

A hybrid particle swarm optimization and an improved heuristic algorithm for an airline crew rostering problem

Varis Limlawan, Boontariga Kasemsontitum, Chawalit Jeenanunta*

Sirindhorn International Institute of Technology, Thammasat University, Pathum Thani 12121 Thailand

*Corresponding author, e-mail: chawalit@siit.tu.ac.th

Received 4 Jul 2013

Accepted 28 Sep 2014

ABSTRACT: The airline crew scheduling problem is a combinatorial optimization problem and belongs to the class of NP-hard problems. An effective method for solving the airline crew scheduling problem can reduce the crew costs and improve crew satisfaction. Because of its complexity, the problem is divided into two subproblems: the crew pairing problem and the crew rostering problem. In this paper, the crew rostering problem is focused on and the objective is to generate a fairness timetable in which the workloads are distributed among each crew equally. We propose a hybrid particle swarm optimization (PSO) and an improvement heuristic (IH) to solve this problem. The IH is designed to improve the standard deviation of the workloads by picking a workload from the high workload crew and assigning it to the low workload crew. The IH improves the solution of the particle after the particle changes position each generation. The proposed algorithm is tested on actual pairing data from Thai Airways and is compared with PSO without IH and the multi-commodity network flow approach. With the combination of PSO and IH, the algorithm can improve the quality of the solution by more than 20% in most cases, and PSO with IH also outperforms the network approach in 6 out of 9 cases and especially in the large size cases for which the network approach cannot find a feasible solution.

KEYWORDS: airline crew scheduling

INTRODUCTION

The airline crew scheduling problem is one of the most difficult scheduling problems because of the large number of flight legs and the complex set of constraints. The problem is very important in an airline operation because the crew costs are the second largest cost. Thus effective solving methods can reduce the crew costs and improve crew satisfaction. Generally, most airlines divide the construction of crew schedules into two phases: an airline crew pairing problem (ACPP) and an airline crew rostering problem (ACRP). In the first phase, the schedule of flight pairings is generated. The flight pairing is a sequence of flight segments on consecutive days, worked by a crew that begins and ends at the same crew base. In general, the objective of the ACPP is to find a minimum pairing schedule which covers all of the flight segments in a specific period. In the second phase, the ACRP constructs a personalized schedule (roster) in the specific period for each crew member. The rosters are created by assigning the crew members to the flight pairings according to the various constraints such as rest periods. In addition, the pre-assigned activities and each crew member's qualifications must be taken into account in each

roster's construction.

The ACRP is a combinatorial optimization problem and belongs to the class of NP-hard problems. To solve a large-scale ACRP in a reasonable time using an exact algorithm is impossible. The existing methods for solving ACRPs are decomposing the problem into the series of assignment problems such as day-by-day method¹⁻³, pilot-by-pilot method⁴, and a combination of the two methods⁵. The disadvantage of the heuristic methods is the quality of the solution and the flexibility in response to the different ACRPs. An ACRP is formulated as a generalized set-partitioning model and solved by branch and bound⁶, column generation technique⁷, branch and cut⁸, or multi-commodity flow approaches^{9,10}. Although the exact approaches can give the optimal solution, the approaches cannot tackle the large-scale ACRP. The metaheuristic approach which is focused on here has been applied to solve ACRPs. Examples of this method are simulated annealing¹¹, genetic algorithm¹¹⁻¹³, Tabu search¹¹ and particle swarm optimization¹⁴. Moreover, there are hybrids of the two metaheuristic techniques for enhancing the performance of the algorithm such as the hybrid between genetic algorithm and simulated annealing¹⁵ and the hybrid scatter search heuristic¹⁶. The method

can generate a good feasible solution in a short time, so the method can solve large-scale problems which the airline company is interested in.

Thai Airways is trying to find an effective method for solving its ACRP. The current method of Thai Airways uses commercial software which is generic. Thus the software takes a long time to run and does not guarantee the feasibility of the result.

PROBLEM DESCRIPTION

Normally, the inputs of an ACRP are objectives for generating the roster, activities or duties (e.g., pairing, ground activities, and training activities), crew profile (e.g., qualifications, pre-assignments, and vacations), and regulations¹⁷. The objectives of an ACRP is normally divided into costs and crew preferences¹⁸. The costs have a direct effect on the airline costs, and the crew preferences have an effect on the quality of services. Constraints in an ACRP depend on the rules. Normally, the rules are about the safety and health of crew members (e.g., rest period restrictions, flight duties period restrictions^{9,13,16}, working hour restrictions and the holiday rule of Nok Air, a Thai domestic low cost airline¹⁹).

The ACRP at Thai Airways

The problem considered in this paper is from Thai Airways international flights which service five routes: North America, Europe, African, Australia and New Zealand, and regional (Asia). The cabin crews on the flights are divided into six categories which are in-flight manager (IM), air purser, first class crew (F), E-business class crew (E), R-business class crew (R), and economic class crew (Y). The ACRP of each crew category is handled separately because the all-categories problem is difficult to solve.

Thai Airways considers the fairness of the roster because it is very important for all cabin crew to receive equal benefits such as workload, station, and per diem. Here we focus only on the workload. The objective of the ACRP is to balance the workload among crew members in the schedule. Some regulations concern the block time or flight time which is the time between the plane moving from the runway at the origin and the plane stopping at the destination airport. The flight time of a crew member must not exceed 34 h in a 7 day period, 110 h in a 28 day period, or 1000 h in a year. The flight duty period is the period that starts one hour before block time to thirty minutes after the block time. A crew who is scheduled to a flight duty must immediately be assigned to a rest period. The rest period is a period during which a crew must not

Table 1 The flight duty period constraints.

flight duty period (h)	rest period (h)
0–8	≥ 8
8–10	≥ 10
10–12	≥ 12
12–14	≥ 14
14–16	≥ 16

be assigned to any duty. The length of a rest period is associated with the length of flight duty (Table 1).

Problem formulation

In order to distribute the workload as equally as possible among the crew members, the objective of the formulated model is to minimize the standard deviation of the workload. The mathematical model can be formulated as a nonlinear optimization as follows:

$$\min \sqrt{\frac{\sum_{c=1}^n [\sum_{p=1}^m x_{cp} w_p - (\frac{\sum_{p=1}^m W_p}{n})]^2}{n}} \quad (1)$$

$$\sum_{c=1}^n x_{cp} = 1, \quad \forall p \in P, \quad (2)$$

$$\sum_{d=g}^{g+6} \sum_{p=1}^n x_{cp} B_p^d \leq 2040, \quad \forall c \in C, \forall g \in D, \quad (3)$$

$$\sum_{d=g}^{g+27} \sum_{p=1}^n x_{cp} B_p^d \leq 6600, \quad \forall c \in C, \forall g \in D, \quad (4)$$

$$\sum_{d=g}^{g+364} \sum_{p=1}^n x_{cp} B_p^d \leq 60000, \quad \forall c \in C, \forall g \in D, \quad (5)$$

$$x_{cp} x_{cl} = \begin{cases} 1, & \text{if } x_{cp} = 1 \text{ and } S_l - E_p \geq 0, \\ 0, & \text{otherwise.} \end{cases} \quad \forall c \in C, \forall p \in P, \quad \forall l \in \{p+1, p+2, \dots, m\}, \quad (6)$$

$$x_{cp} \in \{0, 1\} \quad \forall c \in C, \forall p \in P. \quad (7)$$

P, D, C are the sets of pairs, days in the schedule, and crew members in the schedule, respectively, and c, p, d are their respective indices. x_{cp} is a binary variable where $x_{cp} = 1$ if a crew member c is assigned to pair p , and $x_{cp} = 0$, otherwise. x_{cl} is a binary variable for crew member c and the pairs after pair p , where $x_{cl} = 1$ if a crew member c is assigned to pair l , and $x_{cl} = 0$ otherwise. Let w_p, B_p^d, S_l, E_p be the workload

of pair p , the block time of pair p which is on day d , the starting time of pair l , and the ending time of pair p including rest period, respectively.

In the model, the objective function is presented by (1) which is set to minimize the standard deviation of the workload among crew members. The requirement that each pair must be assigned to a crew is expressed by (2), and (3)–(5) are the block time constraints. Constraints in (6) prevents an assignment of a pair to a crew member who is working or is on a rest period. The product of x_{cp} and x_{cl} checks the validity of assigning pair l after pair p was assigned. Pair l can be assigned if the starting time of pair l is not less than ending time of pair p . Eq. (7) is the binary constraint.

PARTICLE SWARM OPTIMIZATION (PSO)

PSO was introduced by Kennedy and Eberhart^{20,21}. The first version was modified by Shi and Eberhart²², and the modified version has become a popular version of PSO. Reviews of the PSO algorithm and its applications can be found in Ref. 23 and Ref. 24.

A member of the population in the algorithm is represented by a particle which contains a position (solution) and a velocity. Each particle moves in a multidimensional space by adjusting its position and its velocity based on its own experience and the experience of the population. Each particle memorizes its best position (p_{best}) and the population also memorizes the best position in the group (g_{best}). The velocity and the position are updated as follows.

$$v_{id}(t+1) = wv_{id}(t) + c_1r(p_{id} - x_{id}(t)) + c_2r(p_{gd} - x_{id}(t)) \quad (8)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (9)$$

where $v_{id}(t)$, $x_{id}(t)$, and p_{id} are, respectively, the d th components of the velocity, position, and best position of particle i at iteration t . p_{gd} is the d th component of the global best position. w is inertia weight which controls the impact of the previous velocity. c_1 , the acceleration of personal best position, is the parameter to control the effect of p_{best} . c_2 , the acceleration of best global position, is the parameter to control the effect of g_{best} . r is a uniform-distributed random number in the range $[0, 1]$. The effect of inertial weight and the accelerations should be set appropriately for balancing the local search and the global search²³.

In order to apply the PSO in discrete binary space, Kennedy and Eberhart introduced the discrete binary PSO²⁵. Each position contains the binary variable number. The velocity equation is the same as the

standard PSO, but the method to change the position is changed. The value of the new velocity is transformed to a value in the range $[0, 1]$ by using the sigmoid function:

$$s(v_{id}(t+1)) = \frac{1}{1 + e^{-v_{id}(t+1)}}. \quad (10)$$

The particle changes the position by

$$x_{id} = \begin{cases} 1, & \text{if } r \leq s(v_{id}(t+1)), \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

PSO FOR AIRLINE CREW ROSTERING

Solution representation

In our previous work¹⁴, we implemented the discrete binary PSO to solve an ACRP. In that implementation, a particle is represented as a binary timetable of crew members in which the rows and the columns represent crews and pairs, respectively. The position of particles is a set of slots in the timetable which represents the assignment of the crew and the pair corresponding to those slots. Let the position of the particle be

$$x_{cp}^k = \begin{cases} 1, & \text{if the } c\text{th crewmember is assigned to,} \\ & \text{the } p\text{th pair in the } k\text{th particle,} \\ 0, & \text{otherwise.} \end{cases}$$

Repairing algorithm

The concept of the algorithm is similar to the discrete binary PSO but the algorithm needs the repairing algorithm for fixing the infeasible particle. The first step is to identify and repair the covering problems. For example, more than one crew is assigned to operate a pair and no crew is assigned to operate a pair. Once the covering problems are solved, all pairs are assigned to all crews. The second step is to repair the illegal pairs which violate the constraints such as block time and rest period. The illegal pairs are deleted from the schedule and added to the deleted pair list. The third step is to reassign the pairs in the deleted pair list to the schedule. The reassignment considers the feasibility and the value of the objective function. The deleted pairs are assigned to a crew who have the lowest workload in order to balance the workload. Source 1: More than one crew is assigned to a pair. The workloads of a crew assigned to the pair are compared with each other, and the crew with the minimum workload is selected to operate the pair. Source 2: No crew is assigned to a pair. The crew with the minimum workload is assigned to the pair. Source 3: Block time constraint violation. If CB_{cp} exceeds the block time limitation, pair p is deleted from the

schedule of crew c . The deleted pair is added to the deleted pair list. Source 4: The flight duty and the rest period constraint violation. The flight duty period and the rest period constraints are checked following constraint (6). The checking starts from the first pair to the last pair. If there is an illegal pair which is assigned in the rest period of the previous pair, the pair is removed from the schedule and added in the removed-pair set. For example, pair m starts on day 1 and finishes on day 2, and the pair n starts on day 2 and finishes on day 3. If both pairs were assigned to a crew, one of those pairs might be deleted from that crew and assigned to the deleted pair list.

Procedure

The procedure of PSO for the ACRP is similar to the discrete binary PSO, but infeasible particles must be repaired by the repairing algorithm before they are evaluated for the fitness value and moved to the next generation.

- Step 1: Generate initial solution for each particle by adding the c crew member to the p pair randomly until all pairs are assigned. The velocity of each particle is also random.
- Step 2: If the initial solution is feasible, go to Step 3, otherwise the solution is repaired by the repairing algorithm.
- Step 3: For each particle, evaluate the desired optimization fitness function.
- Step 4: Compare particle fitness with the best particle position (p_{best}). If the current value is better than p_{best} , set p_{best} equal to the current value and p_{best} equal to the current location. In some cases, the particle from the repairing algorithm is still infeasible. The infeasible particle will not be saved as p_{best} although the fitness value of the infeasible particle is better than the fitness value of p_{best} .
- Step 5: Compare fitness evaluation with overall best (g_{best}). If the current value is better than g_{best} then set g_{best} and its location equal to the current value and location. In the case of an infeasible particle, the particle will be handled the same way as in Step 4.
- Step 6: Change velocity of each particle using

$$v_{cp}^k(t+1) = wv_{cp}^k(t) + c_1r()(p_{cp}^k - x_{cp}^k(t)) + c_2r()(g_{cp} - x_{cp}^k(t)) \quad (12)$$

- Step 7: Change the position of each particle by using (10) and (11)
- Step 8: Repeat Step 2 until stopping criterion is met.

PROPOSED HYBRID PSO WITH IMPROVEMENT HEURISTIC (PSO-IH)

The PSO-IH is developed based on PSO for an ACRP¹⁴. The structure of the algorithms is the same, but the particle can improve itself. In every iteration, each particle is improved by the improvement heuristic (IH) after adjusting its position.

The improvement heuristic (IH)

The improvement heuristic (IH) is designed for improving the solution of each particle. In this case, the objective function of the ACRP is to minimize the standard deviation of the workload. The crew members in the schedule are ranked based on an individual's workload. The pairs from the crews in the first half of the ranking (and have the highest workload) are picked and reassigned to the crew who have the lowest workload.

- Step 1: Rank the crews in the schedule according to the individual crew member's workload.
- Step 2: Choose a crew in the first half of the ranking starting from the highest ranking crew (the highest workload).
- Step 3: Select a pair from the considered crew.
- Step 4: Check that the selected pair can be assigned to the last crew in the ranking. The addition of the pair should not violate any constraints such as rest period and block time. If the selected pair cannot be assigned to the last crew, select the new pair from the considered crew (go to Step 3). If all pairs in the considered crew were selected and cannot be assigned to the last crew in the ranking, select a pair from the next highest crew in the ranking (go to Step 2). If all crews in the first half were considered, go to Step 7. If the selected pair can be assigned to the last crew in the ranking, go to Step 5.
- Step 5: Reassign the selected pair to the last crew in the ranking.
- Step 6: Repeat Step 1.
- Step 7: End of the process.

Main procedure

- Step 1: Generate an initial solution for each particle.
- Step 2: Repair infeasible particles by the repairing algorithm.
- Step 3: For each particle, perform the IH.
- Step 4: For each particle, evaluate the fitness function of the problem.
- Step 5: Compare particles fitness values for setting p_{best} and g_{best} .

Step 6: Change the velocity of each particle using (12).

Step 7: Change the position of each particle by using (10) and (11).

Step 8: Repeat Step 2 until the algorithm reaches the setting iteration.

COMPUTATIONAL RESULT

In order to evaluate the performance of PSO-IH, the algorithm is compared with PSO without IH and the multi-commodity network flow approach which is based on Ref. 10. There are two experiments in the evaluation. The first experiment aims to determine the best population size and the number of iterations for PSO-IH in each problem. The second one is the performance evaluation of PSO-IH by comparing with PSO and the network approach. All models in the evaluation are run on an Intel Core i5 M650 3.5 GHz, 4.00 GB machine running under Windows. PSO-IH and PSO were coded using Microsoft Visual Studio 2010-C sharp, and the multi-commodity network flow model is coded by IBM ILOG CPLEX 12.0.

Test problem

The proposed algorithm is evaluated on nine generated problems, which represent various types of the problem such as problem size and duration of pairs. All problems focus on assigning the in-flight managers (IM) to the international pairs during 14 days (Table 2).

Population size and number of iterations setting

To find the optimal solution of PSO-IH, population size and the number of iterations are determined by conducting a two-way ANOVA with 95% confidence interval. In this experiment, we have two factors, three levels, ten replications, and nine problems, thus there are 540 cases. The other parameters of PSO are the same for each case: acceleration constant (c_1

Table 2 All test problems.

Instance	No. of pairs	No. of in-flight managers (IM)	Duration of pairs
TA84S	84	30	Short haul
TA84M	84	30	Mix
TA84L	84	45	Long haul
TA150S	150	60	Short haul
TA140M	140	65	Mix
TA146L	146	65	Long haul
TA330S	330	150	Short haul
TA334M	334	150	Mix
TA238L	238	120	Long haul

Table 3 Population size and number of iterations for each problem.

problem	population size	number of iterations
TA84S	100	1000
TA84M	60	500
TA84L	100	500
TA150S	100	500
TA140M	100	100
TA146L	60	1000
TA330S	100	500
TA334M	60	1000
TA238L	20	500

and c_2) = 2, maximum velocity (V_{\max}) = 4.0, and inertial weight (w) is linearly decreased from 0.9 at the first iteration to 0.4 at last iteration. The results are summarized in Table 3.

Comparison of the solutions

In the comparison experiment, the parameters of PSO are set as follows: acceleration constant (c_1 and c_2) = 2, maximum velocity (V_{\max}) = ± 4.0 , and inertial weight (w) linearly decreased from 0.9 at the first iteration to 0.4 at the last iteration. The population size and number of iterations are obtained from the previous section. PSO-IH and PSO are run thirty times for each problem. The averaged results are presented in Table 4. The solution from PSO-IH is better than PSO without IH in all problems and outperforms the multi-commodity network flow approach (last two columns in Table 4) in 6 out of 9 problems. The multi-commodity network flow model cannot find a feasible solution in three large-size instances. For the computational time, PSO-IH is obviously worse than PSO without IH in all problem and is also worse than the multi-commodity network flow model in TA84S and TA84L, but the computational time of both methods are close to each other in the medium size instances.

Based on the results, the addition of IH can improve the solution quality of PSO by more than 20% in 7 out of 9 problems while the little improvements in the TA84L and TA146L are caused by the nature of the problems, which consist of long haul pairs. It is difficult to find the feasible slot in the minimum workload crew for reassigning a long haul pair picked from the maximum crew. PSO-IH performs better than the multi-commodity flow model because the objective function of PSO-IH is more appropriate than the network model in which the objective function is minimizing the upper bound of the workload. The lower bound of the solution obtained from the network

Table 4 Comparison of the solutions.

Instance	Result from PSO	CPU time from PSO (s)	Result from PSO-IH	CPU time from PSO-IH (s)	Result from network model	CPU time from network model(s)
TA84S	13.53	6859	7.96	15 300	6.90	749
TA84M	15.29	2706	8.59	4335	10.38	7927
TA84L	19.28	4876	18.95	8280	20.10	164
TA150S	15.73	10 534	7.35	17 181	6.81	12 363
TA140M	16.43	2655	12.59	5673	13.04	13 464
TA146L	24.45	10 672	22.83	17 673	22.56	24 097
TA330S	19.95	125 783	14.23	249 552	N/A	N/A
TA334M	19.98	113 658	13.37	273 543	N/A	N/A
TA238L	12.54	12 675	9.51	23 798	N/A	N/A

N/A = The model cannot find a feasible solution.

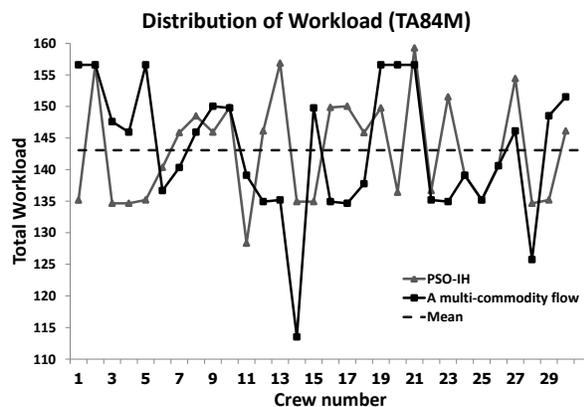


Fig. 1 Distribution of workload for both techniques in TA84M.

model may be far from the average as shown in the examples in Fig. 1. Furthermore, the network model cannot obtain a feasible solution for large problems which normally are found in the practice.

CONCLUSION

In this paper, the hybrid PSO and the improvement heuristics for solving ACRP, called PSO-IH are proposed. The performance of PSO-IH was evaluated by comparing the PSO without IH and the upper bound generated by the multi-commodity network flow approach. The evaluation was on the generated ACRP problems from Thai Airways. The results show that PSO-IH outperforms PSO without IH and the network approach; moreover PSO-IH can solve large ACRP problems which the network approach cannot solve. However, the addition of IH requires

a larger CPU time than PSO without IH while PSO-IH requires shorter time than the multi-commodity network flow approach in large problems. Also the network approach cannot find a feasible solution for the large problems. For further study, the IH should be developed for solving ACRP with different objectives because the proposed IH explained in this paper can only solve the objective of Thai Airways.

Acknowledgements: The research was supported by Low Carbon Technology Research Grants and the National Research University Project of Thailand Office of Higher Education Commission. The authors thank Thai Airways for disclosing operating and flight information.

REFERENCES

1. Nicoletti B (1975) Automatic crew rostering. *Transport Sci* **9**, 33–42.
2. Buhr J (1978) Four methods for monthly crew assignment - a comparison of efficiency. In: *Proceedings of the AGIFORS 18th Annual Symposium*, pp 403–30.
3. Glanert W (1984) A timetable approach to the assignment of pilots to rotations. In: *Proceedings of the AGIFORS 24th Annual Symposium*, pp 369–90.
4. Moore R, Evans J, Noo H (1978) Computerized tailored blocking. In: *Proceedings of the AGIFORS 18th Annual Symposium*, pp 343–61.
5. Antosik JL (1978) Automatic monthly crew assignment: a new approach. In: *Proceedings of the AGIFORS 18th Annual Symposium*, pp 369–402.
6. Ryan DM (1992) The solution of massive generalized set partitioning problems in air crew rostering. *J Oper Res Soc* **43**, 459–67.
7. Gamache M, Soumis F (1998) A method for optimally solving the rostering problem. In: Yu G (ed) *Operation Research in the Airline Industry*, Kluwer, Amsterdam, pp 124–57.
8. Hoffman K, Padberg M (1993) Solving airline crew

- scheduling problems by branch and cut. *Manag Sci* **29**, 657–82.
9. Capanera P, Gallo G (2004) A multicommodity flow approach to the crew rostering problem. *Oper Res* **52**, 583–96.
 10. Jeenanunta C, Intiyot B, Puttapatimok W (2010) A multi-commodity flow approach to the crew rostering problem. In: *Proceedings of the 2nd International Conference on Logistic and Transport (ICLT 2010) and the 1st International Conference on Business and Economics (ICBE 2010)*, New Zealand, pp 735–42.
 11. Lučić P, Teodorović D (2007) Metaheuristics approach to the aircrew rostering problem. *Ann Oper Res* **155**, 311–38.
 12. El Moudani W, Cosenza CAN, Mora-Camino F (2001) An intelligent approach for solving the airline crew rostering problem. In: *ACS/IEEE International Conference on Computer Systems and Applications*, pp 73–9.
 13. Souai N, Teghem J (2009) Genetic algorithm based approach for the integrated airline crew-pairing and rostering problem. *Eur J Oper Res* **199**, 674–83.
 14. Limlawan V, Kasemsontitum B, Jeenanunta C (2011) Airline crew rostering problem using particle swarm optimization. In: *IEEE International Conference on Quality and Reliability*, Thailand, pp 501–5.
 15. Zhang Y, Rao Y, Zhou M (2007) GASA hybrid algorithm applied in airline crew rostering system. *Tsinghua Sci Tech* **12(SUPPL. 1)**, 255–9.
 16. Maenhout B, Vanhoucke M (2010) A hybrid scatter search heuristic for personalized crew rostering in the airline industry. *Eur J Oper Res* **206**, 155–67.
 17. Kohl N, Karish S (2004) Airline crew rostering: problem types, modeling and optimization. *Ann Oper Res* **127**, 223–57.
 18. Ernst AT, Jiang H, Krishnamoorthy M, Sier D (2004) Staff scheduling and rostering: A review of applications, methods and models. *Eur J Oper Res* **153**, 3–27.
 19. Kato T, Jeenanunta C (2010) Crew rostering problem: case of thai domestic low cost airline. In: *Proceedings of the 10th International Conference on Industrial Management*, pp 480–5.
 20. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*, USA, pp 1942–8.
 21. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: *Proceedings of the 6th International Symposium on Micro Machines and Human Science*, Japan, pp 39–43.
 22. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: *Proceedings of the IEEE International Conference on Evolutionary Computation*, USA, pp 69–73.
 23. Eberhart R, Shi Y (2001) Particle swarm optimization: developments, application and resources. In: *Proceedings of the 2001 Congress on Evolutionary Computation*, pp 81–6.
 24. Hu X, Shi Y, Eberhart R (2004) Recent advance in particle swarm. In: *IEEE Congress on Evolutionary Computation*, USA, pp 90–7.
 25. Kennedy J, Eberhart R (1997) A discrete binary version of the particle swarm algorithm. In: *IEEE International Conference on Computational Cybernetics and Simulation*, pp 4104–8.