Solving Airport Gate Allocation Problem using Simulated Kalman Filter

Zulkifli Md Yusof, Siti Nurzulaikha Satiman, Kamil Zakwan Mohd Azmi, Badaruddin Muhammad, Saifudin Razali, and Zuwairie Ibrahim
Faculty of Electrical and Electronics Engineering
Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia
zuwairie@ump.edu.my

Zulfakar Aspar
Faculty of Electrical Engineering
Universiti Teknologi Malaysia Pahang, 81310 Johor Bahru, Johor, Malaysia

Suraya Ismail
Wise Aviation Sdn. Bhd.

Abstract

Airport gate allocation problem refers to the search for optimal assignment of flights to gates at an airport. Assignment of flight to gates has become very complex nowadays, especially for a big size airport. In this study, the airport gate allocation problem is solved using a recently introduced meta-heuristic called simulated Kalman filter (SKF). The SKF is driven by the estimation capability of a well-known Kalman filter. The objective of this study is to minimize the total walking distance. Since the airport gate allocation problem is a combinatorial optimization problem, the original SKF is extended such that it can be used to solve combinatorial optimization problems. A small case study with 15 flights and 16 gates has been chosen. Preliminary results show that SKF is a promising algorithm for solving the airport gate allocation problem.

Keywords: Airport Scheduling, Gate Assignment, Simulated Kalman Filter, Walking Distance.

1. Introduction

Various approaches have been employed for solving airport gate allocation problems (AGAP) such as linear programming (Bihr, 1990), tabu search (Xu & Bailey, 2001), simulated annealing (Drexl & Nikulin, 2008), coevolutionary (Silva Garza and Campos Licastro, 2013), and genetic algorithm (Li & Xu, 2012) (Hu & Di Paolo, 2007) (Gu & Chung, 1999). Inspired by the estimation capability of Kalman filter, we have recently introduced a novel estimation-based optimization algorithm called simulated Kalman filter (SKF) (Ibrahim et al., 2015). Since the SKF algorithm has not been applied in AGAP, this paper introduces the first application of SKF algorithm in AGAP.

The objective of AGAP is to find the optimal assignment of flights to gates at an airport. The quality of the assignment can be evaluated based on certain objectives, such as to minimize the total connection times by passengers (Xu & Bailey, 2001) (Ding, Lim, Rodrigues, and Zhu, 2004) or to minimize the total walking distance (Silva Garza and Campos Licastro, 2013).

This paper is organized as follows. At first, SKF will be briefly introduced followed by the description of the implementation of SKF in solving AGAP. Lastly, a conclusion will be provided at the end of this paper.

2. Simulated Kalman Filter

Every agent in SKF is regarded as a Kalman filter. Based on the mechanism of Kalman filtering and measurement process, every agent estimates the global minimum/maximum. Measurement, which is required in Kalman filtering, is mathematically modelled and simulated. Agents communicate among them to update and improve the solution during the search process. The simulated Kalman filter (SKF) algorithm is illustrated in Figure 1.

Consider n number of agents, SKF algorithm begins with initialization of n agents, in which the states of each agent are given randomly. The maximum number of iterations, t_{max}, is defined. The initial value of error covariance estimate, $P(0)$, the process noise value, Q, and the measurement noise value, R, which are required in Kalman filtering, are also defined during initialization stage.

Then, every agent is subjected to fitness evaluation to produce initial solutions $\{\boldsymbol{X}_1(0), \boldsymbol{X}_2(0), \boldsymbol{X}_3(0), \ldots, \boldsymbol{X}_{n-2}(0), \boldsymbol{X}_{n-1}(0), \boldsymbol{X}_n(0)\}$. The fitness values are compared and the agent having the best fitness value at every iteration, t, is registered as $\boldsymbol{X}_{\mathbf{best}}(t)$. For function minimization problem,

$$\boldsymbol{X}_{\mathbf{best}}(t) = \min_{i \in 1,\ldots,n} fit_i(\boldsymbol{X}(t)) \quad (1)$$

whereas, for function maximization problem,

$$\boldsymbol{X}_{\mathbf{best}}(t) = \max_{i \in 1,\ldots,n} fit_i(\boldsymbol{X}(t)) \quad (2)$$

The-best-so-far solution in SKF is named as $\boldsymbol{X}_{\mathbf{true}}$. The $\boldsymbol{X}_{\mathbf{true}}$ is updated only if the $\boldsymbol{X}_{\mathbf{best}}(t)$ is better (($\boldsymbol{X}_{\mathbf{best}}(t) < \boldsymbol{X}_{\mathbf{true}}$ for minimization problem, or $\boldsymbol{X}_{\mathbf{best}}(t) > \boldsymbol{X}_{\mathbf{true}}$ for maximization problem) than the $\boldsymbol{X}_{\mathbf{true}}$.

The subsequent calculations are largely similar to the predict-measure-estimate steps in Kalman filter. In the prediction step, the following time-update equations are computed.

$$\boldsymbol{X}_{\boldsymbol{i}}(t|t) = \boldsymbol{X}_{\boldsymbol{i}}(t) \quad (3)$$

$$P(t|t) = P(t) + Q \quad (4)$$

where $\boldsymbol{X}_i(t)$ and $\boldsymbol{X}_i(t|t)$ are the current state and current transition/predicted state, respectively, and $P(t)$ and $P(t|t)$ are the current error covariant estimate and current transition error covariant estimate, respectively. Note that the error covariant estimate is influenced by the process noise, Q.

The next step is measurement, which is a feedback to estimation process. Measurement is modelled such that its output may take any value from the predicted state estimate, $\boldsymbol{X}_i(t|t)$, to the true value, $\boldsymbol{X}_{\text{true}}$. Measurement, $\boldsymbol{Z}_i(t)$, of each individual agent is simulated based on the following equation:

$$\boldsymbol{Z}_{\boldsymbol{i}}(t) = \boldsymbol{X}_{\boldsymbol{i}}(t|t) + \sin(rand \times 2\pi) \times |\boldsymbol{X}_{\boldsymbol{i}}(t|t) - \boldsymbol{X}_{\boldsymbol{true}}| \quad (5)$$

The $\sin(rand \times 2\pi)$ term provides the stochastic aspect of SKF algorithm and $rand$ is a uniformly distributed random number in the range of [0,1].

The final step is the estimation. During this step, Kalman gain, $K(t)$, is computed as follows:

$$K(t) = \frac{P(t|t)}{P(t|t)+R} \quad (6)$$

Then, the estimation of next state, $\boldsymbol{X}_i(t+1)$, and the updated error covariant are computed based on Eqn. (7) and Eqn. (8), respectively.

$$\boldsymbol{X}_i(t+1) = \boldsymbol{X}_i(t|t) + \Delta_i \quad (7)$$

$$P(t+1) = \big(1 - K(t)\big) \times P(t|t) \quad (8)$$

where $\Delta_i = K(t) \times (\boldsymbol{Z}_i(t) - \boldsymbol{X}_i(t|t))$. Finally, the next iteration is executed until the maximum number of iterations, t_{max}, is reached.

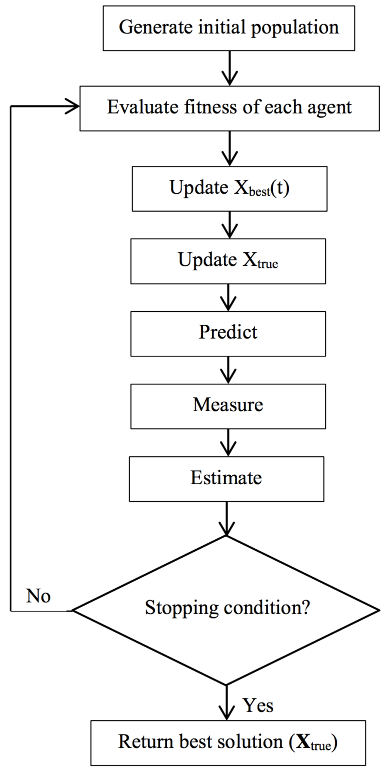

Figure 1. The original simulated Kalman filter (SKF) algorithm.

In order to solve a combinatorial optimization problem using SKF, the Δ_i term in Eqn. (7) is mapped into a probabilistic value [0,1]. Then the probabilistic value will be compared with a random number [0,1] to update a bit string or solution.

For solving combinatorial optimization problem using SKF, most of the calculations are similar to the original SKF. Modifications are needed only during initialization and generation of solution to combinatorial optimization problem.

During the initialization of agents, a random bit string, Σ_i , is generated for each agent. Each bit in the bit string is associated to a dimension. The length of the bit string is problem dependent and subjected to the size of the problem.

In binary gravitational search algorithm (BGSA) (Rashedi, Nezamabadi & Saryazdi, 2010), a function shown in Figure 2 is used to map a velocity value into a probabilistic value within interval [0,1]. Similar function is used in this study. The term Δ_i is mapped to a probabilistic value within interval [0,1] using a probability function, $S(\Delta i(t))$, as follows:

$$S\big(\Delta i(t)\big) = \big|\tanh\big(\Delta i(t)\big)\big| \quad (9)$$

After the $S\big(\Delta i(t)\big)$ is calculated, a random number, *rand*, is generated and a binary value at dimension d of an ith agent, Σ_i^d, is updated according to the following rule:

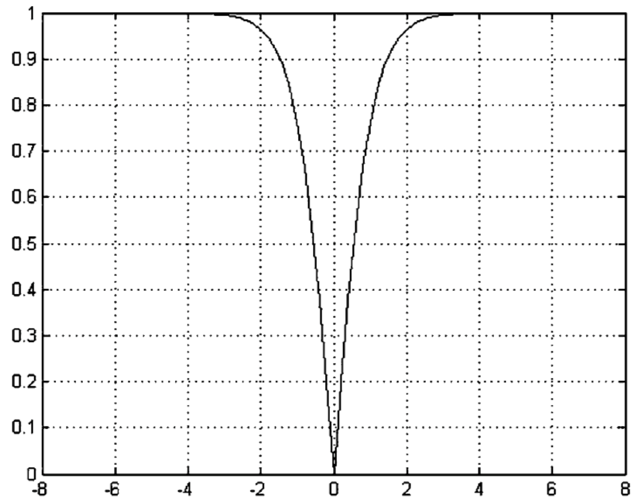

Figure 2. A probabilistic function (Rashedi, Nezamabadi & Saryazdi, 2010).

Table 1. Flight arrival from Penang.

Departure (PENANG)	Arrive (KLIA)	Flight Number
06:50	07:50	F2
08:35	09:35	F1
11:20	12:20	F2
13:10	14:10	F1
14:10	15:10	F2
17:10	18:10	F1
18:20	19:20	F2
19:25	20:25	F3
20:55	21:55	F1
23:30	00:20	F2

Table 2. Flight arrival from Kota Kinabalu.

Departure (KOTA KINABALU)	Arrive (KLIA)	Flight Number
07:10	09:40	F5
09:30	12:10	F6
10:55	13:25	F4
12:30	15:00	F7
15:45	18:15	F8
18:30	21:00	F9
18:55	21:25	F5
19:25	21:55	F6
20:10	22:45	F4
21:30	23:59	F7

Table 3. Flight arrival from Kuching.

Departure (KUCHING)	Arrive (KLIA)	Flight Number
07:00	08:45	F11
09:20	11:00	F10
10:40	12:25	F13
11:55	13:45	F12
15:05	16:45	F11
16:15	17:55	F10
17:35	19:20	F13
18:30	20:15	F12
19:10	21:00	F14
19:45	21:25	F15
22:50	00:50	F10

if $rand < S(\Delta i(t))$
then $\Sigma_i^d(t+1) = \text{complement}\Sigma_i^d(t+1)$
else $\Sigma_i^d(t+1) = \Sigma_i^d(t+1)$
end (10)

3. Airport Gate Allocation Problem

The AGAP involves a number of parameters as follows:

n number of flights
m number of gates
w number of time windows
ω_k walking distance between gate k and entrance/exit

Once a flight arrived at an assigned gate, the passengers have to walk from the date to the entrance/exit hall. In this study, the objective is to minimize the total walking distance between gates and an entrance/exit. This objective is mathematically represented as

$$min \sum_{i=1}^{w} x_{ik}\, \omega_k$$

where
$x_{ik} \in \{0,1\}$ 1 iff time window i is assigned to gate k; 0 otherwise.

In this paper, only flights arrive from Penang, Kota Kinabalu, and Kuching at KLIA are considered as a case study. The arrival time of those flights are tabulated in Table 1, Table 2, and Table 3. This information is obtained from Malaysian Airlines website (http://www.malaysiaairlines.com). Based on this case study, the number of flights, n = 15. Also, number of gates, m = 16, is considered and the distance between gates (in meter) to entrance/exit is shown in Table 4.

Table 4. Distance between gates to entrance/exit.

Gate Number	Distance
GATE 0	1019.5m
GATE 1	1328.6m
GATE 2	1650.0m
GATE 3	2015.6m
GATE 4	2405.2m
GATE 5	2809.0m
GATE 6	3221.6m
GATE 7	3640.1m
GATE 8	1019.5m
GATE 9	1328.6m
GATE 10	1650.0m
GATE 11	2105.6m
GATE 12	2406.22m
GATE 13	2809.0m
GATE 14	3221.6m
GATE 15	3640.1m

In this study, a time window is defined as a duration (from the arrival to the next departure) of a flight at an airport. The time windows are obtained based on the information in Table 1, Table 2, and Table 3. A time window is associated to a flight as shown in Table 5.

4. Application of Simulated Kalman Filter in Airport Gate Allocation Problem

The AGAP is a combinatorial optimization problem. Many combinations of time window to gate number matching exist and the best or optimal matching can be obtained by employing meta-heuristic algorithms such as SKF. In this study, solution representation as shown in Figure 3 is employed. This solution representation requires binary assignment of gates and the assignment is shown in Table 6. Since 16 gates involve in this case study, a 4-bit string is required to represent a gate.

The SKF algorithm for combinatorial optimization problem, as explained in section 2, is employed in solving AGAP. The experimental setting for SKF is shown in Table 7.

Figure 4 and Figure 5 shows two examples of convergence curve based on different maximum iteration values. The convergence curves show that the generation of better gate assignment can be obtained using SKF algorithm. Example of solution produced by SKF algorithm is shown in Table 8 where the shortest walking distance is 6.0337m.

Table 5. Time window, flight, and duration.

Time Window	Flight	Duration
TW1	F4	6AM – 10AM
TW2	F1	7AM – 9AM
TW3	F10	7AM – 9AM
TW4	F7	7AM – 11AM
TW5	F2	8AM – 10AM
TW6	F12	8AM – 11AM
TW7	F11	9AM – 12PM
TW8	F8	9AM – 1 PM
TW9	F1	10AM – 11AM
TW10	F9	10AM – 1PM
TW11	F10	12PM – 3PM
TW12	F2	12PM – 2PM
TW13	F5	12PM – 4PM
TW14	F13	2PM – 4PM
TW15	F1	2PM – 4PM
TW16	F6	3PM – 7PM
TW17	F2	3PM – 5PM
TW18	F12	3PM – 6PM
TW19	F4	4PM – 7PM
TW20	F3	5PM – 6PM
TW21	F7	5PM – 8PM
TW22	F11	5PM – 8PM
TW23	F10	6PM – 9PM
TW24	F8	6PM – 10PM
TW25	F1	7PM – 9PM
TW26	F13	8PM – 10PM
TW27	F9	9PM – 12AM
TW28	F2	9PM – 10PM
TW29	F16	9PM – 12AM
TW30	F3	10PM – 11PM

5. Conclusion

This paper introduced the first application of SKF in AGAP. The original SKF algorithm is extended such that it can be used for solving combinatorial optimization problems such as AGAP. A small scale problem which consists of 15 flights, 16 gates, and an entrance/exit is considered in the experiments. Simulation results show that the SKF algorithm is able to minimize the total walking distance and optimal matching between flights and gates can be obtained.

The next step is to consider larger scale of problem with more flights and gates. Also, the fitness function used for the evaluation of flight-gate matching shall be improved to obtain robust solution. Different variant of SKF algorithms for combinatorial optimization problems will be considered and tested to find the best algorithm for AGAP.

6. Acknowledgement

This work is financially supported by the Knowledge Transfer Program (KTP) Fund (RDU141012) awarded by Ministry of Higher Education to Universiti Malaysia Pahang.

TW1	TW2	TW3	TW4	•••	•••	•••	TW29	TW30
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX

X is either 0 or 1

Figure 3. Solution representation.

Table 6. Binary assignment of each gate.

Binary Representation	Gate Number
0000	G0
0001	G1
0010	G2
0011	G3
0100	G4
0101	G5
0110	G6
0111	G7
1000	G8
1001	G9
1010	G10
1011	G11
1100	G12
1101	G13
1110	G14
1111	G15

Table 7. Experimental setting parameters.

SKF Parameters	
Parameter	**Value**
Error covariant, *P*	1000
Process noise, *Q*	0.5
Measurement noise, *R*	0.5
Rand	[0,1]
$x_{\min}$	-100
$x_{\max}$	100
Number of agents	100

Table 8. Example of solution.

Time Window	Gate	Time Window	Gate
TW1	G15	TW16	G13
TW2	G10	TW17	G9
TW3	G13	TW18	G10
TW4	G11	TW19	G11
TW5	G2	TW20	G12
TW6	G6	TW21	G8
TW7	G9	TW22	G6
TW8	G13	TW23	G0
TW9	G8	TW24	G15
TW10	G2	TW25	G2
TW11	G4	TW26	G1
TW12	G14	TW27	G10
TW13	G8	TW28	G13
TW14	G4	TW29	G9
TW15	G1	TW30	G5

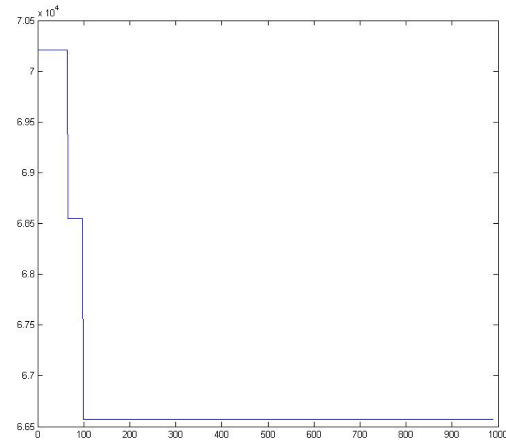

Figure 4. Convergence curve based on 100 iterations. In this figure, the y-axis is the total walking distance in meter (m) and x-axis is the iteration number.

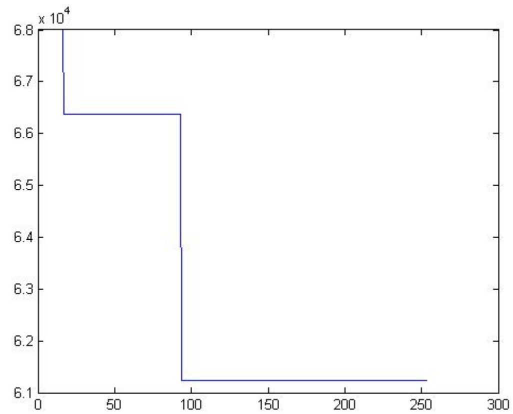

Figure 5. Convergence curve based on 250 iterations. In this figure, the y-axis is the total walking distance in meter (m) and x-axis is the iteration number.

References

Bihr, R. (1990). A Conceptual Solution to the Aircraft Gate Assignment Problem using 0,1 Linear Programming. *Computers & Industrial Engineering*. 19: 280–284.

Ding, H., Lim, A., Rodrigues, B. and Zhu, Y. (2004). Aircraft and Gate Scheduling Optimization at Airports. *In Proceedings of the 37th Hawaii International Conference on System Sciences.*

Drexl, A. and Nikulin, Y. (2008) Multicriteria Airport Gate Assignment and Pareto Simulated Annealing. *IIE Transactions*. 40: 385–397.

Gu, Y. and Chung, C. A. (1999) Genetic Algorithm Approach to Aircraft Gate Reassignment Problem. *Journal of Transportation Engineering*. 125: 384–389.

Hu, X. B. and Di Paolo, E. (2007) An Efficient Genetic Algorithm with Uniform Crossover for the Multi-Objective Airport Gate Assignment Problem. *In Proceedings of the IEEE Congress on Evolutionary Computation (CEC07),* 55–62.

Ibrahim, Z., Abdul Aziz, N. H., Ab Aziz, N. A., Razali, S., Shapiai, M. I., Nawawi, S. W., and Mohamad, M. S. (2015). A Kalman Filter Approach for Solving Unimodal Optimization Problems. *ICIC Express Letters* (accepted).

Li, W. and Xu, X. (2012) Optimized Assignment of Airport Gate Configuration based on Immune Genetic Algorithm. *Measuring Technology and Mechatronics Automation in Electrical Engineering*. 347–355.

Rashedi, E., Nezamabadi-Pour H., and Saryazdi, S. (2010). BGSA: Binary Gravitational Search Algorithm. *Natural Computing*. 9: 727-745.

Silva Garza, A. G. and Campos Licastro, A. P. T. (2003). A Coevolutionary Method for Automating Airport Gate Scheduling. *Proceedings of the Fourth Mexican International Conference on Computer Science*.

Xu, J. and Bailey, G. (2001). The Airport Gate Assignment Problem: Mathematical Model and a Tabu Search Algorithm. *In Proceedings of the 34th Hawaii International Conference on System Sciences.*