CIE44 & IMSS'14 Proceedings, 14-16October 2014, Istanbul / Turkey, Pages: 2167-2181

SOLVING THE AIRLINE CREW PAIRING PROBLEM USING GENETIC ALGORITHMS

Ahmad E. Elhabashy[1],M. Hamdy Elwany[2], M. Nashat Fors[2], and Yasmine Abouelseoud[3*]

[1]Grado Department of Industrial and Systems Engineering,
Virginia Tech, Blacksburg, VA, USA
habashy@vt.edu

[2]Production Engineering Department, Faculty of Engineering,
Alexandria University, Alexandria, Egypt
hamdy@elwany.com and nashatfors@gmail.com

[3]Engineering Mathematics Department, Faculty of Engineering,
Alexandria University, Alexandria, Egypt
yasmine.abouelseoud@gmail.com

ABSTRACT

The crew scheduling problem is especially critical to airlines industry as this industry is labor intensive, with more than one-third of the daily revenue being paid to its workforce. Thus, finding an optimal scheduling policy is an impelling need. This paper investigates the cockpit crew pairing problem, where a two-phase methodology was developed to solve it. In the first phase, valid pairings are generated and then a genetic algorithm is used to choose the optimum set of pairings that cover a set of flights. Our main goal is to experiment with different genetic operators in literature. Benchmark instances as well as a real-life case study are used in our testing phase.

Keywords: Airlines industry, Crew scheduling, Pairings, Genetic algorithms

1 INTRODUCTION

Airlines face many challenging planning problems, which are usually addressed sequentially. The first one is the *schedule design problem*, in which the flights to be flown during a given time period are determined. Next, the type of aircraft is assigned to each flight in the*fleet assignment problem*.This is followed by addressing the*maintenancerouting problem*, where specific aircrafts are chosen within each type for different flights according to maintenance-related considerations. Finally, the problem of *crew scheduling*is to be solved (Gopalakrishnan and Johnson[1]); which, according to Barnhart et al. [2] could be defined as the problem of assigning a group of workers (a crew) to a set of tasks (flights).

The airline crew scheduling problem has received a lot of attention due to the extremely large size that the problem could reach in real-life cases, with complex rules and safety regulations which need to be satisfied. More importantly, the substantially higher airline crew salaries compared to personnel salaries in other modes of transportation made the problem more attractive to researchers, who tailored special techniques to solve it (Gopalakrishnan and Johnson[1]).As reported by the Air Transport Association of America (ATA)[3], more than one-third of the revenueof each day is spent to pay the wages, benefits and payroll taxes of its workforce, and generally labor costs per employee are above average compared to other service industries.

Moreover, airline crews are not salaried, but are rather paid for the time that they spend flying, plus some added compensation for excess time spent on the ground between flights and during rest periods (Barnhart et al. [2]).The total cost for crews, including salaries, benefits, and expenses, is the second largest cost figure, after the cost of the fuel, for

* Corresponding Author

airlines. Unlike the fuel cost, a large portion of flight-crew expenses are controllable, and even a small percentage of savings in flight crew expenses through better scheduling translates into millions of dollars, which ultimately could determine the survival or demise of an airline.

This work is motivated by the fact that tourism in Egypt is considered to be one of the main sources of national income. Accordingly, if savings in expenses were to be applied for local airline companies, this will not only result in increasing its revenue, but will also have positive effects on the Egyptian economy.

The rest of the paper is organized as follows. In the next section, basic terms used to formally define the problem of interest are explained. The crew pairing problem is defined in Section 3, together with its mathematical formulation as a set partitioning problem. In Section 4, a literature review is provided summarizing contributions of researchers to solve the crew pairing problem. Section 5 includes the description of the pairings generation phase and the genetic algorithm implemented details. The results obtained for both bench-mark instances and a real-life case study are given in Section 6. Finally, Section 7 concludes the paper.

2 BASIC TERMINOLOGY

The following terms are used when addressing airline crew scheduling problems as defined by Gopalakrishnan and Johnson[1],Barnhart et al. [2],Bazargan [4],Theil[5], and Vance et al. [6].

- **Crew bases: T**he home station or city in which the crew actually lives.
- **Flight leg:** The trip of an aircraft, from take-off to landing. Itconsists of an origin station, a destination station, adeparture time, and an arrival time.
- **Flight:** A number of several flight legs combined together.
- **Duty periods:** A sequence of flights that can be flown by a single crew member over the course of a work day with sit connection(s) in between them; i.e., flights that are grouped together.
- **Sit connection:** A connection period during a duty is called a sit connection. This involves the waiting times, on the part of the crew, for changing planes onto their next leg of duty. Normally, airlines impose minimum and maximum sit connection times; typically maximum and minimum values are called ***MaxSit***and***MinSit*** and are between 10 minutes and 3 hours.
- **Elapsed time:** The number of minutes that elapse between the beginning of a duty and the end of the duty. The elapsed time includes a ***briefing period*** before the first leg of the duty, and a ***debriefing period*** after the last leg of the duty. It must be less than a maximum allowable value called***MaxElapse***.
- **Rest:** A connection between two duties is referred to as a rest period, which includes overnight connections (or layover). Usually, these rests are within permissible limits.
- **Pairings:** They refer to duties that are strung together. They arecrew trips spanning one or more work days separated by periods of rest. Pairings should start and end at the same crew base.
- **MaxDuties:** Maximum number of duty periods composing a legal pairing.
- **MinRest:** A minimum number of hours of rest between duties that must be allowed in a pairing.
- **Time Away From Base (TAFB)**: It is the number of minutes that elapse between the beginning of the pairing and the end of the pairing.
- **Fly:** The total number of hours of actual flying time. The maximum permissible value is called ***MaxFly***.
- **Roster:** A roster is a list or plan showing turns of duty or leave for individuals or groups in an organization. In airlines context, it represents a potential crew schedule

for a specific crew member during the planning periods of usually two or four weeks. It contains assigned flight duties in addition to pre-scheduled activities and days off.

- **Crew schedule:** An optimal set of individuals rosters flown by crew members.
- **Deadheading crews:** Flying crews as passengers on some of the airlines flights within their schedule in order to reposition them for future assignments.

Figure 1 (fromTheil[5]) illustrates the difference between flights, duties, pairings, and rosters for a given crew member. The figure shows a roster spanning six days including three pairings, a pre-scheduled activity (simulator activity), and a day-off. Pairing 2 is composed of two duties spanning two days requiring a hotel stay (H) between them as well as a transit (T) back to the crew base. Duty 1 is itself composed of three flights combined with rest periods, while Flight 3 is, in turn, composed of two flight legs.

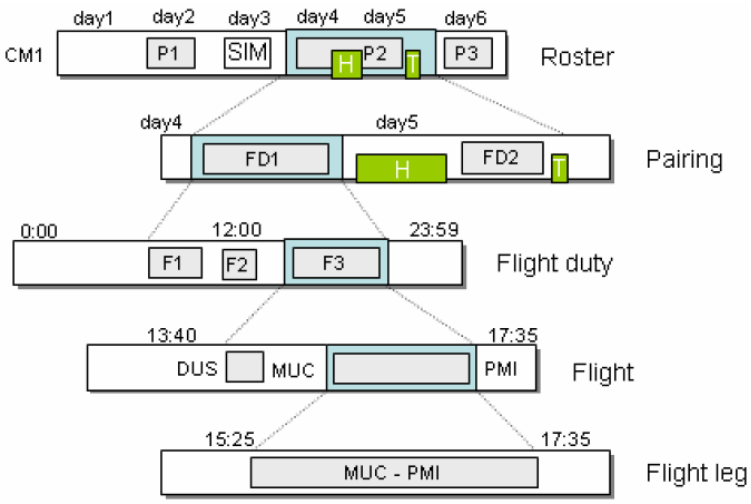

Figure 1: An Example of a Crew Roster and its Components [5]

3 THE CREW PAIRING PROBLEM

3.1 Outline

The aim of the crew scheduling problem is to identify optimum schedules of airline crews by taking the set of flights that need to be covered as input and yielding corresponding duties, pairings, and eventually monthly schedules as output; while taking into consideration the associated cost structure involved and various imposed constraints.

It should be noted that there are two types of airline crews with different scheduling approaches. First, there is the cockpit crew such as the captain and first officers, who tend to remain together for much of their schedule. More importantly, cockpit crews are not qualified to fly all fleet types, as this depends on the flying license they have acquired. On the other hand, there is the cabin crew such as flight attendants, who tend to vary more frequently and are not scheduled together but rather as individuals. Unlike cockpit crew, cabin crew members have no restrictions on fleet types making the size of their associated scheduling problem larger (Vance et al. [6]). This paper focuses on the cockpit crew scheduling problem only.

The crew scheduling problem is typically solved in two sequential phases, the *crew pairingphase* and the*crew rosteringphase*. This is mainly because the two problems are too large to address simultaneously (Bazargan[4]). The crew pairing problem is solved first with the scheduled flights as input, yielding optimum pairings. Subsequently, those pairings act as input to the crew rostering problem that in turn yields different crew rosters that build up the different crew members schedules.

3.2 The Problem and its Formulation

The solution to the crew pairing problem is a set of pairings such that each flight is covered by exactly one pairing and that the total crew cost is minimized as stated by Gopalakrishnan and Johnson[1] and Bazargan [4]. In order to accomplishthis goal, feasible (or legal) pairings have to be generated first, and the optimal subset of these pairing is determined, such that all flights are covered.

In the pairing generation stage, pairings are generated according to certain safety rules and workforce regulations, such as starting and ending at the same crew base and considering the values of *Maxduties*, *MinRest*, *MaxFly*, etc. Since all possible feasible pairings are generated in this stage, the number of generated pairings is usually huge according to Gopalakrishnan and Johnson[1], especially for large airlines. Hence, pairing generators could be coupled with extra rules and filters (heuristics) to choose potentially good pairings. The pairings are traditionally generated from valid duties using a depth-first-search approach as will be discussed later.

As for the optimization stage, a number of pairings is selected from the set of all available pairings (P) such that the selected pairings covers all available flights (F), while achieving the minimum possible associated cost; the mathematical formulation is as follows.

$$\text{Minimize} \sum_{j \in P} c_j x_j \qquad \textbf{(1)}$$

$$Subject\ to\ \sum_{j \in P} a_{ij} x_j = 1 \qquad \forall \text{ flight-legs } i \in F \qquad \textbf{(2)}$$

where $x_j = 1$, if pairing j is selected $(j = 1,2,3,\dots,P)$ and otherwise it is set equal to zero, c_j is the cost of the j^{th} pairing, and $a_{ij} = 1$ if flight i is covered by pairing j. The crew pairing problem is thus a 0-1 integer programming problem which could be seen as a Set Partitioning Problem (SPP); or a Set Covering Problem (SCP), if the set of constraints in (2) are relaxed with equations replaced by inequalities (Borndörfer[7]). In this latter case, deadheading occurs and thus additional cost terms should be added to the objective function. It is well known that these two problems are considered to be NP-Hard (Borndörfer[7]).

Airlines often add *crew-base balancing constraints* to ensure that the distribution ofwork over the set of crew-bases is matched to the crew resources;however, this constraint will not be taken into consideration when dealing with the crew pairing problem in the presentwork.

3.3 Computing the Crew Cost

The cost of crew assignment to a given flight is a complex computation. As stated before, crews do not receive fixed salaries, but they are paid for the time that they spend flying, in addition to some compensation for the excess time spent on the ground between flights and during rest periods. Given this, the 'cost' associated with an individual flight could be simply thought of as the duration of that flight.According to the work by Vance et al. [6] andSouai and Teghem[8], the cost of a duty period (b_d) expressed in hours is the maximum of three quantities: a guaranteed minimum number of hours, mg_1; a fraction, f_1, times the elapsed time of the duty; and the actual flying time in the duty period. The cost could be expressed as:

$$b_d = max\{mg_1, f_1 \times elapse, fly\} \qquad \textbf{(3)}$$

Typical values of the parameters that define legal duty periods and their costs for a domestic carrier, as suggested by Vance et al. [6] and Souai and Teghem[8], are *MaxSit* = 4 hours, *MinSit* = 0.5 hours, *MaxElapse* = 12 hours, *MaxFly* = 8 hours, mg_1 = 3 hours, and f_1 = 4/7.

The cost of a pairing *p* in hours (c_p) is the maximum of three quantities: a minimum guarantee, mg_2, times the number of duties in the pairing (NDP); a fraction, f_2, times the

total elapsed time of the pairing, i.e., the time away from base (TAFB); and the sum of the costs of the individual duties that make up the pairing. This cost could be expressed as:

$$c_p = max\{NDP \times mg_2, f_2 \times TAFB, \sum_{d \in j} b_d\} \quad \textbf{(4)}$$

Typical values here (also suggested byVance et al. [6] and Souai and Teghem[8]) are *Maxduties* = 3, mg_2 = 4.75, and f_2 = 2/7. Because of the cost structure for duties and pairings, a lower bound on the cost of a given schedule is the total number of hours of flying in the schedule. Penalties are incurred by pairings that have high TAFB relative to the number of hours flown and pairings that have few hours of flying per duty.

4 RELATED WORK

Since the optimization phase of the crew pairing problem has been represented as a set partitioning (or covering) problem, according to Theil[5], it could be solved by *mathematical programming techniques*, *network-based models*, or *meta-heuristics*.

Mathematical programming techniques are typically those used to solve binary integer programs such as *branch-and-bound, branch-and-cut,* or *column generation* methods. *Branch-and-cut* methods, as mentioned by Gopalakrishnan and Johnson[1],consisting of a combination of a cutting plane method with a branch-and-bound algorithmhave been successfully applied to the crew pairing problem as demonstrated in the work by Hoffman and Padberg[9]. In the *column generation* technique, an initial subset of pairings is generated and the resulting restricted problem is solvedas suggested by Vance et al. [10]. Subsequently, dual prices are used to determine which pairings need to be added in the solution pool to improve the solution until no improvements can be achieved. Finally, there is the *branch-and-price* method, which applies the column generation technique on the branch-and-bound tree; it has been used by Shenoi[11] and good results to relatively large problem instances have been reported.

The crew pairing problem could also be formulated as a network flow problem. There are usually two types in literature, *trip-as-node* and *trip-as-arc* networks, where the term *trip* usually stands for a flight leg, flight duty, or pairing (Theil[5]). With such a network representation, generation of legal pairings is done simply by generating paths that start and end at a crew base passing through subsequent flight legs (Gopalakrishnan and Johnson[1]). An example for a solution approach using a trip-as-node network representation is that by Yan and Chang [12], which incorporated column generation with the sub-problems solved using shortest path algorithms.

Lastly, meta-heuristics are also used to address this type of problem. Meta-heuristics are characterized by being approaches mimicking naturally occurring processes; for instance, *Ant Colony Optimization* and *Genetic Algorithms* have been successfully applied to solve the crew pairing problem. On one hand, Ant Colony Optimization (ACO) is based on the cooperative behaviour of real ants, which are able to find the shortest path from their nest to a food source (Rao [13]).ACO has been applied in the work by Crawford et al. [14], where the problem was solved using hybridizations of ACO with constraint programming and optimal solutions for a number of benchmark instances were reached. Another ACO-based approach was presented by Deng and Lin [15], where a trip-as-node network representation was used along with ACO to solve the crew scheduling problem.

On the other hand, Genetic Algorithm (GA), first introduced by Holland[†],is based on the principles of natural genetics and natural selection. The basic elements of natural genetics – reproduction, crossover, and mutation – are used in the geneticsearch procedure (Rao [13]).A GA-based heuristic for the SPP was reported as early as 1995 by Chu and

[†] This is in reference to J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press,Ann Arbor, MI, 1975.

Beasley[16].This heuristic was able to solve, to optimality or near optimality, 55 real-life problems from the ones that appearedin the work of Hoffman and Padberg[9]. However, it was not computationally competitive with exact solution techniques, except in perhaps problems when the gap between the LP relaxation solution and the optimal integer-valued solution is large.The same authors developed a similar GA-based heuristic in[17], but this time it was for theSCP; and had also made slight enhancements to their previous work for the SPP in[18], but not with much effect on the results. In addition, they gathered test problems from a number of sources (as Hoffman and Padberg[9]and Levine [19]) and made them available to researchers online[20]. Another similar approach is the work by Levine [19] which uses a GA along with a local search heuristic that helps in finding feasible or near-feasible solutions. Kornilakis and Stamatopoulos[21] proposed a two-phase procedure; in the first phase - the pairing generation phase, a depth first search approach was employed, while the second phase dealt with the selection of a subset of the generated pairings with a near optimal cost using GA. Finally, another GA approach was suggested by Zeren and Özkol[22], which was an extension of the work by Kornilakis and Stamatopoulos[21] and Beasley and Chu [17]; in which a new operator was proposed, and faster convergence with less computational time needed was reported.

5 THE PROPOSED APPROACH

This work proposesa two-phase methodology to address both sequential stages of the airline cockpit crew scheduling problem. For the first stage, the pairing generation phase, the developed methodology is inspired from the enumeration approach presented by Kornilakis and Stamatopoulos[21]. As for the second phase, the optimization phase, the developed approach in the present work is based on a GA that attempts to combine the strength points of the different GA-based approaches in related literature.

There are a number of reasons for using GA to solve the optimization phase. Firstly, there would be no need for solving the LP relaxation as in exact methods, and it allows more flexibility in handling the problem. Secondly, the previous GA-related approaches showed promising results. More importantly, GAs would be able to provide a population of possible solutions at any iteration instead of a single optimum. Finally, exact methods may not be able to provide optimal solutions for large problem instances.

5.1 The Pairing Generation Phase

In this phase, a Breadth-First Search (BFS) enumeration algorithm has been used. It starts by generating all duties of length one, and then those of length two and so on. As shown in Figure 2, the algorithm begins by generating all valid single-legged duties from all available flight legs, and then seeks to find whether additional flight legs could be added to each of these duties such that various constraints are satisfied. Finally, subsets of these duties are connected to generate all possible valid crew pairings.

5.2 The Optimization Phase

The GA developed in this paper has been inspired by most of the previous work done in this field using GAs, and it attempts to combine the advantages of most of them while avoiding their shortcomings. The basic steps in any genetic algorithm are the generation of an initial population of chromosomes representing candidate solutions of the problem and then reproducing a number of generations by applying a group of different genetic operators (such as parent selection, crossover, and mutation) in order to improve the characteristics of the current population and make it become closer to the optimum solution; i.e. converges to optimality; the following sub-sectionsdescribe the detailed steps incorporated in this phase.

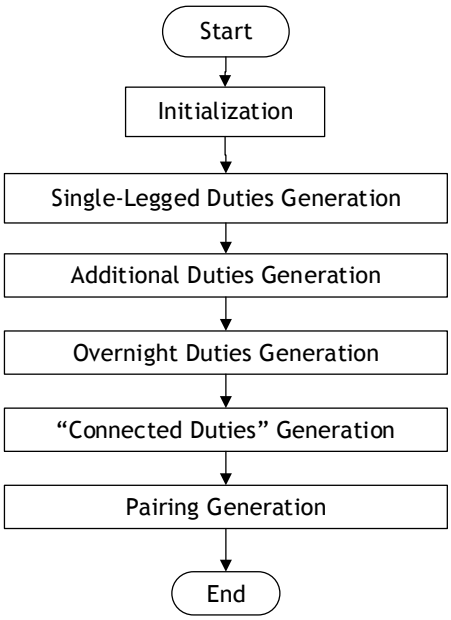

Figure 2: Overview of BFS Pairing Generation Algorithm

5.2.1 Setup Phase

This is a preliminary step that sets the various algorithm parameters:

- The population size (N) is 20 chromosomes.
- The maximum number of iterations to be performed in the algorithm is 1,500 iterations.
- The basic flight data is fed as input from an external file.

5.2.2 Chromosome Encoding

Another important step that precedes the generation of the initial population is deciding how the solution is to be represented, i.e., the suitable chromosome encoding method. Column-based representation was used where a chromosome is composed of a number of genes equal to P (number of pairings), and each gene has a binary value representing whether this column (pairing) is included in this solution or not, as shown in Figure 3.

column (gene)	1	2	3	4	…	P
bit string	1	0	1	1	…	0

Figure 3:Column-Based Chromosome Representation

5.2.3 Initial Population Generation

The initial population represents the original pool of chromosomes on which genetic operators are performed to enhance the solution. It is very important to create an initial population that is as diverse as possible for the GA to explore the solution space, and then allowing it to determine the most promising ones.

The approach adopted is based on that of Zeren and Özkol[22], which is a flight-based randomized approach rather than just setting the genes arbitrarily to ones or zeroes. The main goal of this approach is to generate an initial population consisting of feasible (or close to feasible) chromosomes and, additionally, to avoid flights being covered by more than one pairing. This is accomplished by not including in the search pairings for a given flight in case this flight is already covered in the current solution. The algorithm begins by selecting the first flight from all available flights set, and then choosing a random pairing. If this pairing actually covers this flight, a check is made to ensure that this pairing does not cause any

other flight to be covered by more than one pairing before adding it to this solution. If over-covering occurs, another random pairing is chosen and this step is repeated again for a fixed number of times. This whole process is repeated for all remaining uncovered flights until all of them have been covered by pairings.

5.2.4 Feasibility Operator Implementation

With the flight-based method of initial population generation, all flights are covered by at most one pairing and there is no guarantee that the generated chromosomes are valid; that is, some flights may be under-covered. Hence, an operator was needed that attempts to make all infeasible chromosomes feasible or at least makes them less infeasible; not only will this operator be applied to the newly generated population of chromosomes, but also to any other resulting chromosome(s) that will be generated from the application of the different genetic operators and may be infeasible.

This operator starts by checking the status of each chromosome by counting the number of covered flights, and checking the flights that are covered more than once. If the chromosome is under-covered, an "ADD" procedure is applied where more pairings are added to the chromosome to cover any remaining uncovered flights. This procedure is repeated for the same chromosome until it contains no more uncovered flights. If the chromosome is over-covered, a "DROP" procedure is applied that removes as much redundant pairings as possible to make any given flight covered by the minimum number of pairings possible if not just one, this is carried out without making this chromosome under-covered. If the chromosome is neither under-covered nor over-covered, the feasibility operator function is terminated.

5.2.5 Fitness Function Evaluation

The fitness of all the chromosomes in the population reflects "how good" a chromosome is with regard to solving the problem; this is done using a suitable fitness function. There are a lot of suggestions for dealing with the fitness function to make it a better estimate; however, the present work uses theoriginal objective function as in Equation (1), without adding anything to it, since deadheading would not be allowed, and no penalties would be used.

5.2.6 Genetic Operators

Different operators are applied in an iterative manner in search for the optimum solution, where each newiteration is known as a generation.

5.2.6.1 Parent Selection

Two chromosomes are chosen according to a certain criterion (such as their fitness) to be the parent chromosomes, which will then have the genetic operators applied to them, in order to produce a new offspring. The most commonly used selection method is the binary tournament method (Zeren and Özkol[22]). In this method, two chromosomes are chosen randomly, and the one having a better fitness value would be the first parent. The second parent is selected in a similar manner. This method was chosen because it does not require many calculations and runs fast. Moreover, it is less likely to get trapped in a local minimum compared to other selection methods such as the roulette wheel method.

5.2.6.2 Crossover Operator Implementation

After the parents have been selected, the crossover operator is implemented. Its objective is to create a new offspring by exchanging the information among chromosomes in the current population; this is usually done by exchanging some portions of the parents' chromosomes. The resulting offspring will have characteristics from both parents and if the parents are of high fitness themselves, then the offspring is expected to have the same or even a better fitness value.

Two crossover operators were implemented in this work. The first one is a single-point crossover operator, where a random crossover site for the parent chromosome is selected as applied by Mohan and Ozdemir [23], see Figure 4 (from Rao [13]), where the random site is the third gene in this case, and genes to the right of this site are swapped between the two parent chromosomes creating two new offspring chromosomes.

Parent 1:	0	1	0	1	0	1	1	0	1	1
Parent 2:	1	0	0	0	1	1	1	1	0	0
Offspring 1:	0	1	0	0	1	1	1	1	0	0
Offspring 2:	1	0	0	1	0	1	1	0	1	1

Figure 4: Example of Applying a Single-Point Crossover Operator at the Third Gene

The second crossover operator to be applied is the fusion crossover operator; which uses fitness values of parent chromosomes in a probabilistic way to select genes during the crossover operation. According to Beasley and Chu [17], the underlying principle behind this operator is that the choice of which of the two parent chromosome genes are passed to the offspring chromosome should be made based on their relative fitness values; as it is assumed that the inheritance of a particular gene from a *more* fit parent is likely to contribute *more* to the offspring's overall fitness than that from a less fit parent.It is applied in the following manner, gene values which are identical in both parent chromosomes are just copied to the offspring chromosomes; otherwise genes are copied from the fittest parent chromosome with the probability (1 – *p*) in Equation 5, and from the less fit parent chromosome with probability (*p*).

$$p = \frac{Fittness\ value\ of\ the\ more\ fit\ parent\ chromosome}{Sum\ of\ fitness\ values\ of\ both\ parent\ chromosomes} \quad \textbf{(5)}$$

The fusion crossover operator in particular was chosen for implementation due to its capability of generating new chromosomes when the parent chromosomes have a similar structure (Beasley and Chu [17]). While the one-point crossover operator was selected to further increase the population diversity.

5.2.6.3 Mutation Operator Implementation

The mutation operator is usually applied with a specific probability. It flips the value of a randomly chosen gene from 1 to 0 and vice versa. The main reason for applying this operator is to avoid getting stuck in a local optimum solution without ever reaching the global optimum by introducing new material into the population. In other words, its aim is to maintain the diversity of the solution, and to obtain chromosomes that could not be generated otherwise.

A range of mutation operators were suggested for implementation in related literature that are typically divided into static or dynamic mutation operators. The static type mutates a fixed number of genes through all iterations, while the dynamic type mutates a varying number of genes in each iteration. In this paper, three mutationoperators are implemented; all of them could be considered to be dynamic.

The first mutation operator implemented (suggested by Levine [19]) randomly selects either of the two parent chromosomes to apply mutation on. For each gene in this selected chromosome, mutation is applied to it with a probability equal to the reciprocal of the total number of genes.

While the second mutation operator implemented (suggested by Kornilakis and Stamatopoulos[21]) depends on the density of genes equal to 1 in the fittest chromosome in the population. Mutation here is applied on one of the child chromosomes in which a number of genes (randomly pre-specified) are mutated to 1 with a probability equal to the

percentages of 1's in the fittest chromosome(s), and vice versa.Those two mutation operators were chosen because the latter is based on the fittest chromosome(s) and will most likely be able to produce an offspring with a better fitness value, while the former ischosen to further increase the population diversity.

In addition to those two, another mutation operator was implemented based on an idea similar to that of the perturbation operator discussed by Zeren and Özkol[22], which makes a number of feasible chromosomes infeasible on purpose in an attempt to improve the solution.This new mutation operator generates a new chromosome that is the same as the fittest chromosome in the population but with lesser number of genes equal to 1 in it; the choice of which genes in the fittest chromosomes that are flipped to zero is randomly made. It is applied only with a certain probability, only 25% of the time. The resulting chromosome could be on one hand feasible, and then it would be just added to the population. But on the other hand, if not feasible, it is not added to the population unless it has a fitness function value less than that of the fittest chromosome without applying the feasibility operator, even if it is over or under-covered.

5.2.7 Sorting and Solution Evaluation

The fitness function values for all new offspring are then calculated, and the population of the current chromosomes, including the original ones and the new offspring, is ranked in an ascending order according to their fitness function values with the chromosome having least fitness function value in the first position. The first feasible chromosome (neither over-covered nor under-covered) would be this iteration's best solution; from which the global optimum would be easily identified.It is noteworthy that all the generated offspring chromosomes from all operators are added to the current population causing its size to increase. The population will be restored to its original size using a suitable replacement strategy.

5.2.8 Replacement Strategy

There are two common replacement strategies used in literature. The first one is called the "Generational" method in which the whole population is replaced by the new offspring chromosomes; but in this way the best chromosome(s) may not survive in the new generation, and the new generation may not necessarily be as fit as the previous one. The other method replaces the less fit chromosomes with the new offspring chromosomes, allowing the fittest chromosomes to be present in the new generation; this method is called the "Steady-State" method and is the strategy adopted by Beasley and Chu [17],Levine [19],Kornilakis and Stamatopoulos[21], and Zeren and Özkol[22].

Hence, it was opted to use latter type so as to keep most fit chromosomes within the new population; more specifically, an "Elitist" strategy was adopted in this work. As suggested by Zeren and Özkol[22], the "Elitist" strategy ranks the chromosomes in an ascending order according to their fitness values such that the fittest chromosome is first, and then only keeps the first N (20) chromosomes for the next generation. In this way, the new offspring generated by the different operators could then fit within the new generation according to their fitness values, and chromosomes with the lowest fitness values would be removed.

5.2.9 Algorithm Overview

The previous sub-sections illustrated the proposed GA with its basic components; a concise overview is presented in Figure 5, where each step represents a function (subroutine) that has been implemented in the proposed approach.

The first step is to generate an initial population of chromosomes where each chromosome is unique and each flight is covered by at most one pairing. After that, the feasibility operator together with the fitness function evaluation is applied to each chromosome. Then, it is the turn for the backbone of the GA, which is the application of the different genetic operators

to the current population to obtain new offspring chromosomes including parent selection, crossover operators, and mutation operators, respectively. For each of those offspring chromosomes, the feasibility operator is applied, no repeated chromosomes are allowed, and their fitness values are calculated. Finally, chromosome sorting to allow the application of the selected population replacement strategy to obtain a new generation is carried out. This whole process starting from the application of the genetic operators is repeated for a pre-specified number of iterations.

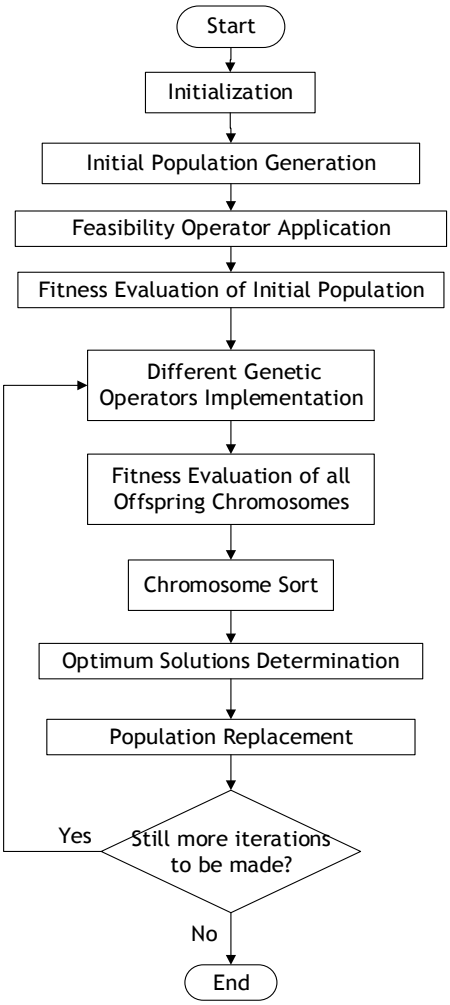

Figure 5: Overview of the Pairing Optimization Phase Algorithm

6 TESTING AND IMPLEMENTATION OF THE PROPOSED METHODOLOGY

6.1 Benchmark Test Instances

After the methodology to be used has been established, the next step is to test it. For the pairing generation phase, three different problems from literature were solved and almost identical pairings are generated using the proposed breadth-search methodology to those reported. For the pairing optimization phase, the SPP bench-mark instances presented by Chu and Beasley[18]were used. The majority of those instances have been solved to optimality by Chu and Beasley[18], but were originally contributed by Hoffman and Padberg[9] as real-life problems from different airlines; those instances are available online as text files from[20].

Due to the randomness present in the proposed GA, the algorithm has been applied 5 times to allow for a different set of random numbers generated in each trial. The details of those trials are shown in Table 1, together with the instance size and its optimal solution (as

reported by Chu and Beasley[18] and Levine [19]) with "o" representing an optimal value being obtained.

Table 1: Results of the Proposed GA on the Test Instances

Instance Name	Flights Count	Pairings Count	Optimal Solution	Best GA solution in each of the 5 trials				
nw41	17	197	*11,307*	o	o	o	o	o
nw32	19	294	*14,877*	o	o	o	o	o
nw40	19	404	*10,809*	o	o	o	o	o
nw08	24	434	*35,894*	o	o	o	o	o
nw15	31	467	*67,743*	o	o	o	o	o
nw21	25	577	*7,408*	o	o	o	o	o
nw22	23	619	*6,984*	o	o	o	o	o
nw12	27	626	*14,118*	o	14,318	14,318	14,318	o
nw39	25	677	*10,080*	o	o	o	o	o
nw20	22	685	*16,812*	o	o	17,058	o	o
nw23	19	711	*12,534*	o	o	o	o	o
nw37	19	770	*10,068*	o	o	o	o	o
nw26	23	771	*6,796*	o	o	o	6,842	o
nw10	24	853	*68,271*	o	o	68,382	o	o
nw34	20	899	*10,488*	o	o	o	o	o
nw43	18	1,072	*8,904*	o	o	o	o	o
nw42	23	1,079	*7,656*	o	o	7,666	o	7,666
nw28	18	1210	*8,298*	o	o	o	o	o
nw25	20	1217	*5,960*	o	o	o	o	o
nw38	23	1220	*5,558*	5,592	o	o	o	5,592
nw27	22	1,355	*9,933*	o	o	o	o	o
nw24	19	1,366	*6,314*	6,514	o	6,432	o	o
nw35	23	1,709	*7,216*	o	o	o	o	o
nw36	20	1,783	*7,314*	o	o	o	o	o
nw29	18	2,540	*4,274*	o	4,324	4,324	4,324	4,324
nw30	26	2,653	*3,942*	4,108	4,108	4,108	o	o
nw31	26	2,662	*8,038*	o	o	o	o	o
nw19	40	2,879	*10,898*	11,434	11,434	11,434	11,124	11,700
nw33	23	3,068	*6,678*	6,682	6,682	o	6,682	o
nw09	40	3103	*67,760*	68,522	68,522	68,522	o	68,648

Out of the 30 instances used for testing, 29 of them (96.67%) were solved to optimality with the instance "nw19" as the only exception; other than this one, all the other reached optimality at least twice (except for instances "nw29" and "nw09" reaching optimality only once). Table 2 shows that for the non-solved instance, the maximum deviation from optimal value is just 7.36% (occurred in one run only, whereas in all of the other four the deviation was only 2.07%); whereas for the instances yielding an optimal solution only once, the maximum deviation from optimum value did not exceed 1.31%.

Table 2: Range of Deviation from Optimal Values for Certain Instances

Instance Name	Flights Count	Pairings Count	Min Deviation	Max Deviation
nw19	40	2,879	2.07%	7.36%
nw29	18	2,540	1.17%	1.17%
nw09	40	3103	1.12%	1.31%

6.2 Real-life Case Study

The validated methodologies were then applied in a local airline company in Egypt whose main crew base is at Cairo International Airport, and serves a number of different

destinations such as Luxor, Sharm El-sheikh, Aswan, and Hurghada among others. The methodologies were applied on the available data,which was the second week of March, 2011 and the results were compared with those of the real-life situation. At the beginning, the time span of a pairing was set to one day and then pairings spanning the whole week were considered.

First, Saturday of this week was considered; which involved 37 flights. Using the pairing generation algorithm, 52 pairings were generated on this day in accordance with national regulations (found at [24]) and international ones. The next step before seeking their optimization was to calculate their corresponding costs according to Equations (3) and (4). Finally, when the proposed GA was applied, two different pairings were found to be optimal with a corresponding objective function value of 64.45, which is 14.20% less than the objective function value of the pairing used by the company with the value of 73.6.

After that, pairings were generated every day in the second week of March 2011, in the same manner, and compared with the company's strategy at the time; a summary of the results obtained is given in Table 3. The proposed GA was superior to the current company strategies, with improvements in the results starting from 7.64% on certain days and reaching up to 26.81% on other days.Furthermore, the solution provided by the proposed GA exhibited another advantage, which is the ability of offering the company an alternative pairing combination with the same cost (when applicable as in the case of Saturday), which would, in turn, provide the airline company more flexibility during scheduling.

Table 3: Real-life Case Study Pairing Optimization Daily Results

Day	Proposed GA Objective Function Value	Company Objective Function Value	Percentage Savings
Saturday	64.45	73.6	14.20%
Sunday	60.26	69.21	14.85%
Monday	50.83	64.11	26.13%
Tuesday	55.66	59.91	7.64%
Wednesday	46.18	54.61	18.25%
Thursday	55.66	59.91	7.64%
Friday	46.81	59.36	26.81%

Although the company only generates daily pairings, the results just presented could be further improved if the week is considered as a whole, especially since the option of overnight stays is allowed in the company. In fact, dealing with the company on a weekly basis would be better for this company, as weekdays flights are not actually repeated the next week, i.e. flights on the second Saturday for instance are not identical to those of the first or third Saturdays of the same month. This will also allow the company to work on a planning horizon of one week for the crew pairing problem, which would be more appropriate in case of the non-fixed weekly flights and more suitable to the flying personnel themselves, as well.

Consequently, the proposed methodology with its two sequential phases was applied for all week flights as a whole and a total of 342 pairings were generated, and an objective function value of 380.57 was obtained by the proposed GA after 10 trials. For the sake of comparison, a value representing the company's solution to the whole week crew pairing problem was needed; this was simply the summation of the values of daily solutions of all days of this week. The solution provided by the proposed GA is better than the company's solution achieving a 15.80% cost reduction.

7 CONCLUSION AND RECOMMENDATIONS

In this work, a solution methodology for the airline cockpit crew pairing problem has been presented with the main objective of successfully turning scheduled flights within a certain

time horizon into valid pairings and then selecting the pairings having least possible cost while covering all available flights. Hence, a two-phase methodology is developed. In the first phase, valid pairings are generated through a breadth-first-search pairing generation algorithm, and then the second phase seeks to optimize the pairings selection through a GA-based optimization technique.

For the first stage, the pairing generation algorithm presented accurate results for all cases it is applied to. As for the second phase, the results obtained by the proposed GA for the pairing optimization phase are quite satisfactory. The GA developed attempted to find a good mix of the various genetic operators available in the literature. The GA has been applied on thirty benchmark instances for five random trials, and the optimal solution was obtained for 96.67% of them at least once, while for the one not reaching optimal value, the maximum deviation ranges from 2.07% to 7.36% only.

Additionally, the effectiveness of the GA-based developed methodology was tested on a real-life case study in one of Egypt's local airline companies. Using available flight data from the company, both phases are applied to it successfully. When comparing the results obtained by the proposed methodology to those obtained by the current company's methodology for every single day within a specific week, the proposed methodology was superior.Additionally, both methodologies were applied to the whole week – a planning horizon which is more suitable to the company's nature – and the resultswere satisfactory and promising.

Finally, it could be concluded that the two-phase methodology presented in this study provides near optimal solutions for the crew pairing problem. Furthermore, it has the advantage of giving more than one alternative to the optimum solution, which would give airline companies more flexibility in scheduling.

8 REFERENCES

[1] **Gopalakrishnan, B. and Johnson, E. L.2005.** Airline Crew Scheduling: State-of-the-Art, *Annals of Operations Research*, vol. 140, pp. 305-337.

[2] **Barnhart, C. et al.** 2003. Airline Crew Scheduling, in *Handbook of Transportation Science*, 2nd Edition., Kluwer Academic Publishers, ch. 14, pp. 517-560.

[3] **Air Transport Association** 2009. Airline Handbook. [Last Accessed 8 Novmeber 2010] http://www.airlines.org/ATAResources/Handbook/Pages/AirlineHandbookChapter4AirlineEconomics.aspx

[4] **Bazargan, M.** 2010. Crew Scheduling, in *Airline Operations and Scheduling*, 2nd Edition, Ashgate Publishing Limited, ch. Six, pp. 83-102.

[5] **Theil M. P. 2005.** Team-oriented Airline Crew Scheduling and Rostering: Problem Description, Solution Approaches, and Decision Support, Paderborn University, Paderborn, D.Sc. Thesis.

[6] **Vance, P. H. et al. 1997**. Airline Crew Scheduling: A New Formulation and Decomposition Algorithm, *Operations Research*, vol. 45, no. 2, pp. 188-200.

[7] **Borndörfer, R. 1998.** Aspects of Set Packing, Partitioning, and Covering, Berlin University, Beriln, D.Sc. Thesis.

[8] **Souai, N. and Teghem, J. 2009.** Genetic algorithm based approach for the integrated airline crew-pairing and rostering problem, *European Journal of Operational Research*, vol. 199, no. 3, pp. 674–683.

[9] **Hoffman, K. L. and Padberg, M. 1993.** Solving Airline Crew Scheduling Problems by Branch-and-Cut, *Management Science*, vol. 39, no. 6, pp. 657-682.

[10] **Vance, P. H. et al. 1997.** A Heuristic Branch-and-Price Approach for the Airline Crew Pairing Problem, Auburn University, Working Paper.

[11] **Shenoi, R. G. 1994.** Solving the Long Haul Crew Pairing Problem, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Massachusetts, M.Sc. Thesis.

[12] **Yan, S. and Chang, J. 2002.** Airline Cockpit Crew Scheduling, *European Journal of Operational Research*, vol. 136, no. 3, pp. 501-511.

[13] **Rao, S. S.** 2009. Modern Methods of Optimization, in *Engineering Optimization, Theory and Practice*, 4th Edition, John Wiley & Sons, Inc., ch. 13, pp. 693 - 736.

[14] **Crawford, B. et al. 2006.** A Constructive Hybrid Algorithm for Crew Pairing Optimization, in *Lecture Notes in Artificial Intelligence (LNAI)*, Springer-Verlag Berlin Heidelberg, 2006, vol. 4183, pp. 45 - 55.

[15] **Deng, G. and Lin, W. 2011.** Ant Colony Optimization-based Algorithm for Airline Crew Scheduling Problem, *Expert Systems with Applications*, vol. 38, no. 5, pp. 5787 – 5793.

[16] **Chu, P. C. and Beasley, J. E. 1995.** A Genetic Algorithm for the Set Partitioning Problem, The Management School, Imperial College, London, Technical Report.

[17] **Beasley, J. E. and Chu, P. C. 1996.** A Genetic Algorithm for the Set Covering Problem, *European Journal of Operational Research*, vol. 94, no. 2, pp. 392-404.

[18] **Chu, P. C. and Beasley, J. E.1998.**Constraint Handling in Genetic Algorithms: The Set Partitioning Problem,*Journal of Heuristics*, vol. 4, no. 4, pp. 323-357.

[19] **Levine, D. 1996.** Application of a Hybrid Genetic Algorithm to Airline Crew Scheduling, *Computers & Operations Research*, vol. 23, no. 6, pp. 547 – 558.

[20] **Beasley, J.E.** OR-Library - Set Partitioning. [Last Accessed 7 August 2011]. http://people.brunel.ac.uk/~mastjjb/jeb/orlib/sppinfo.html

[21] **Kornilakis, H. and Stamatopoulos, P. 2002.** Crew Pairing Optimization with Genetic Algorithms, in *Lecture Notes in Artificial Intelligence (LNAI)*, Springer-Verlag Berlin, Heidelberg, vol. 2308, pp. 109 - 120.

[22] **Zeren, B. and Özkol, I. 2012.** An Improved Genetic Algorithm for Crew Pairing Optimization, *Journal of Intelligent Learning Systems and Applications*, vol. 4, pp. 70-80.

[23] **Mohan, C. K. and Ozdemir, H. T. 2001.** Flight Graph based Gentic Algorithm for Crew Scheduling in Airlines, *Information Sciences*, vol. 133, pp. 165 - 173.

[24] **ECAR121. 2010.** SUBPART Q: The Avoidance of Excessive Fatigue in Aircrew. [Last Accessed 2 January 2011] http://www.civilaviation.gov.eg/HTML/WEB-ARABIC/LOWS_REGULATIONS/ECAR_JULY,2010/Part121/Part121Q.htm#Part121Q_470