic approaches is that they are ineffective in solving the problems of large sizes.

In [7], the authors describe the mathematical programming approaches used to solve the GAP. They choose to describe two specific approaches. The first one is based on quadratic assignment model and the second one is a multi-mode scheduling formulation.

2.2 Heuristics and meta-heuristics: with the inefficiency of the deterministic approaches while treating the problems of large sizes, many heuristics were developed. The heuristic proposed in [8] consists in assigning the aircraft with the larger passenger volume to the gate with a smaller average walking distance in order to minimize the total passenger walking distance. The heuristic approach to solve static and real time GAP proposed in [9] is used to study the impact of the flight delays on the initial solution obtained using Cplex solver. It consists in iterating the process of reassigning flights and revising penalty values to obtain a feasible solution. The two heuristics described in [6] are based on the way of sorting the airports gates. The first heuristic named FUGA consists in assigning the first arriving aircraft to the first unused gate in order to favor the use of a specific number of gates especially because of services that they offer. The second one named MUGA consists in assigning the first arriving aircraft to the most unused gate in order to equilibrate the use of different gates. The common disadvantage of heuristic approaches is that they are generally designed for specific instances of the GAP without giving a generic approach. Many meta-heuristics were developed and their efficiencies were tested under the considered assumptions. In [10], the comparison between four developed approaches based on genetic algorithm, simulated annealing, Tabu search and hybrid approach using Tabu search and simulated annealing showed that the hybrid approach is the most efficient in terms of solution quality and computational time. Then, it is Tabu search which has the best performance among the three meta-heuristics. The hybrid simulated annealing with Tabu search approach was also implemented in [11]. The authors dealt with a multi-objective problem in which they minimize the number of flights unassigned to gates and the total walking distance. Other approaches such as expert systems [12, 13] and knowledge based systems [14] were used to formulate and solve the GAP.

3. Problem formulation: deterministic approach

In this section we refer to [6] and we introduce some modification to the proposed mathematical formulation. The main modification is that we don't deal with a multi-objective problem. The objective function is to maximize the number of flights assigned to gates equipped with aerobridges. This choice is also justified in [7]. In fact, authors considered that transferring passengers from flights assigned to the apron to the terminal building using transfer busses may increase connection time and can hardly be regarded as desirable if our goal is to minimize total passenger walking distance and connection time.

Taking into account these modifications, the resulting mathematical model is as follows

- N: the number of flights to assign to airport gates during the assignment day;
- N_0 : The number of flights initially assigned;
- M: the number of airport gates.
- M1: the number of airport gates equipped with aerobridges.
- A_i : arriving time of flight number i to the airport gate.
- L_i : leaving time of the flight number i from its gate.
- $y0_{i,k} = 1$: if the aircraft performing flight number i is initially assigned to the gate number k, 0 otherwise.
- $A0_i$: arriving time of flight i for which a gate has been designated beforehand;
- $L0_i$: Scheduled departure time of flight i for which a gate has been designated beforehand;
- $typeA_i$: type of gate requested by the flight number i (1 if the aircraft performing the flight number i can use aerobridge gate, 0 otherwise);
- $typeP_i$: type of gate number i (1 if the gate is equipped with aerobridge, 0 otherwise);
- $C_{i,j} = 1$ If we cannot use gates i and j together, 0 otherwise. This type of constraints is common especially when aerobridges are used. Indeed, for each position of the aerobridge, a new gate is defined but the bridge can be used in only one position at a time;
- $wing_i$: Wingspan of the aircraft performing flight number i.
- $width_k$: Width of the gate k (it corresponds to the wingspan of the largest aircraft for which gate number k was created) ;
- safe_dist: Safety distance between gates. It gives some leeway to aircrafts moving from or to their gates.
- Decision variable $y_{i,k} = 1$ if the aircraft performing flight number i is assigned to gate number k, 0 otherwise;

The mathematical model that we propose to solve is represented by the following optimization formulation:

$$\text{Maximize } \sum_{i=1}^{N} \sum_{j=1}^{M} y_{i,j} * typeA_i * typeP_j \quad (1)$$

Under the constraints:

$$y_{i,k}. wing_i + 2 * safe_{dist} \leq width_k \forall\, 1 \leq i \leq N, \forall 1 \leq k \leq M; \quad (2)$$

$$y_{i,k}. typeP_k \leq y_{i,k}. typeA_i \;\; \forall\, 1 \leq i \leq\; N, \forall 1 \leq k \leq M; \quad (3)$$

$$\sum_{j=1}^{M} y_{i,j} = 1 \qquad \forall i, 1 \leq i \leq N; \quad (4)$$

$$C_{k1,k2}. y_{i,k1}. y_{j,k2}(L_j - A_i).(L_i - A_j) \leq 0 \; (\forall i, j, 1 \leq i, j \leq N, i <> j, 1 \leq k1, k2 \leq M) \quad (5)$$

$$C_{k1,k2}. y_{i,k1}. y0_{j,k2}.(L0_j - A_i).(L_i - A0_j) \leq 0 \; (\forall i, j, 1 \leq i, j \leq N_0, 1 \leq k1, k2 \leq M) \quad (6)$$

$$y_{i,k} \in \{0,1\} (\forall i, 1 \;\leq i \leq N, \forall k, 1 \;\leq k \leq M) \quad (7)$$

$$typeA_i \in \{0,1\}, typeP_k \in \{0,1\} \;\; (\forall i, 1 \;\leq i \leq N, \forall k, 1 \;\leq k \leq M) \quad (8)$$

$C_{i,j} \in \{0,1\}(\forall i, j 1 \leq i, j \leq M)$ (9)

$A_i, L_i \in [0,1440](\forall i \; 1 \leq i \leq N)$ (10)

4. Genetic algorithms Approach

4.1. General Genetic Algorithm

Genetic algorithms (GA) are part of evolutionary algorithms that became popular through the work of John Holland [15] who used this approach in the prisoner's Dilemma. The GA's principle is based on techniques inspired from natural evolution. A basic GA can be represented by the following graph.

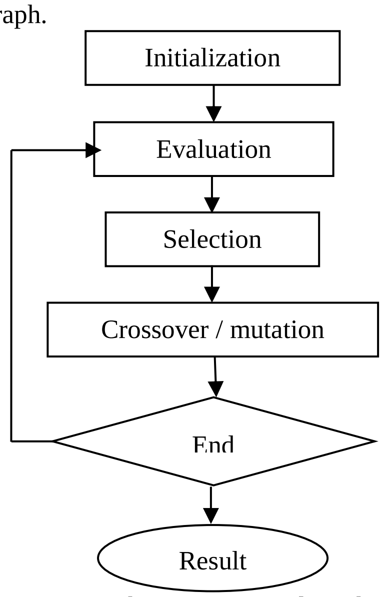

Fig 1: basic Genetic Algorithm

The initialization is to generate an initial population using, usually, a randomly generator based on simple techniques.

To pass from one generation to the next, the following operations are repeated:

- Couples of parents P1 and P2 are selected according to their fitness;
- The crossover operator is used to generate a couple of descendants C1 and C2 from P1 and P2. It is used with a probability Pc ;
- Other individuals are selected according to their fitness. The mutation operator is used to generate new chromosomes from the selected individuals with a probability Pm and new mutated individuals can be generated ;
- The Fitness of the new descendants and mutated individuals are then evaluated to decide about inserting them into the new generation ;

4.2. encoding approach

To introduce the data structure used to represent a possible solution or a chromosome and in order to minimize its size, we used two simple rules for formatting data. The first one is to sort all flights in the ascending order of their arrival times and to assign to each one a serial number. The second one consists in assigning to each gate a serial number.

Thus, a chromosome is represented by a vector of N integers (N being the number of flights). The index of each gene represents the flight number and its value represents the number of the gate to which the flight will be assigned.

In order to explain our encoding approach, we use the following example (table 1) which consists in assigning 6 flights to the 5 gates from which 3 are equipped with aerobridges (gates 1, 2 and 3). The representation is simplified and only the most significant properties are illustrated (tables 2 and 3).

5	1	2	3	4	2

Table 1: Example of chromosome

1			2			3			4			5		
code	width	aerobridge	Code	width	aerobridge	code	width	aerobridge	code	width	aerobridge	code	width	aerobridge
P1	100	Y	P2	100	Y	P3	100	Y	L1	100	N	L2	100	N

Table 2: Gate table

1			2			3			4			5			6		
AT156			LH132			TU514			TU526			TU716			TU542		
Arrival	Departure	Aircraft	arrival	Departure	Aircraft	Arrival	Departure	Aircraft	arrival	Departure	Aircraft	arrival	Departure	Aircraft	arrival	Departure	Aircraft
02h25	03h30	B737-800	02h30	03h50	A319	07h10	10h00	A320	07h25	09h00	B737-600	07h50	09h30	A320	07h55	10h00	B737-600

Table 3: Flight table

4.3. Fitness Function

GA aims at maximizing the number of flights assigned to the gates equipped with aerobridges. Let f_k be the fitness of individual number k.

$$f_k = \sum_{i=1}^{N} C(g_i)$$

Where g_i is the value of gene i. and $C(j) = 1$ if j represents a gate equipped with aerobridge, 0 otherwise.

4.4. Initial population

Before generating the initial population, we initialized the population size. In our case, we tested three different sizes. 5, 10 and 20 individuals for each generation.

The initialization operator uses three main heuristics to generate the initial population:

1. First unused gate assignment (FUGA) heuristic: The flights are sorted according to their arrival times and the gates are sorted according to their numbers (the first M1 gates are those equipped with aerobridges). For each new flight, we start by testing if the first gate can receive the considered flight. If a flight can't be assigned to gate number i, we consider the $(i+1)^{th}$ gate. The process is repeated for all the flights. In addition to favoring the

utilization of aerobridges. This approach defines airport manager's preferences to use some specific gates which are considered more beneficial;

2. Most Unused Gate Assignment (MUNGA) heuristic: in this case, the flights are also sorted using the same rule as before. But the gates are sorted in the ascending order of their occupancy frequency. For each new flight, the gates are reordered according to their occupancy frequency and those equipped with aerobridges are placed in the first M1 positions. The assignment process is the same as that used in FUGA.

3. Most used gate assignment (MUGA) heuristic: in this case, the flights are also sorted using the same rule as before. But the gates are sorted in the descending order of their occupancy frequency (number of flights accepted by each gate). For each new flight, the gates are reordered according to their occupancy frequency and those equipped with aerobridges are placed in the first M1 positions. The assignment process is the same as that used in FUGA.

Then, we use the previous heuristics with a random sorting of gates. Thus, new solutions can be reached at each new sorting of gates. For example, when using FUGA heuristic, the flights are always sorted according to their arrival times but the sorting order of gates changes.

4.5. Selection method

The selection method used in our GA is the "In fitness proportionate selection" known as "roulette wheel selection". The fitness function assigns a fitness value to solutions or chromosomes. This value of fitness is used to associate a probability of selection with each individual chromosome. If f_i is the fitness of individual i in the population, its probability of being selected is $p_i = \frac{f_i}{\sum_{i=1}^{S} f_i}$, where S is the number of individuals in the population. To implement this approach we use the uniform low to select a number x between zero and $\sum_{j=1}^{s} f_j$. If $\sum_{j=1}^{i-1} f_j < x < \sum_{j=1}^{i} f_j$, then the chromosome number i is selected.

4.6. Crossover operator

To improve the efficiency of GA approach we use the following crossover operators:

1- One point crossover: To perform this type of crossover on chromosomes consisting of N genes, we randomly select a common position to both parents. Then, we exchange the ends of the two chromosomes, producing two children C1 and C2;

2- Two point crossover: Two-point crossover calls for two points to be selected from the parents. All genes that are between the two points are swapped to generate new solutions;

3- Greedy procedure Crossover (GPX): we use the GPX approach introduced in [16]. The implementation of this approach is described as follows:

a. We consider the first parent. We chose genes representing the most used gate which is equipped with aerobridge and copy them to the first son ;
b. We erase correspondent genes from the two parents;
c. We consider the second parent and copy to the first son the genes representing the most used gate which is equipped with aerobridge;
d. We erase correspondent genes from the two parents;
e. We repeat previous steps while the chromosome in construction represents a feasible assignment and the number of iterations is minus than M1;
f. We assign the rest of genes by choosing them randomly from their parents. We verify at each assignment of a new gene that the chromosome encodes a feasible assignment ;
g. To obtain the second son, we restart the process and begin with the second parent.

There are other crossover operators that can be used in the literature. We choose these three operators because of their ability to keep a significant number of genes from parents. This quality is considered as an important parameter that facilitates convergence of our GA by choosing neighboring solutions.

4.7. Mutation

The mutation operator consists in selecting randomly one gene and replacing it by a random value between 1 and M. the probability of mutation P_m is fixed to P_m=1/N for each gene for each generation.

4.8. Evaluation

The evaluation step consists in calculating the fitness of the generated chromosomes to decide about their acceptance into the new generation. We assume that the new chromosome is automatically maintained into the new generation if its fitness is better than that of its parents, otherwise the new chromosome is maintained with a probability P_e=1/N.

It is important to notice that the crossover and mutation operators don't automatically generate feasible solutions. Thus, the assessment of the quality of generated chromosomes is performed using flights and gates properties. It is done during the crossover or mutation steps and can take a considerable time compared to the time spent in different steps of the GA.

4.9. Termination

In order to decide about the termination of generational process, three different terminating conditions were used:

- An optimal solution is reached: we consider that an optimal solution is reached when all the flights are assigned to gates equipped with aerobridges.
- No improvement in fitness function was noticed: we consider that fitness can't be improved if the best value is not changed after 5 generations;
- The maximum number of iterations is reached: the process is stopped after fifty iterations even if the first two conditions have not been met.

5. Experimentation and discussion

In order to verify the efficiency of the GA developed in this paper, we use the same approach used in [6] to generate and represent test cases. Each flight is represented using the following tuple: <flight_number, arrival_time, gate_occupation_duration, preferred_type_of_gate> when:

- The "flight_number" is an automatic sequential number;
- The "arrival_date" is an integer between 0 and 1440;
- we generate the set of "gate_occupation_duration" using normal distribution with a mean equal to 50 minutes and standard deviation equal to 20 minutes;
- The set of "Preferred_type_of_gate" is generated according to aircraft type or model.

The simulation approach is based on an instance of airport which is composed of 49 gates including 11 gates equipped with aerobridges.

The number of received flights varies between 10 and 300 flights per day. The number of flights during the rush hour varies between 5 and 35 flights per hour. The mathematical model was solved using Cplex 12.4.

In order to obtain the better GA algorithm configuration, we tested different implementations of the described GA:

1. Crossover is ensured using the one point and two point crossover operators only ;
2. Crossover is ensured using only GPX operator;
3. Crossover is ensured using a random selection of one operator from the listed ones at each generation;

We compared the results obtained by the three implementations of GA each other and with those obtained by the deterministic approach solved using CPLEX 12.4. Two main families of problems have been identified. The family of relatively simple problems which contains the problems generated by considering a balanced schedule that minimize the number of arriving flights during rush hours (Fig 2) and the family of the more complex problems for which rush hours are heavily fraught (Fig 3).

The experimental results were obtained using a personal computer with i5 (2.4 GHz) processor and RAM of 6 Go.

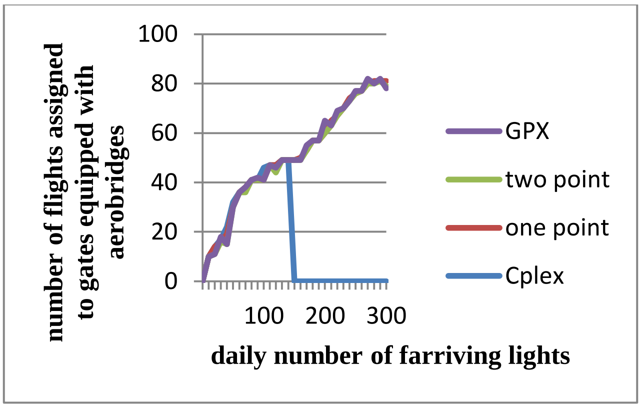

Fig 2: experimental results (simple scenarios: less than 15 flights during rush hour)

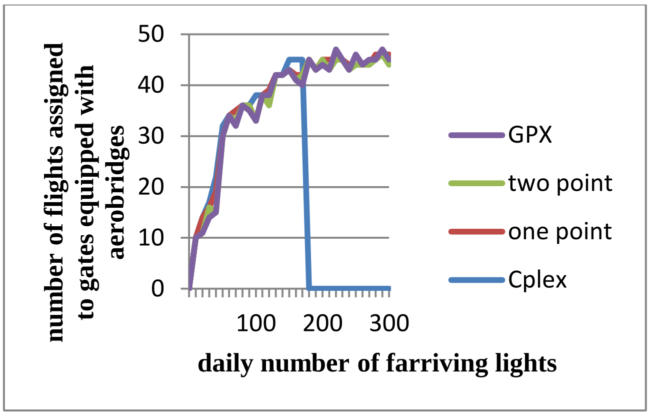

Fig 3: experimental results (complex scenarios: more than 30 flights during rush hour)

- For the relatively simple problems (less than 15 arriving flights during the rush hour), the deterministic approach generates optimal solutions for the problems of small size, but The GA generates different feasible solutions for all the problem sizes. For the populations of small size (less than 10 individuals) and with all the GA implementations, the termination criterion was the notification of non-improvement in the fitness function. For the populations of important size (about twenty individuals per generation), the termination criterion was the reaching of maximum number of iterations for the first two implementations and the non-improvement in the fitness function for GPX implementation;

- For more complex configurations (more than 25 arriving flights during the rush hours), the deterministic approach doesn't find any feasible solution because of lack of memory, but the GA based approach obtained feasible solutions. In term of fitness, the results obtained using different crossover operators were equivalent for the different implementations of GA. The termination criteria were the reaching of maximum number of iterations for the implementations with one point and two point crossover operators and the notification of non-improvement in the fitness function for the GPX implementation. The computation time was significantly greater for the implementations using one point and two point crossovers;

- For all the GA implementations the number of flights assigned to the gates equipped with aerobridge is greater when the rush hours are controlled. This indicates the importance of preparing the daily schedule and managing the arrival and departure times;

- The use of GPX allows a faster convergence to a better solution. If a solution generated using GPX is acceptable, it will have a better fitness because the new descendant is gradually generated by keeping the best assignments. It is also more interesting than other operators because the convergence using GPX is faster. This is explained by the fact that we can obtain two descendants which have fitness values better than fitness of their parents when we use GPX but with the use of other operators, there is only one descendant who can have a better fitness. The experimentation showed that the use of GPX becomes more efficient when the population size becomes more important.

6. CONCLUSION

In this paper, we treated the optimization of gate assignment problem. We develop a GA to solve the gate assignment problem. The described approach was implemented using specific parameters. In fact, three specific crossover operators were used. The selection of these operators was justified by their ability to let new descendants inherit an important number of genes from their parents. The first two operators are well known in the literature. The third one was inspired from GPX operator.

The used encoding approach which was based on the use of an array of N integers offers the facility of implementing different operators (crossover and mutation). It is also a simple formulation that can be easily understood. The main drawback of the used formulation is that it depends on the number of flights to be received. This number is not fixed and its increase can be annoying. Knowing that the number of flights is usually larger than the number of gates, another encoding approach can be used and consists in dealing with the number of gates. The chromosome will be an array of M columns. Each column contains flights that will be assigned to the correspondent gate. In this case, crossover and mutation operators should be reviewed.

The initialization process is based on three different heuristics. The use of simple heuristics guarantees the quality of initial population. This approach appears to be efficient, but it has some limitations especially with the problems of larger size. In this case the generation process becomes too slow. The wasted time is especially caused by the examination of the various constraints in order to test the feasibility of the solution. It should be noted that some constraints may be omitted depending on the configuration of the studied airport and the daily flight schedule. So, the computation time can be reduced considerably when we are studying real cases.

The use of GPX allows a faster convergence to a better solution. If a solution generated using GPX is acceptable, it will have a better fitness because the new descendant is gradually generated by keeping the best assignments. It is also more interesting than other operators because the convergence using GPX is faster. This is explained by the fact that we can obtain two descendants which have fitness values better than fitness of their parents when we use GPX but with the use of other operators, there is only one descendant that can have a better fitness. The experimentation showed that the use of GPX becomes more efficient when the population size becomes more important.

We dealt with the number of flights assigned to gates equipped with aerobridges because we consider that this objective can implicitly cover some objectives addressed in the literature such as minimizing passenger walking distances. However, some parameters such as the total number of passengers and the inoccupation time of gates were not treated. A study which aims at optimizing the gates idle time and takes into account the number of passenger and their destinations after leaving the aircraft can improve the operational procedures.

To validate the implemented solution, we used a simulation approach and we generated different scenarios. This approach helps to identify the airport capacity and to simulate critical situations. However, a study based on real data can be conducted to improve our approach.

7. REFERENCES

[1]. T. Obata. The quadratic assignment problem: evaluation of exact and heuristic algorithms. Technical Report TRS-7901, Rensselaer Polytechnic Institute, Troy, New York, 1979.

[2]. R. A. Bihr. A conceptual solution to the aircraft gate assignment problem using 0, 1 linear programming. Computers ind. Engng Vol. 19, N. 1-4, pp. 280-284, 1990.

[3]. G. vanderstraeten Automatic assignment of aircraft to gates at a terminal. Computers ind. Engng. Vol 14.N. 1, pp 15-25,1988.

[4]. S. Bandara and S.C. Wirasinghe. Walking distance minimization for airport terminal configurations. Transportation Research 26A:59–74, 1992.

[5]. B. Maharjan, T. I. Matis. An optimization model for gate reassignment in response to flight delays. Journal of Air Transport Management 17 pp 256-261, 2011.

[6]. H. Ghazouani, M. Hammami, and O. Korbaa. Airport Gate Assignment problem: Mathematical formulation and resolution. ROIS2013. International Conference on Reasoning and Optimization in Information Systems. Hammam sousse. 2013.

[7]. Ulrich Dorndorf, Andreas Drexl,Yury Nikulin, Erwin Pesch. Flight gate scheduling: State of the art and recent developments. Omega 35 pp 326 – 334. 2007.

[8]. H. Ding, A. Lim, B Rodrigues, Y. Zhu. The over constrained airport gate assignment problem. computer and operations research 32 pp 1867-1880, 2005.

[9]. C.H. Cheng, S. C. Ho and C.L.Kwan. The use of meta-heuristics for airport gate assignment. Expert Systems with Applications (2012), http://dx.doi.org/10.1016/j.eswa.2012.04.071.

[10]. J. Xu and G. Bailey. The airport gate assignment problem: Mathematical model and a tabu search algorithm. In Proceedings of the 34th Hawaii International Conference on System Sciences. pp 1–10. 2001.

[11]. B. Maharjan, T. I. Matis. Multi-commodity flow network model of the flight gate assignment problem. computer and industrial engineering 63 pp 1135-1144, 2012

[12]. K. Srihari. An expert system methodology for aircraft gate assignment. Comp ind. Engng. Vol 21.Nos 1-4, pp 101-105, 1991.

[13]. Y. Cheng. A knowledge based airport gate assignment system integrated with mathematical programming. Computers ind. Engng Vol. 32, No. 4. pp. 837-852, 1997

[14]. C. Li. airport gate assignment-A hybrid Model and Implementation. Computer Science Department, Texas Tech University. 2009.

[15]. John H. Holland Genetic Algorithms. http://www.econ.iastate.edu/tesfatsi/holland.GAIntro.htm

[16]. P. Galinier and J-K. Hao. Hybrid evolutionary algorithms for graph coloring. Journal of combinatorial optimization 3, 379-397,1999.