INCREASING RUNWAY CAPACITY USING GENETIC ALGORITHMS AND ENHANCED HEURISTICS

Ákos Kovács

Department of Aircraft and Ships
Budapest University of Technology and Economics
Stoczek u. 6., Budapest, Hungary, H-1111
Tel.: +36-1-463-1992, Fax: +36-1-463-30-80
E-mail: akos@rht.bme.hu

Keywords: *genetic algorithms, sequencing, neural networks, heuristics*

Abstract

The paper aims to present a method to increase runway capacity by planning optimal sequences with genetic algorithms using enhanced heuristics and parameter estimation with neural networks. A Timed Stochastic Coloured Petri Net model of the runway is used to generate pre-optimized sequences as the initial population of the genetic algorithm. The performance of the algorithm has been investigated by simulation for the case of arrival peak and arrival and departure peak and 20% saving in sequence completion time could be achieved. With use of enhanced heuristics computational time decreased by 10%.

1 Introduction

Nowadays, the air transport organised using the conventional hubing (hub and spoke) philosophy tends to its capacity limits. One of the problems is associated with **runway capacity**. The major international organisations as ICAO, EUROCONTROL, IATA, ACI Europe initiated several programs for increasing runway capacity. Currently running projects are focused on organisational problems and developing of new models for optimising and maximising the throughput of the air transport system.

The **aim** of our paper is to present a method to increase runway capacity by planning optimal sequences with genetic algorithms and thus increasing airport capacity and air transport system efficiency. The planning of optimal sequences at runways corresponds to the pre-tactical and tactical phase of air traffic control, and thus optimisation methods used for this purpose has to work very fast. Genetic algorithms have proven their potential to be used in such applications [3,5,6,7,8]. In order to further improve the efficiency of the applied method enhanced heuristics are used. The flexibility of the method is ensured by online parameter estimation with neural networks.

2 Basic notions

2.1 The selected approach and tools

As the chromosomes of the **genetic algorithm** represent aircraft sequences, the algorithm uses special crossover operator and a modified mutation operator. This is necessary due to the fact, that standard crossover or mutation operators for sequencing and scheduling problems are not suitable, because their mechanism could lead to badly formed strings [3]. The use of genetic algorithms and thus the computation of optimised sequences has the potential to show best results in congested traffic conditions, while in light traffic minor or no modifications to the First Come First Serve (FCFS) principle is needed. Sequences are coded as vectors where the elements represent aircraft. These elements have the form of vectors containing information on runway occupancy time and separations.

In addition, the parameters used in the algorithm are not fixed as in most applications, but can be dynamically modified due to changes in weather (wind, temperature, runway surface conditions etc.) and operational conditions (closing or taxiways, change of runway-in-use) using **neural networks** for parameter estimation. For each parameter and aircraft type a small neural network is created and trained offline (e.g. one network gives the runway occupancy time parameter of a Boeing 737-800 in the full operational range of temperature and wind). The trained nets can then be used for online parameter estimation. The use of such networks is described in [2].

Some extensions of Coloured Petri Nets have already been used in ATM [4,5]. A **Timed Stochastic Coloured Petri Net** (TSCPN) model of the runway is used to generate pre-optimized sequences. These are then used as heuristics, i.e. they are the initial population in the optimization process with genetic algorithms, instead of a randomly generated initial population. The result is either a saving in computational time or better quality solutions. The TSCPN presented in [1] is further developed to contain wake turbulence separations and thus to reflect real operational circumstances. The initial population gained with this heuristics has greater average fitness than the randomly generated one.

The input of the TSCPN model is the natural sequence, and the output is the pre-optimised sequence. This is gained by using the arrivals priority principle in the model. The reason why it is possible to gain a set of pre-optimised sequences is that the model puts some noise on the natural sequence, i.e. changes the planned times over the threshold and planned take-off times with a few seconds randomly every times it is presented to the model. As a result one planned sequence (or natural sequence which is in fact reflection of the FCFS) a number of pre-optimised sequences are obtained. These are then used as the **initial population of the genetic algorithm**. In addition, there are possibilities to put a new aircraft into the queue, and change the status of any of rapid exit taxiways during the execution of the TSCPN model. This is an important feature, because sequence planning is done for a definite time ahead, and as this time window slides forward new aircraft may enter to the sequence to be optimised [8].

The performance of the method applied in this paper is demonstrated under congested traffic patterns, i.e. sequences used as inputs correspond to congested traffic. Computational demonstrations are conducted on a single runway model with two rapid exit taxiways.

The lecture deals with description of the developed method and shows and discusses the results of application of the method described.

2.1 Problem formulation

Given a set of aircraft with estimated landing times (ELDT) and estimated take-off times (ETOT) for arriving and departing aircraft respectively.

An optimized sequence is to be calculated in terms of an objective function if the aircraft use the same runway and a set of constraints are to be satisfied.

The constraints are maximum time and position shifting. These are applied for arriving and departing aircraft separately. Each flight can be shifted by a predefined number of positions in its arrival or departure queue from the FCFS positions. Time shifting constraint is calculated for each individual flight from its ELDT/ETOT, and from the time the flight would use the runway if it were cleared according to the FCFS sequence. An optimized sequence is not acceptable if:

- An aircraft is to use the runway much earlier than its estimated time

Table 1. Separations

		Following aircraft									
[NM]		1	2	3	4	5	6	7	8	9	10
Preceding aircraft: *L*	1	*ROTD*	*ROTD*	*ROTD*	*ROTD*	*ROTD*	*ROTD*	*ROTD*	*ROTD*	*ROTD*	*ROTD*
SID1 prop M	2	*5*	*5*	*6*	*3*	*5*	*6*	*5*	*ROTD*	*ROTD*	*ROTD*
SID1 jet M	3	*5*	*5*	*5*	*3*	*3*	*5*	*3*	*ROTD*	*ROTD*	*ROTD*
SID2 prop M	4	*5*	*3*	*5*	*5*	*6*	*5*	*6*	*ROTD*	*ROTD*	*ROTD*
SID2 jet M	5	*5*	*3*	*3*	*5*	*5*	*3*	*5*	*ROTD*	*ROTD*	*ROTD*
SID1 H	6	*6*	*5*	*5*	*5*	*5*	*5*	*4*	*ROTD*	*ROTD*	*ROTD*
SID2 H	7	*6*	*5*	*5*	*5*	*5*	*4*	*5*	*ROTD*	*ROTD*	*ROTD*
L	8	*ROTA*	*ROTA*	*ROTA*	*ROTA*	*ROTA*	*ROTA*	*ROTA*	*ROTA*	*ROTA*	*ROTA*
M	9	*ROTA*	*ROTA*	*ROTA*	*ROTA*	*ROTA*	*ROTA*	*ROTA*	*5*	*ROTA*	*ROTA*
H	10	*ROTA*	*ROTA*	*ROTA*	*ROTA*	*ROTA*	*ROTA*	*ROTA*	*6*	*5*	*4*

		Following aircraft									
[sec]		1	2	3	4	5	6	7	8	9	10
Preceding aircraft: *L*	1	*90*	*90*	*90*	*90*	*90*	*90*	*90*	*90*	*90*	*90*
SID1 prop M	2	*135*	*135*	*160*	*100*	*135*	*160*	*135*	*75*	*75*	*75*
SID1 jet M	3	*120*	*120*	*120*	*90*	*90*	*120*	*90*	*65*	*65*	*65*
SID2 prop M	4	*135*	*100*	*135*	*135*	*160*	*135*	*160*	*75*	*75*	*75*
SID2 jet M	5	*120*	*90*	*90*	*120*	*120*	*90*	*120*	*65*	*65*	*65*
SID1 H	6	*140*	*120*	*120*	*120*	*120*	*120*	*105*	*65*	*65*	*65*
SID2 H	7	*140*	*120*	*120*	*120*	*120*	*105*	*120*	*65*	*65*	*65*
L	8	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
M	9	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*120*	*0*	*0*
H	10	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*140*	*120*	*95*

- An aircraft is to use the runway much later than its FCFS time (i.e. it suffers much more delay than without sequence planning)

3 Modeling

3.1 The genetic algorithm

As the chromosomes of the genetic algorithm represent aircraft sequences, the algorithm uses a special crossover operator (OCGS) and a modified mutation operator. This is necessary due to the fact, that standard crossover or mutation operators for sequencing and scheduling problems are not suitable, because their mechanism could lead to badly formed strings [3,8]. We do not use OCGT as in [3] because it is generally the same as OCGS. Mutation, which is a random swap of two genes is modified in a way in a way that the closer the genes are to each other the more probable they change positions (i.e. the probability of selecting closer genes for swapping is higher).

Since we aimed at optimising sequences corresponding to congested patterns, the objective function to minimise is sequence completion time. Maximum time shifting and maximum position shifting checks are applied to ensure operationally acceptable solutions (extreme delays and position shifts are filtered out). The fitness is the inverse of the objective function (F_{obj}):

$$F_{itt}=1/F_{obj} \qquad (1)$$

3.2 Sequence coding and separations

Arriving aircraft are considered in three classes according to ICAO wake turbulence categories: light (L), medium (M) and heavy (H). Different types are represented within the categories at arrival runway occupancy time (ROTA) estimation with neural networks. Departing aircraft are coded as *8-10* in Table 1.

For departing aircraft seven classes are defined according to wake turbulence category (L, M, H), standard instrument departures (SIDs for medium and heavy) and speed (for medium): light (*1*), heavy with SID1 and SID2 (*6,7*), medium jet and turboprop SID1 and SID2 (*2-5*). The numbers in brackets correspond to the coding in Table 1.

Separations applied are based on ICAO wake turbulence separations, radar separations and runway occupancy times generated by neural networks. The times are estimated from the given distances in Table 1 using airspeeds of representing aircraft of the class. Since wind has also an effect on the conversion (from distance to time, since the calculation should be based on ground speeds which varies with wind velocity) in the simulation we used different separation matrixes to demonstrate this (the second table in Table 1 corresponds to calm wind). The zeros in the converted table standing for ROTAs generated for each individual arriving aircraft by neural networks. Consider separation matrix (*SEP*). The separation between flight A of class i and flight B of class j, is *SEP(i,j)* or if *SEP(i,j)=0* then the ROTA generated for flight A. (Note that occupancy time is generated for the individual flight, not for the class of flight).

3.3 Using of neural networks for parameter estimation

The used neural networks generate ROTAs as the function of wind and runway status, similarly as in [2] (in [2] the inputs were wind and temperature).

The effect of wind and closing of taxiways is not straightforward. For example in case of strong headwind an aircraft would be able to use a RET (taxiway C in Fig. 2) instead of vacating at the end (taxiway D in Fig. 2) resulting in a smaller ROTA. On the other hand if the aircraft still wont be able to use the RET it results in greater ROTA due slower ground speed.

In case of closing a RET only aircraft that usually vacating via that taxiway will have greater ROTA other occupancy times remain the same (since the other taxiways are still available).

4 Dynamic model

4.1 Concept

The model is based on the assumption that the planning of sequences starts 15 minutes ahead (i.e. the first aircraft in the queue estimates landing or take-off in 15 minutes) in a defined time window (in this case 550 seconds). The reason for this is that 15 minutes is estimated to be close for arriving flights entering the terminal maneuvering area (TMA) and some minutes before start-up for departing aircraft. Of course this time may vary at different airports. Time checks in respect of earliest departure and earliest landing time are based on the possibility to be able to fly or taxi faster to meet the earliest time to the runway. The applied 15 minutes allows controllers to speed up arriving aircraft or instruct aircraft for faster taxi and arrange predeparture sequence.

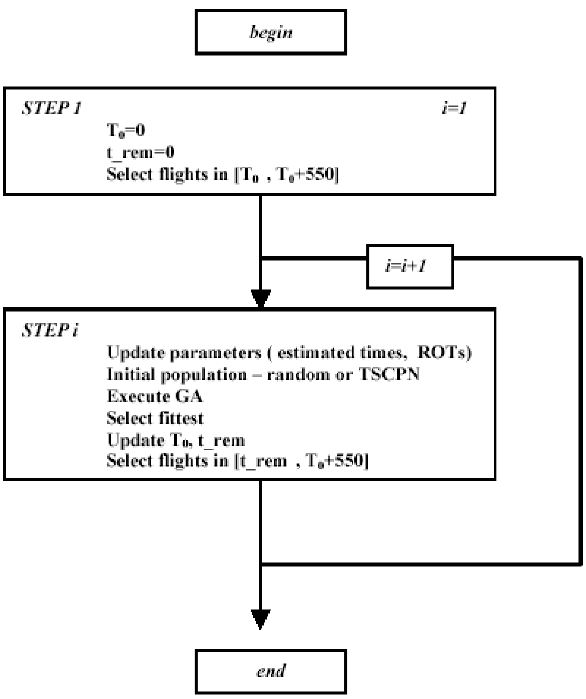
The method is similar to RHC described in [8] or the concept used in Departure Manager [9], i.e. optimization is done in a definite time window sliding forward in time instead of optimizing the sequence till the end of the operating day. The time window moves forward and aircraft are removed from the beginning of the queue and other entering to the end of the queue. This procedure is feasible for computational time saving and due to the fact that uncertainties of estimated times long ahead make the solution too much uncertain.

In this work after an optimization step in a time window the positions of the first two aircraft are decided. The first is removed and the window slides forward by the separation time between the first and the second aircraft. In the next step the flights in the next window are selected and the new sequence is optimized. The first one is removed (from the optimized sequence) and the window slides forward. This process is repeated until there are aircraft in the window.

The overall optimized sequence is given by the flights removed from the first positions in the subsequent steps. Between the steps it is possible to update estimated times (as there may be new information) and parameters due changes in wind, operational circumstances (closing of taxiways, change of runway-in-use).

A block diagram in Fig. 1 shows how the dynamic model works. The first position is fixed in every optimization step because the start of the first aircraft was calculated from the separation from the preceding one. After an optimization step the first aircraft (of the fittest sequence) is removed and the second is fixed, as it is the first in the next step. *t_rem* is defined as the earliest estimated time among aircraft in an optimized sequence from the third position till the end of the queue. *t_rem* is used because it is possible that the first aircraft in the new sequence (starts at updated T_0 as the time window slides forward) is not the one with the earliest estimated time, and if the flights would be selected for the new sequence from interval $[T_0, T_0+550]$ it could be missed out.

Fig. 1. Block Diagram of the Dynamic Model



4.2 Heuristics

The space defined by the random initial population generally used for sequencing problems solved by genetic algorithms is in fact much wider than the solution space of interest. This is because vast majority of the sequences generated randomly are not acceptable (too much time or position shift, or simply not efficient).

If we generate an initial population that is closer to the optimized solution it can be assumed that the algorithm finds the solution earlier or it finds a better solution at the same time.

Using the above-mentioned heuristics has certainly some drawbacks. If the space defined by the heuristics is too narrow (maybe even more narrow than the solution space) the algorithm might not find the optimum or the best quality solution just a sub-optimum (as it possibly happened in some of the cases of the simulation). This problem is also mentioned in [7].

Further investigation is required to explore best balance between narrowing the space defined by the heuristics and running time and solution quality. Basically we can say that the heuristics' space should be wider than the solution space.

The 'working space' of the algorithm is wider than the heuristics defined (consider mutation), but in case of a too narrow initial population it is less probable to find solution outside that space (in a finite number of generations).

For generating heuristics a TSCPN model was used described in [1]. The model in [1] was modified in terms of random time shift applied for the FCFS sequence to fit the genetic algorithm and to contain separations.

Table 2. Results

[sec]	t_fcfs	t_opt	avg_c_t	avg_fcfs_d	avg_opt_d
a_calm_open	2206	1740	6.5254	256.69	47.531
a_calm_opencpn	2206	1740	6.0274	256.69	61.094
a_calm_Cclosed	2557	2306	8.7974	432.97	304.13
a_calm_Cclosedcpn	2557	2222	6.9099	432.97	232.44
a_max_open	2416	1830	6.7538	368.22	81.406
a_max_opencpn	2416	1895	6.4129	368.22	117.5
ad_calm_open	2518	1858	6.8183	395.44	93.313
ad_calm_opencpn	2518	1858	6.2515	395.44	94.219
ad_calm_Cclosed	2718	2158	7.8313	501.22	224.41
ad_calm_Cclosedcpn	2718	2183	6.9005	501.22	232.69
ad_max_open	2800	2039	7.1202	537.63	181.25
ad_max_opencpn	2800	2099	6.6812	537.63	207.78

5 Algorithm implementation

The algorithm is intended to be used online. Under real circumstances the computational time of *one step* has to be low compared to the time the window slides forward (the time between two aircraft).

Neural networks enable online parameter estimation for the algorithm. These however have to be trained offline on measured or calculated data for each parameter needed. This may be a long process but once the trained nets are available they are feasible for online use.

Heuristics generation was done offline because the TSCPN model was not connected to the GA implementation. Of course in an online environment this has to be integrated to the algorithm, but for our purposes (to show heuristics performance) it showed to be acceptable.

The GA model was implemented in Matlab6.5 and was run on a P4 PC.

Our aim was to investigate the effect of enhanced heuristics and online parameter estimation with neural networks. The parameters of the algorithm are tuned to be able to demonstrate our aims but not fine-tuned to optimize running time.

The main parameters are the following:

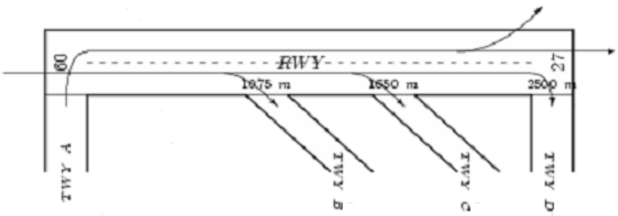
- Population size: 20
- Number of generations (in each step): 80
- Chromosome length: depending on number of flights in time interval
- Mutation probability: 0.01

Parent selection is based on chromosome normalized fitness. A chromosome (S_i) is selected for reproduction with a probability equal to its normalized fitness (*fittnorm*) in the population:

$$fittnorm = F_{obj}(S_i)/\textstyle\sum_i F_{obj}(S_i) \qquad (2)$$

The model runway (Fig. 1.) is 2500 meters long with two RETs (taxiway B and C at 1050 m and 1625 m respectively) and a normal 90° taxiway (D) at the far end.

Fig. 2. The Model Runway



6 Simulation results

For simulation two schedules were created (Sch1, Sch2). Both cases correspond to congested traffic conditions, Sch1 for arrival peak and Sch2 for arrival and departure peak with 32 flights planned in 30 minutes. In Sch1 the share of arriving aircraft is 75% in Sch2 50%.

Both for Sch1 and Sch2 six cases were tested, three with random initial population and three with TSCPN generated one. The three cases were to demonstrate the effect of wind and closing of a RET (namely taxiway C):

- Wind calm, all taxiways available
- Wind calm, taxiway C closed
- Wind maximum (we applied 20 knots), all taxiways available

Typical results can be seen in Table 2. There are six block of rows in the table. The first three are for Sch1, the second three for Sch2. Pairs within the blocks correspond to random and TSCPN generated initial population. The FCFS sequence completion times (*t_fcfs*) and average delays for FCFS (*avg_fcfs_d*) are the same for these pairs since they correspond to the same conditions. The second column is the optimized (overall) sequence completion time (*t_opt*) the third is average computational time for one step (*avg_ct*) and the fifth is the average delay (*avg_opt_d*).

Delays are listed for information only since they are not included in the objective function. However time and position checks have much effect on delays. It is clear that through minimizing sequence completion times delays are also reasonably reduced.

The table shows that in all of the cases with the use of the algorithm an approximately 20% gain in sequence completion time can be achieved. The saving is defined as *1-(t_opt / t_fcfs)*. The slight difference of time saving in favor of arrival-departure peak (*ad_*...) on arrival peak (*a_*...) might be due to the fact that departure aircraft were splitted into more classes in the model.

The effect of operational circumstances (closing of taxiway C (…*_Cclosed*)) and wind (…*_calm* or …*_max*) can be seen comparing the completion times of the same schedules (e.g. in the first three blocks for Sch1). The great differences in these times explain the advantage of using online parameter estimation.

Evaluating heuristics performance it can be said that about 10% of computational time can be saved by TSCPN generated heuristics, though in some of the cases *t_opt* is slightly worse. The difference (where applicable) is rather small compared to the saving in respect to *t_fcfs*. As mentioned is section 4.2 the reason can be that the defined space of the applied heuristics is too narrow. In other cases the same or even better *t_opt* was computed together with saving in computational time, which indicates that more appropriate selection of heuristics should lead to at least same solution quality with decrease in *avg_ct.*

7 Conclusions and future work

In this paper we presented a method to increase runway capacity by planning optimal sequences with genetic algorithms using enhanced heuristics and parameter estimation with neural networks. A Timed Stochastic Coloured Petri Net model of the runway was used to generate pre-optimized sequences as the initial population of the genetic algorithm. The performance of the algorithm has been investigated by simulation for the case of arrival peak and arrival and departure peak and 20% saving in sequence completion time could be achieved. With use of enhanced heuristics computational time decreased by 10%. Neural networks proved to be effective tools for online parameter estimation.

The results show the potential of using enhanced heuristics to save computational time together with good solutions.

Further work will be done on investigation of appropriate generation of heuristics without even minimally degrading solution quality and preserving desired saving in computational time.

References

[1] Kovács Á, Németh E, Hangos K. M. Modeling and Optimisation of Runway Traffic Flow Using Coloured Petri Nets. *Proc 5th IEEE International Conference on Control and Automation*, pp. 881–886, 2005.

[2] Kovács Á. Modellierung des Fluzeugverkehrs einer Start- und Landebahn. *Proc 17th Frühlingsakademie*, pp. 73-76, 2005.

[3] Capri S, Ignaccolo M. Genetic algorithms for solving the aircraft sequencing problem: the introduction of departures into the dynamic model. *Journal of Air Transport Management,* Vol. 10, pp. 345-351, 2004.

[4] Everdij M.H.C, Blom H.A.P, Klompstra M.B. Dynamically Coloured Petri Nets for Air Traffic Management Safety Purposes. *Proc 8th IFAC Symposium on Transportation Systems*, pp. 184–189, 1997.

[5] Shortle J.F, Xie Y, Chen C.H, Donohue G.L. Simulating Collision Probabilities of Landing Airplanes at non-towered Airports. *Transactions of the Society for Computer Simulation*, Vol. 79, No. 10, 2003.

[6] Pinol H, Beasley J.E. Scatter Search and Bionomic Algorithms for the aircraft landing problem. *European Journal of Operational Research* , Vol. 31, pp. 439-462, 2006.

[7] Hansen J.V. Genetic Search methods in air traffic control. *Computers & Operations Research* , Vol. 10, pp. 445-459, 2004.

[8] Hu X-B, Chen W-H. Genetic algorithm based on receding horizon control for arrival sequencing and scheduling. *Engineering Applications of Artificial Intelligence,* Vol. 18, pp. 663-642, 2005.

[9] Magill A. Departure Manager Feasibility Report. *EATMP DMAN project* , 1999.