

Airline Crew Scheduling Using The Differential Evolution Method

Budi santosa
Industrial Engineering
Kampus ITS Sukolilo
Surabaya
phone +62315992364
budi_s@ie.its.ac.id

Andiek Sunarto
Industrial Engineering
Kampus ITS Sukolilo
Surabaya
phone +62315992364

Arief Rahman
Industrial Engineering
Kampus ITS Sukolilo
Surabaya
phone +62315992364

ABSTRACT

Airline crew rostering is the assignment problem of crew members to planned rotations/pairings for certain month. Airline companies have the monthly task of constructing personalized monthly schedules (roster) for crew members. This problem increased more and more complex and difficult while the aspirations/criterias grew to assess the quality of roster and the constraints increased excessively. This paper proposed the differential evolution (DE) method to solve the airline rostering problem. Different from the common DE, this paper presented random swap as mutation operator. The DE algorithm is proven able to find the near optimal solution accurately for optimization problem. Through numerical experiment with some datasets from Merpati Nusantara Airlines company, DE showed more competitive results than two other method, column generation and MOSI (the one used by the Airline). DE produced good results for small and medium datasets, but still showed reasonable results for large dataset.

Keywords

differential evolution, crew scheduling, pairing, rostering.

1. INTRODUCTION

In airline industry, what become priority in human resources department is development of crews rostering plan which able to produce the high utility of crews. It is estimated that optimization software which have been developed for airline could save more than US \$ 20 million per year. Saving 1% in crew utilization can save cost largely.

Though airline crews scheduling became attention in many operation research literature such as (Anbil et al., 1998; Barnhart et al., 1998; Gershkoff et al., 1996; Hoffman and Padberg, 1993; Ribeiro et al., 1994; Vance et al., 1997) but airline crews scheduling remains to become the main attention for many researchers that is caused by level of complexity and difficulty to solve it. Therefore method and approach which is used to solve it more and more grow to get the better result in optimality side and speed of computational time. Beside of reasons which mentioned previously. Generally, solving airline crew scheduling is done by decomposition approach (Chu, S.C.K, 2007; Lučić, Pand Teodorović, D, 1999), it divides problem to two problems, that are crew pairing and crew rostering. Crew pairing is done to get initial feasible solution, that is sequence of flight which begin and end on the same home base. Crew rostering assigns pairings which were arranged for the certain month to set of crews based on individual calendar. Decomposition approach is very effective to solve the difficult and complex problem but this method loss the global treatment since crew pairing and crew rostering done separately. Some other researchers developed the integrated approach to overcome obstacle, such as Souai and Thegem (2009), where crew pairing and rostering were done simultaneously to get level of better optimality.

Many optimization methods have been developed to solve crew scheduling to increase roster quality and to improve computational time such as simulated annealing (Lučić, and Teodorović, 1999), genetic algorithm (Souai, and Thegem, 2009), tree search algorithm (Beasley, and Cao, 1996), hybrid genetic algorithm (Levine, 1996), and GASA hybrid algorithm (Yinghui et al, 2007).

This research focused on developing differential evolution algorithm applied on intelligent airline crew rostering system. This paper is organized as follows. The second section reviews differential evolution (DE). Section 3 describes the formulation of crew scheduling problem. In section 4, we describe our methodology. Section 5 explains the experimental setting and the results. Section 6 discusses and concludes the results

2. DIFFERENTIAL EVOLUTION

Differential evolution is an evolutionary population-based algorithm proposed by Storn and Price (Kenneth, 2005). Since its initiation in 1995, DE has shown its performance as a very effective global optimizer. DE originated with Genetic Annealing (GA) Algorithm. Since GA was very slow and effective control parameters were hard to determine, the modification of the GA algorithm were made. DE uses a floating-point instead of bit-string encoding and arithmetic operations instead of logical ones. DE differs significantly from the evolutionary algorithms in the sense that distance and direction information from the current population is used to guide the search process. DE uses the differences between two randomly chosen vectors (individuals) as the base to form a third vector (individual), referred to as the target vector. Trial solutions are generated by adding weighted difference vectors to the target vector. This process is referred to as the mutation operator where the target vector is mutated. The next step is recombination or crossover which is applied to produce an offspring. This new individual is only accepted if it improves on the fitness of the parent individual.

DE has been applied in many field successfully. In 1995, DE has been used by Ken to solve 5-dimension Chebyshev model. By the time, Ken modified *genetic annealing* algorithm with *differential mutation operator* (Kennet et al, 2005). Tasgetiren (2007) used a discrete

differential evolution (DDE) algorithm to solve the single machine total earliness and tardiness penalties with a common due date. A new binary swap mutation operator called *Bswap* is presented. In addition, the DDE algorithm is hybridized with a local search algorithm to further improve the performance of the DDE algorithm. The performance of the proposed DDE algorithm is tested on 280 benchmark instances ranging from 10 to 1000 jobs from the OR Library. The computational experiments showed that the proposed DDE algorithm has generated better results than those in the literature in terms of both solution quality and computational time.

A genetic differential evolution (GDE) was derived from the differential evolution (DE) and incorporated with the genetic reproduction mechanisms, namely crossover and mutation used to solve traveling salesman problems (TSP). GDE was implemented to the well-known TSP with 52, 100 and 200 cities with variable parameters. Based on analysis and discussion on the results, typical values of the parameters were given, with which GDE provided effective and robust performance (Jian et al., 2008).

Omran and Salman (2009) improved Differential evolution by combining with chaotic search, opposition-based learning, and quantum mechanics, called CODEQ, to solve constrained optimization problems. The performance of the proposed approach when applied to five constrained benchmark problems is investigated and compared with other approaches proposed in the literature. The experiments conducted show that CODEQ provides excellent results with the added advantage of no parameter tuning.

3. PROBLEM FORMULATION

Crew rostering problem is a assignment problem of set of pairings which planned on certain month to set of crews. In this assignment model, some criterias will be achieved with considering some constraints. We modified the model which developed by Panta Lučić and Teodorović (Lučić and Teodorović, 1999) based on the real condition of rostering in PT. MNA (Merpati Nusantara Airlines). We modified the single crew and single aircraft become multiple crews and multiple aircrafts, also we added open time criteria to the objective function.

2.1.1 Index

k is index for kind of crews $k = 1, \dots, K$

i is index for numbers of crew members $(1, \dots, m_k)$.

j is index rotation/pairing which assigned to crew members $(1, \dots, n_k)$.

l is numbers of days in a month $(1, \dots, 31)$

2.1.2 Parameters

d_{jk} is length of rotation- j which assigned to crew k . it is expressed in hours.

$$p_{ilk} = \begin{cases} 1, & \text{if member } i \text{ from crew } k \text{ can be assigned to day } l \\ 0, & \text{otherwise} \end{cases}$$

$$q_{jlk} = \begin{cases} 1, & \text{if rotation / pairing } j \text{ assigned to crew } k \text{ start to day } l \\ 0, & \text{otherwise} \end{cases}$$

$d_{max,k}$ is maximum flight times crew k for one month

v_{jk} is numbers of take-off rotation j assigned to crew k

$v_{max,k}$ is maximum take-off in one month

$D_{min,jk}$ is minimum of numbers of crews k needed to complete rotation j

t_{jk} is numbers of duty period needed crew k to complete rotation j

$t_{max,k}$ is maximum of flying day before free day

$$\tilde{n}_{rsk} = \begin{cases} 1, & \text{if rotation } r \text{ overlap with rotation } s \text{ when assigned to crew } k \\ 0, & \text{otherwise} \end{cases}$$

3.1 Objectives Function

The objective function of this airline crew rostering is minimizing three terms of criterias.

Cost of roster

Cost of roster paid by airline company to crew is variable cost. By assumption that salary per hours is same to all crew, cost of roster can be represented by actual flying hours.

$$\min \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} d_{jk} x_{ijk} \quad k = 1, \dots, K \quad (1)$$

Deviation of flying days between crew members

Let $\bar{t}_k$ is average flying days per month crew members k , then

$$\bar{t}_k = \frac{\sum_{i=1}^{m_k} \sum_{j=1}^{n_k} t_{jk} x_{ijk}}{m_k} \quad k = 1, \dots, K \quad (2)$$

And deviation of total flying days per month can formulated as

$$\min \sum_{i=1}^{m_k} \left| \sum_{j=1}^{n_k} t_{jk} x_{ijk} - \bar{t}_k \right|^p \quad k = 1, \dots, K \quad (3)$$

Where p is positive integer. In this paper we use $p = 1$.

Open time

Open time is days when a crew member has not flying duty. If there are 31 days in a scheduling month, then *open time* for crew member k can be formulated as

$$\min \sum_{i=1}^{m_k} \left(31 - \sum_{j=1}^{n_k} t_{jk} x_{ijk} \right) \quad k = 1, \dots, K \quad (4)$$

There are some constraints which must be satisfied when constructing a roster. The following are the constraints used:

Flight time constraint

Maximum flying hours for pilot and co-pilot are 110 hours per month, and for cabin crew is 120 hours per month. So $d_{\max,k} = 110$ for $k = 1, \dots, 7$ and $d_{\max,k} = 120$ for $k = 8$.

$$\sum_{j=1}^{n_k} d_{jk} x_{ijk} \leq d_{\max,k} \quad i = 1, \dots, m_k, k = 1, \dots, K. \quad (5)$$

Duty Period Constraint

Maximum duty period which allowed to crew members k is 21 days.

$$\sum_{j=1}^{n_k} t_{jk} x_{ijk} \leq 21 \quad i = 1, \dots, m_k, k = 1, \dots, K. \quad (6)$$

Numbers of take-off

Numbers of maximum *take off* allowed to pilot is 90. then $v_{\max,k} = 90$. but cabin crew have no take off constraint.

$$\sum_{j=1}^{n_k} v_{jk} x_{ijk} \leq v_{\max,k} \quad i = 1, \dots, m_k, k = 1, \dots, K. \quad (7)$$

Numbers of crew requirement

Every rotation need minimum numbers of crew.

$$\sum_{i=1}^{m_k} x_{ijk} \leq D_{\min,jk} \quad j = 1, \dots, n_k, k = 1, \dots, K \quad (8)$$

Free day constraint

Every crew members must be given free day maximum after 7 flying day.

$$\sum_{j=1}^{n_k} t_{jk} x_{ijk} \sum_{l=p}^{p+7} q_{jkl} \leq 7 \quad i = 1, \dots, m_k, \quad p = 1, \dots, 23, \quad k = 1, \dots, 8 \quad (9)$$

Rotation without free day

When crew members complete this rotation not allowed free day.

$$\sum_{j=1}^{n_k} x_{ijk} = \sum_{j=1}^{n_k} x_{ijk} \sum_{l=1}^{31} q_{jkl} \prod_{s=l}^{l+t_{jk}-1} p_{isk} \quad i = 1, \dots, m_k, \quad k = 1, \dots, K \quad (10)$$

No overlap constraint

Two rotation in series may not be overlap each others. It means that precedence rotation must be finished when following rotation will start.

$$x_{ijk} \sum_{s=1}^{n_k} \rho_{jsk} x_{isk} (s - j) = 0 \quad i = 1, \dots, m_k, \quad j = 1, \dots, n_k, \quad k = 1, \dots, K \quad (11)$$

Airline rostering problem has many constraints which must be satisfied, while metaheuristic method can not solve constrained problem directly. Therefore the constrained problem must be transformed into unconstrained optimization problem. We use external penalty function method (Fox, 1971) to do this transformation. Basically, this method incurs big penalty while the solution violated any constraints. This method moved from infeasible solution forward to feasible solution.

$$\begin{aligned} \min & \beta_1 \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} d_{jk} x_{ijk} + \beta_2 \sum_{i=1}^{m_k} \left| \sum_{j=1}^{n_k} t_{jk} x_{ijk} - \bar{t}_k \right|^p + \beta_3 \sum_{i=1}^{m_k} \left(31 - \sum_{j=1}^{n_k} t_{jk} x_{ijk} \right) + \\ & (r_1 C_1 + r_2 C_2 + r_3 C_3 + r_4 C_4) + (r_5 C_5 + r_6 C_6 + r_7 C_7 + r_8 C_8) \end{aligned} \quad (12)$$

where :

$$\begin{aligned} C_1 &= \sum_{i=1}^{m_k} \left(\max \left(0, \sum_{j=1}^{n_k} d_{jk} x_{ijk} - d_{\max, k} \right) \right)^2 \quad k=1, \dots, K \\ C_2 &= \sum_{i=1}^{m_k} \left(\max \left(0, \sum_{j=1}^{n_k} v_{jk} x_{ijk} - v_{\max, k} \right) \right)^2 \quad k=1, \dots, K \\ C_3 &= \sum_{j=1}^{n_k} \left(\max \left(0, \sum_{i=1}^{m_k} x_{ijk} - D_{\min, jk} \right) \right)^2 \quad k=1, \dots, K \\ C_4 &= \sum_{i=1}^{m_k} \left(\max \left(0, \sum_{j=1}^{n_k} t_{jk} x_{ijk} \sum_{l=p}^{p+7} q_{jkl} - 7 \right) \right)^2 \quad k=1, \dots, K \\ C_5 &= \sum_{i=1}^{m_k} \left(\max \left(0, \sum_{j=1}^{n_k} x_{ijk} - \sum_{j=1}^{n_k} x_{ijk} \sum_{l=1}^{31} q_{jkl} \prod_{s=l}^{l+t_{jk}-1} p_{isk} \right) \right)^2 \quad k=1, \dots, K \\ C_6 &= \sum_{i=1}^{m_k} \left(-\min \left(0, \sum_{j=1}^{n_k} x_{ijk} - \sum_{j=1}^{n_k} x_{ijk} \sum_{l=1}^{31} q_{jkl} \prod_{s=l}^{l+t_{jk}-1} p_{isk} \right) \right)^2 \quad k=1, \dots, K \end{aligned}$$

$$C_7 = \sum_{i=1}^{m_k} \left(\max \left(0, x_{ijk} \sum_{s=1}^{n_k} \rho_{jsk} x_{isk} (s-j) \right) \right)^2 \quad k=1, \dots, K$$

$$C_8 = \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} \left(-\min \left(0, x_{ijk} \sum_{s=1}^{n_k} \rho_{jsk} x_{isk} (s-j) \right) \right)^2 \quad k=1, \dots, K$$

β_1, β_2 , and β_3 are weight coefficient of objective function, $r_1, \dots, r_8$ are penalties which are given since model violate any constraints where $r_1, \dots, r_8 \rightarrow \infty$ will assure that algorithm will satisfy constraint early before considering objective function. Equality (12) becomes fitness function of differential evolution method. If we assume cost of roster more important than deviation of flying day and more important than open time, then β_1, β_2 , and β_3 is selected carefully where $\beta_1 \gg \beta_2 \gg \beta_3$ and assure followed inequality is true :

$$\beta_1 \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} d_{jk} x_{ijk} \gg \beta_2 \sum_{i=1}^{m_k} \left| \sum_{j=1}^{n_k} t_{jk} x_{ijk} - \bar{t}_k \right|^p \quad (13)$$

$$\gg \beta_3 \sum_{i=1}^{m_k} \left(31 - \sum_{j=1}^{n_k} t_{jk} x_{ijk} \right)$$

Term (13) will assure hirarchical ordering of solving solving iteratively by differential evolution method.

4. SOLVING THE MODEL USING DE

4.1 Initialization

Initial solution x_{ijk} is defined by generating biner random number [0 and 1] by $n_k \times m_k$ dimension to the amount of n_pop . Let $X_{np,0}$ is initial solution population np , then

$$X_{np,0} = rand_{np}[0;1] \quad np = 1, \dots, n_pop \quad (14)$$

4.2 Mutation

Different from those which usually used as mutation operator in DE, this paper introduce a random swap as mutation operator. Let $r0$ be random number between 0 and 1 with $m_k \times n_k$ dimension for every population np , $v_{np,r0,g}$ be element of solution V at column $r0$ and generation g , if $W_{np,g-1}$ is the best population for generation $g-1$ and $w_{np,r0,g-1}$ element at column $r0$ of $W_{np,g-1}$, then mutant of generation g (15)

$$v_{np,r0,g} = \begin{cases} (W_{np,r0,g-1} + 1) \bmod(2), & \text{if } r0 < c_m \\ W_{np,r0,g-1}, & \text{otherwise} \end{cases} \quad (15)$$

where c_m is mutation probability between 0 and 1 which represent mutation power done to the best population of previous generation. The selection of c_m must be done carefully because too small c_m can cause old solution is difficult to exit from local optima while too big c_m cause noise solution then fast convergence toward global optima can not be achieved. Selecting c_m accurately become success key of this algorithm.

4.3 Crossover

Crossover change over parent solution $X_{np,g}$ by mutant solution $V_{np,g}$ to construct a new solution $U_{np,g}$. Crossover is done by defining threshold probability ($0 < c_r < 1$) for mutant to change old solution. Then we generate n_pop random numbers (0,1). If random numbers $< c_r$, then the mutant replaces the old solution and otherwise.

$$U_{np,g} = \begin{cases} V_{np,g}, & \text{if } rand_{np}(0,1) < c_r \\ X_{np,g}, & \text{otherwise} \end{cases} \quad (16)$$

4.4 Selection

This process is done by comparing the fitness function between parent solution and new solution which produced from crossover process. Parent population which has better performance than new population will be kept on next iteration, otherwise new solution will replace the old solution. The fitness function refers to equation (12). Then, solution of the following generation $X_{np,g+1}$ can obtained from this formula :

$$X_{np,g+1} = \begin{cases} U_{np,g}, & \text{if } f(U_{np,g}) < f(X_{np,g}) \\ X_{np,g}, & \text{otherwise} \end{cases} \quad (17)$$

$f(U_{np,g})$ is fitness function of $U_{np,g}$ while $f(X_{np,g})$ is fitness function of $X_{np,g}$.

Mathematical model which is constructed consists of some aspirations/criterias as objective functions and some constraints. Objective function includes minimum flying times, deviation of flying days, and open time. Some constraints which are considered when constructing a roster include overlap, crews requirement of pairing, free day before seven days of flying days, maximum flying times, and maximum numbers of take off.

Datasets used in this research come from private Indonesian company, MNA (Merpati Nusantara Airlines) . We used Matlab to construct computer program. This step must be done since very larger dataset must be solved where need thousands of iteration and very long computational time.

5. EXPERIMENTS AND ANALISYS

We show result of numerical experiments and compare results which produced by other method, called column generation, MOSI, and exact decomposition. The algorithm will stop when maximum numbers of iteration or generations achieved (g_{max}) . Eksperiments are done using Datasets consist of pairings, numbers of crews, kind of flected, and rules (Rusdiansyah, 2007).

Tabel 1. characteristic of datasets

No	Name of	type of	numbers	numbers
	aircraft	crews	of crews	of pairings
1	F-100	Pilot	5	4
2	CN-235	Pilot	4	8
3	DHC-6	Pilot	3	10
4	Cassa-212	Pilot	6	13
5	Boeing 737-200	Pilot	17	82
6	Boeing 737-200	stewardess	55	114

Through experiments, we got the best mutation probability (c_m) is 0.1 and the best crossover probability (c_r) is 0.5. Penalties of overlap and rotation without free day constraint, numbers of pilot requirements and day off are 10^{15} , 10^{13} and 10^{11} . While, the penalty for flying day constraint, flying times constraint, and numbers of take off is 10^6 . The Weights for flying time, deviation of flying days and open time respectively are 10^4 , 100, and 1.

Dataset is divided into two sizes, small datasets and large datasets. Small datasets consist of assignment of F-100, CN-235, DHC-6, and Cassa 212 pilot. While, large dataset consist of assignment of Boeing 737-200 pilot and stewardess.

Eksperimental results show that DE, column generation, and MOSI produced the same performance to meet the roster constraints. Exception to Cassa 212 aircraft, DE can meet all of constraints while column generation and MOSI contravene pilot requirements constraints. Column generation and MOSI just assign 1 pilot to pairing 8981 while pilot requirements of pairing 8981 is 2 persons. From roster quality side, DE is only inferior for flying times criterias to assignment of Cassa 212 aircraft. While all of method showed the same result to assignment other aircrafts.

Tabel 2. Comparing flying times

Name of Aircraft	Fying time		
	DE	Column Generation	MOSI
F-100	66.0	66.0	66.0
CN-235	92.0	92.0	92.0
DHC-6	156.0	156.0	156.0
Cassa 212	251.0	242.0	242.0

From table 3, DE produced superior deviation of flying days to assignment of three aircraft than other methods. And MOSI is superior to Cassa 212 aircraft.

Tabel 3. Comparing deviation of flying days

Name of Aircraft	Flying time		
	DE	Column Generation	MOSI
F-100	66.0	66.0	66.0
CN-235	92.0	92.0	92.0
DHC-6	156.0	156.0	156.0
Cassa 212	251.0	242.0	242.0

At open time criterias, DE produced superior result than other methods to Cassa 212 aircraft and the same result to other aircrafts.

Table 4. Comparing open time

Name of Aircraft	Open time		
	DE	Column Generation*	MOSI*
F-100	137.0	137.0	137.0
CN-235	88.0	88.0	88.0
DHC-6	50.0	50.0	50.0
Cassa 212	119.0	123.0	123.0

Dataset for Boeing 737-200 aircraft has larger size (17 pilots and 82 pairings, 55 stewardess and 114 pairings) than the other airplanes. These need high number of iterations and computational time to obtain near optimal solution if random initial solution is used. This paper proposed the sequence of partial optimization and total optimization techniques. In partial optimization, dataset are splitted into some sets and the corresponding optimization problems are solved by DE method. At the end, all of solutions are combined all together to form initial solution for total optimization. The better initial solution will decrease computational time and increase processes efficiently.

At assignment of Boeing 737-200 pilot, we compare 6 methode, called differential evolution (DE), column generation, MOSI, and exact dekomposition for 21 flying days (DE21). All of methods contravene flying days constraint. we recorded that DE assigned smoother flying days than other methods if it is seen from deviation standart which produced. We also recorded that DE produced the least number of pilot which contravene flying days constraint, 3 pilots. While column generation produced 6 pilots which contravenes the constraint, 7 pilots by MOSI, and 7 pilots by DE21.

Tabel 5. Comparing num. and Fly. days

Pil	Differential Evolution		Column Generation*		MOSI*		Metode DE21 **	
	fly day	diff.	fly day	diff.	fly day	diff.	fly day	diff.
1	21	0	20	0	21	0	20	0
2	20	0	18	0	21	0	22	1
3	20	0	20	0	23	2	26	5
4	21	0	20	0	22	1	24	3
5	21	0	17	0	18	0	17	0
6	21	0	20	0	22	1	24	3
7	22	1	21	0	16	0	18	0
8	21	0	26	5	21	0	22	1
9	20	0	16	0	17	0	20	0
10	18	0	18	0	16	0	13	0

11	22	1	23	2	22	1	19	0
12	21	0	21	0	21	0	22	1
13	20	0	24	3	24	3	22	1
14	21	0	23	2	24	3	20	0
15	20	0	20	0	19	0	21	0
16	20	0	22	1	21	0	21	0
17	22	1	22	1	23	2	18	0
Total	351	3	351	14	351	13	349	15
Avrg	21		21		21		21	
Std.D	1		3		3		3	
Num of pil contravene fly. days		3		6		7		7

Difference from pilots assignment, at stewardess assignment of Boeing 737-200 aircraft we compare 3 methods, called DE, column generation, and MOSI. Table 6 showed that DE only collide with crews requirement, while column generation and MOSI collide with both of crews requirement and flying days constraint. We can see in table 6 that column generation assigns 6 stewardesses, exceed maximum flying days, 21 days. And, MOSI assigns 7 stewardesses, exceeds 21 days.

Tabel 6. Violation of flying days constraint for assignment of Boeing 737-200 stewardess

Constraints	Method		
	DE	Column Gen.	MOSI
Flying days	-	√	√
Flying times	-	-	-
Take off	-	-	-
Overlap	-	-	-
Free day	-	-	-
Requirement of pilots	√	√	√

At assignment of Boeing 737-200 pilot, DE is superior for minimum deviation of flying days than other methods.

Tabel 7. Roster quality of assignment of Boeing 737-200 pilots

Criterias	Methods			
	DE	Column	MOSI*	DE21**
		Gen.*		
Flying times	1,114.0	1,114.0	1,114.0	1,114.0
Dev. of fly. days	13.1	33.6	34.5	38.5
Open time	176.0	176.0	176.0	178.0

Tabel 8. Roster quality of assignment of Boeing 737-200 stewardess

Criterias	Methods		
	DE	Column Generation	MOSI*
Flying times	3,488.0	3,285.0	3,285.0
Dev. of fly. days	92.0	110.0	106.2

Open time	693.0	658.0	658.0
-----------	-------	--------------	--------------

At assignment of Boeing 737-200 stewardess, DE is superior for minimum deviation of flying days than other methods. But inferior to other criterias.

6. CONCLUSIONS

The rostering problem in MNA has characteristic indifferent generally from other airline companies about roster constraint and roster quality. Selecting mutation and crossover probability accurately become success key to use DE. The best mutation and crossover probability are 0.1 and 0.5 respectively. Different from using DE generally, this paper introduced random swap as mutation operator. For small datasets, DE was proven more competitive then other methods, called column generation and MOSI. It is seen from meeting constraint and roster quality. At assignment of Boeing 737 pilot, DE produced smoother flying days and the least pilots which violate flying days constraint compared than column generation, MOSI, and DE21. And at assignment of Boeing 737-200 stewardess, DE violated the least constraint compared than column generation and MOSI. While DE produced superior deviation of flying days and inferior to both of other criterias

7. REFERENCES

- [1] Anbil, R., Forrest, J. J., and Pulleyblank, W. R. Column generation and the airline crew pairing problem. Documentation Mathematica Extra Volume ICM 1998, III, 677-686
- [2] Barnhart, C., Johnson, E.L., Nemhauser, G. L., Savelsbergh, M. W. P., and Vance, P. H. Branch and price : Column generation for solving huge integer programs. Operation research, vol. 46, no. 3, 1998
- [3] Beasley, J.E. and Cao, B. A tree search algorithm for the crew scheduling problem. European Journal of Operational Research 94, 517-526. 1996.
- [4] Chu, S.C.K. Generating, scheduling, and rostering of shift crew-duties: Applications at the Hongkong International Airport. European Journal of Operational Research 177, 1764-1778. 2007.
- [5] Fox, R.L. Optimization Methods for Engineering Design. Addison-Wesley, Reading, Mass. 1971.
- [6] Gershkoff, I., Graves, G. W., Mc Bridge, R. D., Anderson, D., and Mahidhara, D. Flight crew scheduling. Management science, vol. 39, no. 6, 1993
- [7] Hoffman, K. L. and Padberg, M. Solving airline crew scheduling problems by branch and cut. Management science, vol. 39, no. 6, 1993
- [8] Kennet V.P, Rainer M.S., and Jouni A.L. Differential Evolution : a practical approach to global optimization. Springer-Verlag Berlin Heidelberg. 2005.
- [9] Levine, D. Application of a hybrid genetic algorithm to airline crew scheduling. Computer Operations Research, volume 23, no. 6, pp. 547-558. 1996.
- [10] Lučić, P. and Teodorović, D. Simulated annealing for the multi-objective aircrew rostering problem. Transportation Research Part A 33, 19 – 45. 1999.
- [11] Ribeiro, C. C. A column generation approach to the multiple depot vehicle scheduling problem. Operation research, vol. 42, no. 1, 1994
- [12] Rusdiansyah, A., Mirenani, Y.D., and Labiba, Z. Pemodelan dan penyelesaian permasalahan penjadwalan pilot dengan metode eksak dekomposisi. Jurnal Teknik Industri vol. 9, No. 2, Desember 2007: 112-124. 2007.
- [13] Souai, N. and Thegem, J. Genetic algorithm based approach for the integrated airline crew-pairing and rostering problem. European Journal of Operational Research 199, 674-683. 2009.
- [14] Vance, P. H, Barnhart, C., Johnson, E. L., and Nemhauser, G. L. Airline crew scheduling : A new formulation and decomposition algorithm. Operation research, vol. 45. no. 2, 1997.
- [15] Yinghui, Z., Yunbao, R., and Mingtian, Z. GASA hybrid algorithm applied in airline crew rostering system. Tsinghua Science and Thechnology ISSN 1007-0214 46/49 pp. 225-259, volume 12, number S1. 2007.