Research Journal of Applied Sciences, Engineering and Technology 6(12): 2136-2140, 2013
DOI:10.19026/rjaset.6.3837
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: December 3, 2012 Accepted: January 19, 2013 Published: July 30, 2013

Research Article

Hybrid Genetic-cuckoo Search Algorithm for Solving Runway Dependent Aircraft Landing Problem

[1]Hongqing Zheng, [1, 2]Yongquan Zhou and [1]Peigang Guo
[1]College of Information Science and Engineering, Guangxi University for Nationalities,
Nanning 530006, China
[2]Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Nanning 530006, China

Abstract: As the demand for air transportation continues to grow, some flights cannot land at their preferred landing times because the airport is near its runway capacity. Therefore, devising a method for tackling the Aircraft Landing Problem (ALP) in order to optimize the usage of existing runways at airports is the focus of this study. This study, a hybrid Genetic-Cuckoo Search (GCS) algorithm for optimization the ALP with runway is proposed. The numerical results showed that the proposed GCS algorithm can effectively and efficiently determine the runway allocation, sequence and landing time for arriving aircraft for the three test cases by minimizing total delays under the separation constraints in comparison with the outcomes yielded by previous studies.

Keywords: Aircraft traffic control, cuckoo search algorithm, genetic algorithm, runway scheduling

INTRODUCTION

Due to rapid global economic growth in recent decades, air transportation of both passengers and cargo has dramatically increased, leading to the problem of congestion at airports. Therefore, devising a method for tackling the Aircraft Landing Problem (ALP) in order to optimize the usage of existing runways at airports is the focus of this study.

The ALP is a type of NP-hard problem that can be viewed as a job machine scheduling problem with release times and sequence dependent processing time (Beasley *et al*., 2000). In brief, the objective of the ALP is to determine the best combination of assigning the sequence and the corresponding landing time for a given set of aircraft to a runway, which in turn minimizes the sum of the deviations of the actual and preferred landing times of aircraft by simultaneously satisfying the minimum separation requirement between landing aircraft. Two types of ALP, runway independent versus runway dependent, have been investigated in previous studies. The runway independent ALP assumes that the earliest landing time is independent of the runway assigned for landing. However, this assumption rarely holds in real-life situations, due to the fact that the aircraft close to the airport will have different earliest times for different runways, depending on the respective directions of the specific flight and runway. Therefore, this study focuses on investigating and developing algorithmic procedures for effectively and efficiently solving the runway dependent ALP.

During the past decade, several researchers have proposed solution procedures to search for optimal runway assignment and scheduling for the runway dependent ALP, based on different metaheuristics, such as genetic algorithms (Cheng *et al*., 1999; Hansen, 2004), scatter search and bionomic algorithms (Pinol and Beasley, 2006). Recently, a new heuristic search algorithm, called Cuckoo Search (CS) (Yang and Deb, 2009), has been developed by Yang and Deb (2009), has been shown to yield promising outcomes for solving various engineering optimization problem (Yang and Deb, 2010) and so this study devises and tests an optimization procedure based on this framework for solving runway dependent ALP. Essentially, this study designs a search procedure by incorporating initialization (for randomly generating the initial population), swap exchange, selection scheme and a single-point crossover operator into the CS framework. Each solution needs to be decoded, followed up by its objective function being calculated. To validate the effectiveness and efficiency of the proposed algorithmic procedure, test cases used in previous studies (Hansen, 2004; Pinol and Beasley, 2006; Yang and Deb, 2009) are adopted for the purpose of comparison. The comparative results can be used to demonstrate the potential of the proposed algorithm with regard to solving the runway dependent ALP.

In this study, we aim to develop a hybrid Genetic-Cuckoo Search (GCS) algorithm for solving the ALP

Corresponding Author: Yongquan Zhou, College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China

with runway dependent attributes. A set of numerical experiments were conducted to test the validity of the proposed algorithm based on three test instances created and investigated by previous studies. The numerical results showed that the proposed GCS algorithm can effectively and efficiently determine the runway allocation, sequence and landing time for arriving aircraft for the three test cases by minimizing total delays under the separation constraints in comparison with the outcomes yielded by previous studies.

THE MATHEMATICAL MODEL OF THE RUNWAY DEPENDENT ALP

With reference to the general mathematical formulation of ALP defined by Beasley *et al.* (2000), Cheng *et al.* (1999) and Yang and Deb (2009), this section depicts the mathematical model of the runway dependent ALP. The runway dependent ALP aims at finding the best arrangement of sequences, runway and corresponding landing time for a given set of landing aircraft to minimize total delay by following separation requirements. To illustrate the mathematical formulation, some notations are defined as follows: A denotes the set of aircrafts waiting to land (A = {1 ,…, n}, |A| = n); R represents the set of runways available for landing R = {1 ,…, m}, |R| = m); E_{ik} is the earliest landing time of aircraft i landing on runway k (∀ i ∈ A, ∀ k ∈ R); τ_i represents the minimum earliest landing time of all available landing runways for aircraft i (i.e., τ_i = min $\{E_{ik}$, k = 1,…,$m\}$ S_{ij}; denotes the minimum separation time required if aircraft i lands before j ($S_{ij} \geq 0$, ∀i, j ∈ A) on the same runway. In this study, all the aircraft scheduled to land on a specific runway should assure specific separation rules between the leading aircraft and the trailing aircraft based on aircraft types. Three aircraft types are used in this study. They are heavy (H), large (L) and Small (S). x_i represents the scheduled landing time for aircraft i; λ_{ik} and γ_{ij} are the decision indicator variables. λ_{ik} equals 1 if aircraft i lands on runway k, with λ_{ik} equal to 0 otherwise; γ_{ij} equals 1 if aircraft i and j land on the same runway, with γ_{ij} equal to 0 otherwise. Hence, the mathematical model of the runway dependent ALP can be stated as below:

$$\min \; Z = \sum_{i=1}^{n} (x_i - \tau_i)^2 \tag{1}$$

$$s.t. \; x_i \geq \lambda_{ik} E_{ik} ., \forall i \in A, \forall k \in R \tag{2}$$

$$x_j \geq x_i + S_{ij}\gamma_{ij}, \text{if} \, x_i \leq x_j, \forall i, j \in A \tag{3}$$

$$\sum_{k=1}^{m} \lambda_{ik} = 1, \quad \forall i \in A, \forall k \in R \tag{4}$$

$$x_i \text{ is interger}, \; \forall i \in A \tag{5}$$

$$\lambda_{ik}, \gamma_{ik} \in \{0,1\}, \; \forall i, j \in A, \forall k \in R \tag{6}$$

The objective function defined by a previous study (Pinol and Beasley, 2006) is the sum of squared deviations of the scheduled landing time (x_i) and the minimum earliest landing time (τ_i) of each aircraft, as shown in Eq. (1). Equations (2) to (6) pertain to the constraints of the model. Equation (2) assures that every landing time must be set after the earliest landing time of aircraft i landing on runway k. Equation (3) ensures that the separation times between aircrafts are satisfied. Equation (4) guarantees that each aircraft is only assigned to one runway. Equations (5) and (6) represent the feasible values of the variables. For solving the runway dependent ALP, the number of aircrafts (n) and runways (m) are fixed and given together with all information about each aircraft (i.e., aircraft type, E_{ik}). The required separation time for any combination of aircraft types in terms of leading and trailing aircrafts is also specified. A solution for the runway dependent ALP can then be defined by specifying the scheduled landing time and runways for all aircrafts.

HYBRID GENETIC-CS FOR SOLVING RUNWAY DEPENDENT ALP

The study proposes an optimization procedure based on cuckoo search for solving runway dependent ALP. The GCS algorithm is hybrid genetic-cuckoo algorithm achieved by incorporating genetic algorithm into a cuckoo search framework. The algorithm is inspired by the reproduction strategy of cuckoos. At the most basic level, cuckoos lay their eggs in the nests of other host birds, which may be of different species. The host bird may discover that the eggs are not it's own and either destroy the egg or abandon the nest all together. This has resulted in the evolution of cuckoo eggs which mimic the eggs of local host birds. When using the CS to solve an optimization problem, the decision variables are commonly encodes as nest. These nests represent a solution. The entire population of such nests represents a generation. Each generation will create a new set of nests by using the fitter nests form the previous generation.

An example with 3 runways and 12 decision variables is used to illustrate the above terms, as shown in Fig. 1. There are 12 genes encoded as 1, 2, 3, 3, 3, 2, 1, 1, 1, 2, 2 and 3. The first gene represents aircraft 1 landing on runway 1; the second gene represents

1	2	3	3	3	2	1	1	1	2	2	3

Runway #1: 1 7 8 9 (the number of aircraft)

Runway #2: 2 6 10 11

Runway #3: 3 4 5 12

Fig. 1: A nest representation of twelve decision variables for the runway dependent ALP

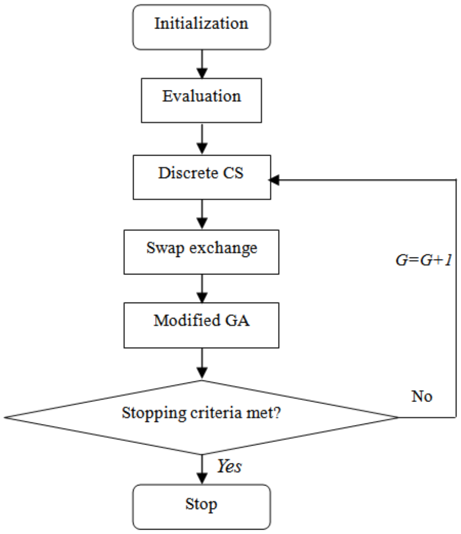

Fig. 2: The general procedure of the GCS for solving the runway dependent ALP

aircraft 2 landing on runway 2 and so on. A nest representing a solution is composed of 12 genes.

Based on the above nest representation, the GCS implemented in this study consist of seven components, as follows: initialization, sequencing and scheduling methods (Yu-Hsin, 2011), discrete cuckoo search algorithm, swap exchange, objective function evaluation, modified genetic algorithm and termination checking. The general procedure of the GCS for solving the runway dependent ALP is illustrated in Fig. 2. In the first step, during initialization, an initial population is randomly generated. For each initial solution, the scheduled landing times of aircraft can be assigned based on sequencing and scheduling methods, by which the corresponding objective function (or fitness) values can be calculated. The discrete cuckoo search is then used to fine-tune the initial population. The objective function (or fitness) values form the basis for comparing all the nests in the population against each other. The fitter ones will be selected to propagate to the next generation. To further increase the computational efficiency the swap exchange is applied to fine-tune the newly generated solutions from the discrete cuckoo search procedure, a modified genetic algorithm is applied. The population of strings (solutions) is allowed to evolve through successive generations until a termination criterion is met.

MODIFIED CUCKOO SEARCH

Original CS: CS is a heuristic search algorithm which has been proposed recently by Yang and Deb (2009). The algorithm is inspired by the reproduction strategy of cuckoos. At the most basic level, cuckoos lay their eggs in the nests of other host birds, which may be of different species. The host bird may discover that the eggs are not it's own and either destroy the egg or abandon the nest all together. This has resulted in the evolution of cuckoo eggs which mimic the eggs of local host birds. To apply this as an optimization tool, Yang and Deb (2010) used three ideal rules:

- Each cuckoo lays one egg, which represents a set of solution co-ordinates, at a time and dumps it in a random nest
- A fraction of the nests containing the best eggs, or solutions, will carry over to the next generation
- The number of nests is fixed and there is a probability that a host can discover an alien egg. If this happens, the host can either discard the egg or the nest and this result in building a new nest in a new location.

When generating new solution $x^{(t+1)}$ for, say cuckoo i, a Lévy flight is performed:

$$x_i^{(t+1)} = x_i^{(t)} + \partial \oplus \text{Lévy}(\beta) \quad (7)$$

where, $\partial>0$ is the step size which should be related to the scales of the problem of interests. In most cases, we can use $\partial = 1$.

Discrete CS: Aiming at the particularity of the discrete space optimization problem such as runway dependent ALP, we adapt discrete CS to our problem. The basic CS and its variants have successfully operated for continuous optimization problems. In order to extend the application to discrete space, we proposed a discrete version of CS. Where a nest moves in a state space restricted to the integer between 1 to m on each dimension where each step size (the step length produced by the Lévy flight) represents the probability of bit step size taking 1. Thus, the nest trajectories are defined as the change in the probability and step size is a measure of individual current probability of taking 1. If the step size is higher it is more likely to choose 1 and lower values favor choosing 0. A sigmoid function is applied to transform the step size from real number space to probability space:

$$s(stepsize_{jd}) = \frac{1}{1+\exp(-stepsize_{jd})} \quad (8)$$

$$mv = \begin{cases} 1 & if \quad rand \le s(stepsize_{jd}) \\ 0 & \quad otherwise \end{cases} \quad (9)$$

$$x_i^{(t+1)} = \mathrm{mod}(x(i) + mv, m) + 1 \quad (10)$$

General description of hybrid genetic-CS: The proposed algorithm for the solution of the runway dependent ALP combines a GA and a Cuckoo search algorithm. In the following, the outline of the proposed algorithm is presented.

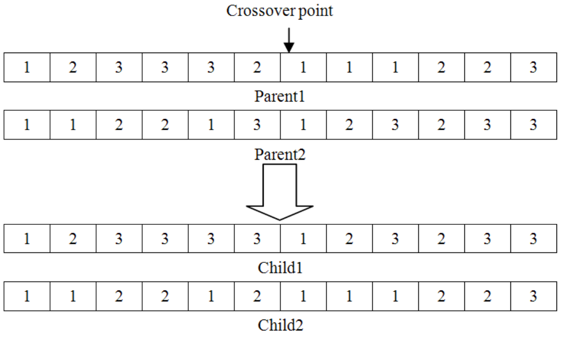

Fig. 3: A pair of string with 12 genes before and after the crossover operator

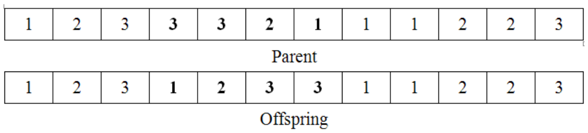

Fig. 4: An example of inversion node

Initialization:

- The initialization of a nest is performed by randomly selecting a runway from the available runways for each aircraft.
- Evaluate the fitness of each individual.

Main algorithm: Set the number of generations equal to zeros.

Do while stopping criteria are not satisfied

Discrete cuckoo search algorithm

Calculate the step size of nest i according to levy flight.

Calculate the $x_i^{(t + 1)}$ of nest I according to Eq. (8), (9), (10).

Evaluate its quality/fitness

Swap exchange

Evaluate its quality/fitness

Modified genetic algorithm

Select two parents from the current population via roulette wheel selection.

Apply the crossover operator between the two parents (Fig. 3).

Improve each offspring by the mutation operation (inversion node as shown in Fig. 4) and insert the resulting offspring to the new population.

Enddo: Return the best individual.

COMPUTATIONAL RESULTS

In this study, all computational experiments are conducted with Matlab R2010a and run on Celeron(R) Dual-core CPU T3100, 1.90GHZ with 2GB memory capacity. The essential parameters of GCS model for the runway dependent ALP is set as follows. Let the maximal number of iterations be 200, the number of nests be 30 and probability of crossover is 0.85.

Numerical experiments are conducted to investigate the performance of the proposed GCS algorithm. Test cases used in Hansen (2004), Pinol and Beasley (2006) and Yu-Hsin (2011) are used and their results are included as benchmarks for the purpose of comparison. H1-12-3 is the first test case used in Hansen (2004) with 12 aircraft and three runways; H2-15-3 represents the second test case in Hansen (2004) with 15 aircraft and three runways; H3-20-5 signifies the third test case used in Hansen (2004) with 20 aircraft and five runways. the numerical results of the GCS algorithm based on the parameter values are used to assess the effectiveness of the proposed GCS algorithm compared with GA, Scatter Search (SS), a Binominal Algorithm (BA) and a genetic local search algorithm with a threshold accepting mechanism (GLS), which were tested in previous studies (Hansen, 2004; Pinol and Beasley, 2006; Yang and Deb, 2009) with regard to their effectiveness in solving the runway dependent ALP. Several significant results were obtained, as shown in Table 1, further confirm that proposed GCS algorithm can find the currently known minimum values for these test cases. In terms of H2-15-3, our result is better than in the literature ((Pinol and Beasley, 2006), the final landing schedule is shown in Table 2. As for H3-20-5, our result is better than GA,

Table 1: Comparison of the results for runway dependent ALP problems

Test case	Best val	GA (a)	SS (b)	BA (b)	GLS (c)	GCS (d)
H1-12-3	3.5	3.50	3.75	3.75	3.5	3.5
H2-15-3	12.25	12.5	12.25	12.25	12.25	12.25
H3-20-5	7.75	12.00	8.75	9.75	7.75	7.75

(a) Algorithm proposed and examined by Hansen (2004); (b) algorithm proposed and examined by Pinol and Beasley (2006); (c) algorithm proposed and examined by Yu-Hsin (2011); (d) algorithm proposed and examined in this study

Table 2: Best schedule for the 15-aircraft, 3-runway scenario

Runway#1	STA	Delay	Runway#2	STA	Delay	Runway#3	STA	Delay
DL1920	6	0	NW410	7	1	UA410	4	0
SW200	7	1	DL200	8.5	1.5	NW2123	5	0
DL3319	8	1				SW185	6.5	0.5
DL510	10	1				AA1225	7.5	0.5
UA1133	11	1				UA123	8.5	1.5
AA335	15	0				DL130	10	0
AA205	16.5	1.5						

Table 3: Best schedule for the 20-aircraft, 5-runway scenario

Runway#1	NW410	DL200	DL510	UA1133	AA335	AA205
STA	6	7.5	9	10	15	16.5
Delay	0	0.5	0	0	0	1.5
Runway#2	SW200	DL3319				
STA	7	8				
Delay	1	1				
Runway#3	UA410	SW185	AA1225	UA123	SW442	
STA	4	6	7	8	10	
Delay	0	0	0	1	0	
Runway#4	AA127	DL1920	UA555	SW250		
STA	6	7	8	9.5		
Delay	0	1	1	0.5		
Runway#5	NW2123	AA1410	DL130			
STA	5	7	9			
Delay	0	0	0			

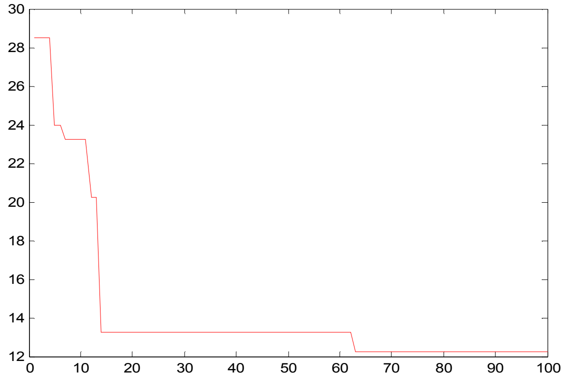

Fig. 5: The convergence curve of H2-15-3

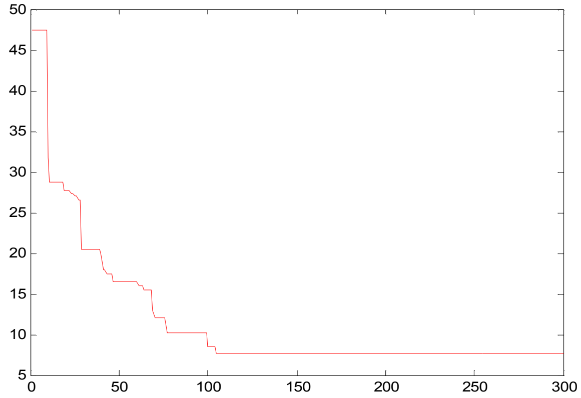

Fig. 6: The convergence curve of H3-20-5

SS, BA and it is identical to GLS, the final landing schedule is shown in Table 3. The convergence curve of H2-15-3 and H3-20-5 are shown in Fig. 5 and 6, respectively.

CONCLUSION

In this study, GCS is proposed to solve the runway dependent ALP. The results from computational experiments support the superiority of GCS in terms of finding the objective function values and computational time, as compared to GA, SS, BA and GLS. The proposed GCS algorithm can find the currently known minimum values for the three test cases. The obtained results show the potential of proposed algorithm in solving the runway dependent ALP. In the future, the performance of different components under the proposed GCS algorithm, such as levy flight, selection scheme and crossover operator can be investigated. Moreover, the GCS algorithm can be modified to solve the ALP with time window constraint. This will be the direction in the future research.

ACKNOWLEDGMENT

This study is supported by National Science Foundation of China under Grant No. 61165015. Key Project of Guangxi Science Foundation under Grant No.2012GXNSFDA053028, Key Project of Guangxi High School Science Foundation under Grant No. 20121ZD008.

REFERENCES

Beasley, J.E., M. Krishnamoorthy, Y.M. Sharaiha and D. Abramson, 2000. Scheduling aircraft landings-the static case. Transport. Sci., 34: 180-197.

Cheng, V.H.L., L.S. Crawford and P.K. Menon, 1999. Air traffic control using genetic search techniques. Proceedings of the IEEE International Conference on Control Appliacations, Hawaii, USA.

Hansen, J.V., 2004. Genetic search methods in air traffic control. Comput. Oper. Res., 31: 45-59.

Pinol, H. and J.E. Beasley, 2006. Scatter search and bionomic algorithms for the aircraft landing problem. Eur. J. Oper. Res., 171: 439-462.

Yang, X.S. and S. Deb, 2009. Cuckoo search via Lévy fligh. Proceeding of World Congress on Nature and Biologically Inspired Computing (NaBIC 2009), December 2009, India, IEEE Publications, USA, pp: 210-214.

Yang, X.S. and S. Deb, 2010. Engineering optimization by cuckoo search. Int. J. Math Model. Num. Optim., 4: 330-343.

Yu-Hsin, L., 2011. A genetic local search algorithm with a threshold accepting mechanism for solving the runway dependent aircraft landing problem. Optim. Lett., 5: 229-245.