Aircraft Route Recovery Based on An Improved GRASP Method

Yang He [1,a], Jinfu Zhu [1], Qiang Gao [1], and Bo Zhu[2]

[1] *College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*
[2] *Post-doctoral Station, Shenzhen Airlines, Shenzhen 518128, China*

Abstract. Aircrafts maintenance, temporary airport closures are common factors that disrupt normal flight schedule. The aircraft route recovery aims to recover original schedules by some strategies, including flights swaps, and cancellations, which is a NP-hard problem. This paper proposes an improved heuristic procedure based on Greedy Random Adaptive Search Procedure (GRASP) to solve this problem. The effectiveness and high global optimization capability of the heuristic is illustrated through experiments based on large-scale problems. Compared to the original one, it is shown that the improved procedure can find feasible flight recovered schedules with lower cost in a short time.

1 Introduction

Over the course of normal airline operations, some unavoidable disturbances can be caused by unplanned factors like technical aircraft failures, inclement weather conditions, crew absences, etc. These could lead to the delay and/or cancellation of some flights. As a consequence, an aircraft's routing can become disconnected.

The airline controller in the operations control centre can repair disconnected aircraft routings using a variety of options, such as delaying all subsequent flights, cancelling a set of consecutive flights or swapping the affected flights to other aircraft.

However, the scale of disruption can easily grow to the extent that it is impossible for a human to fulfill this task with a viable outcome in a timely manner.

Instead, an intelligent algorithm is required to identify disruptions and create options to help the controller recover from the disruption and bring operations back to normal. The ideal solution should identify a variety of recovery options and choose the best one based on its revenue and cost to the airline in a reasonable period of computational time.

Teodorovic et al. [1] are pioneers to solve irregular flight recovery problem. With one type of fleet consisting of three aircraft executing eight scheduled flights, the branch and bound method is adopted to recover schedules. Teodorovic et al. [2] improve above problems, in that they put forward a dynamic programming greedy heuristic algorithm. To minimize lost passenger revenue and subject to constraints of curfew and crew availability, Clarke et al. [3] proposes an aircraft recovery model that can be promoted to a multi-fleet, with demonstrating calculation with forty-nine aircrafts and more than two hundred schedules. A column generation scheme is developed by Eggenberg et al. [4] to

[a] Corresponding author : heyang@nuaa.edu.cn

solve it. Tang et al. [5] combines GRASP algorithm with simulated annealing, and reduce the possibility of falling into local optimum. Yan et al. [6] also research flight recovery problem under the circumstances of airport closures. On basis of that, Thengvall et al. [7, 8] solves the problem of multi-fleet and long closures of hubs by proposing a model on time-space network, in which strategies of flight delay, cancellation and swap are brought together. Although applied by ILOG CPLEX, it can only provide one solution, while controllers prefer to choose from serval solution in actual scheduling. Petersen et al. [9] are known as the first scholars that put forth the integrated recovery formulation. Similarly, Sinclair et al. [10] design a large neighbourhood search heuristic algorithm to solve the problem above. Additionally, dynamic framework of the aircraft recovery is established by Vos et al. [11]. Optimized by an efficient aircraft selection algorithm, the framework is established on a linear-programming model that is able to trace aircraft status on parallel time-space networks. And to deal with the stochastic aircraft recovery problem, Mou et al. [12] solve it through building an uncertain programming model based on Hungarian algorithm. Furthermore, Arias et al. [13] combine simulation and optimization techniques to cope with it.

Although linear programming is widely applicable, its poor flexibility leads to the run time being exponential. Considering that the schedule recovery problem is NP-hard, existing experiment results demonstrate that some available methods can solve relatively small problems but difficult to promote to practical flight operation and management. In contrast, heuristics can generate feasible solutions of high quality in limited time and reasonable alternatives can also be offered.

Argüello et al. [14] proposes the greedy randomized adaptive search procedure (GRASP) to solve the flight schedule recovery problem when aircraft are in shortage caused by failure. This method proves its superiority in irregular flight recovery problems that involves sixteen aircraft, forty-two flights and thirteen airports. However, in real world data, there is still much scope for improvement in convergence effect. In this paper, we present an improved GRASP for irregular flight recovery with better convergence and more adaptation. Hence, confronted with two types of disruptions, an improved GRASP for aircraft route recovery is proposed in this paper, and various scenarios is considered in computer experiments to verify the reliability.

2 Problem Description

2.1 Problem statement

The recovery problem considers the scheduled flights that are assigned to aircrafts during a fixed time, generally about three days regarded as recovery period. When disruptions occur, the operational control centre needs to collect information about airport status, maintenance schedules, etc. This paper considers two types of disruptions, airport closure and unplanned maintenance.

Due to any number of unforeseen reasons, an airport may be required to close during a given period of time. During this time, flights are not allowed to departure or arrive at this airport. Apart from airport closures mentioned above, in a real scenario, if an aircraft requires unplanned maintenance due to a faulty component, it is required to remain at the airport for a given period of time. Because the maintenance is unplanned, it may overlap with some downline flights for which it has been assigned. Therefore, the aircraft cannot operate any flights during the unplanned maintenance. The aircraft must be at the specified airport at the beginning of the maintenance period, which means the maintenance can begin immediately upon the aircraft's arrival at the airport, and it can only take off again when the maintenance has been completed.

Apart from the assumptions aforementioned, the operational hard constraints below should be taken into consideration when rerouting aircraft: (i) At any time, at most one task (flight or maintenance) can be assigned to an aircraft; (ii) Each aircraft has a start available time and end available time. The start time of the first task that is assigned to this aircraft must be later than or equal to the aircraft start time. Similarly, the end time of the last task must be earlier than or equal to the aircraft end time; (iii)A fixed 30 minutes idle time (also known as turn time) is required for an aircraft between any two consecutive flight tasks; (iv) Each aircraft has a start available airport from the input of the problem. The aircraft start airport must match its first task's start airport. Any two consecutive tasks (flight or maintenance) that are assigned to an aircraft should be connected. That means the airport where the first task is completed should be the same as the airport where the second task begins; (v) If there is any airport closure, flights are not allowed to departure or arrive at this airport.

Apart from that, there are also some soft constraints: (i) The number of aircraft that are ending at a particular airport should equal to that in the original input. For example, if there are 2 aircrafts at PEK at the recovery end time, it also requires 2 aircraft at PEK at the recovery end time in the solution if terminal airport balance is satisfied. However, this constraint is a soft requirement. If there is the terminal airport balance is violated, a cost penalty will be incurred. (ii) The terminal aircraft positioning for an aircraft requires that the aircraft should be at the same airport as its end available airport in the solution. If the terminal aircraft positioning is violated, there will be a slight cost.

2.2 Recovery policy

During the recovery period, different recovery operation can be applied to recover from the disruption as soon as possible and avoid delay propagation. In practice, four recovery options, including flight delay, flight swap, flight cancellation and maintenance cancellation, are allowed. The four options are described in details below. (i) In a recovery situation, this flight can be delayed for a certain time. During the actual operation, there will be a global delay time threshold of 180 minutes, meaning that flights can be delayed any time between 0 to 180 minutes. The shorter the total delay time is, the better the solution quality will be; (ii) Generally, a scheduled flight is assigned to an aircraft, which is identified by its tail number. During the recovery process, the airline can assign the scheduled flight to any aircraft among a given list of aircraft. If the flight is assigned to an aircraft whose tail number is different to the original assigned aircraft, we call this is a flight swap. The lower the number of flight swaps, the better the solution quality will be; (iii) If in no way the flight is able to be assigned an aircraft, it is cancelled. The lower the number of cancelled flights is, the better the solution quality will be; (iv) If in no way the maintenance is able to be assigned the original planned aircraft, this maintenance is cancelled. The lower the number of cancelled maintenance is, the better the solution quality will be. It is noted that a maintenance cannot be swapped to another aircraft.

In GRASP, in every iteration, it constructs neighborhood alternatives by selecting a pair of aircraft randomly. Besides, it does not involve the consideration of maintenance and airport closures. The complexity in original GRASP algorithm may cause a mountain of operation of choosing and cannot get the optimal solution in specific time.

Compared to original GRASP, in the new algorithm, when the pair is neither disrupted, the routes will be retained. In other words, in every iteration, it constructs neighborhood by selecting a pair of route, one of the route comes from disrupted aircraft routes, the other one comes from the reminder of routes which contains normal, disrupted routes and fictitious flight. The main reason is that all disrupted routes could be improved through swapping a segment of route with the rest ones, with original normal routes remaining the same. Additionally, the new algorithm can deal with situation involving temporary airport closures by delaying, swapping or cancelling relevant flights.

3 Improved GRASP for Flight Recovery

To solve the recovery problem, a heuristic algorithm that integrates the superiority of GRASP and simulated annealing algorithm is proposed. The improved GRASP algorithm answers three key issues: how to construct initial feasible solution, how to construct neighboring solution and how to search globally optimal solution.

3.1 Construction of initial feasible solution

When flight schedule is disturbed by some unplanned events, initial feasible solution can be obtained by simply postponing subsequent flights. Moreover, temporally cancelling the most severely affected flights is also a reasonable method to make the schedule executable. It is important to note that subsequent flights should be removed from the route if one aircraft's available time beyond its first flight departure time, until the aircraft is available. The main process of constructing initial feasible solution is represented as follows.

Step 1. Create initial incumbent solution, and define aircraft route according to flight schedules.

Step 2. Check and deal with the disconnection between start available airport and the departure airport of the first flight, as well as the space disconnection on an aircraft routing.

Step 3. Reassign the departure and arrival time of every aircraft route considering the aircraft start available time, minimum connection time and the relative airports closures

Step 4. Generate feasible solution, which deals with the flights that delayed longer than max delay time and beyond the end available time of aircraft.

Step 5. If the new feasible solution generated in Step 4 avoids constraints on aircraft start available time, connection time and airports closures, then proceed to Step 3, otherwise proceed to Step 6.

Step 6. The construction of initial feasible solution stops and output feasible solution.

3.2 Construction of neighbouring solutions

Route pairs are adopted to construct neighborhood solutions through part of disrupted routes being replaced by other disrupted routes or newly generated route, keeping undisrupted pairs unchanged. There are three main types of methods to generate neighbouring solution: route augmentation, route exchange and route cancellation, which can be divided into seven specific operations. Examples of the methods to generating neighborhood solution are shown in Table 1 and Figure 1.

Table 1. Routing pair sample.

Route	Flight	Departure airport	Arrival airport
1	11	PEK	TAO
1	12	TAO	WUH
1	13	WUH	SHA
1	14	SHA	NKG
1	15	NKG	SHA
2	21	SHA	CKG
2	22	CKG	TAO
2	23	TAO	DLC
2	24	DLC	SHA
2	25	SHA	XMN

(1) route augmentation: It removes one flight or a sequence of several flights from one route source and places these flights in the other target route. Thus, the target route is augmented by the flights removed from the route, which can be implemented by three different operations: (i) pre_change: A circuit that originates and terminates at the same airport in the source route can be placed in front of the target route; For example, flight 14 and 15 in Route1 make a circuit, and put them in beginning of Route2 (as shown in Fig.2 Route1/2 a); (ii) middle_change: A circuit mentioned above in the source route can be placed in the middle of the target route when its starting and ending airport intersects a point in the route; An example of middle_change involves removing the circuit from route1 and relocating them in front of flight 24 in route2 (as shown in Fig.2 Route1/2 b); (iii) tail_change: It removes a sequence of flight, which origins the same airport to target route's termination airport, from the source route and then appends it to the end of the target route, and causes exchange of termination stations. An example of tail_change involves removing the flight 15 from route1 and putting them at the end of Route2 (as shown in Fig.2 Route1/2 c).

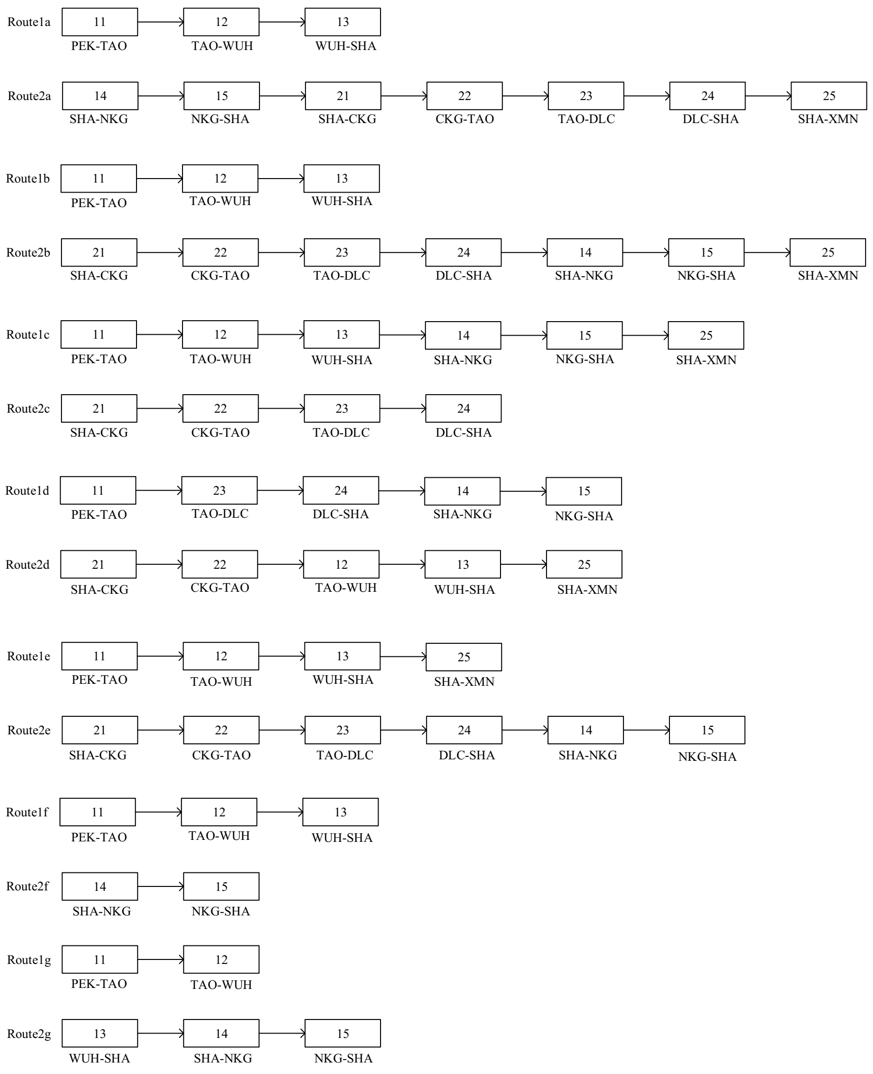

Figure 1. Sample of routing pairs and neighbours.

(2) route exchange: The exchange of a pair of flight sequences is conduct between source route and target route. (i) d_same: The sequence of flights should have the same origination airport, with the termination airports swapped. For example, flight23 and flight24 in Route2d substitute for flight12 and flight13 in Route1d; (ii) d_a_same: The sequence of flights should have the same origination airport, as well as the same endpoint. An example of that is flight14 and flight15 in Route1d substituting for flight25 in Route2d.

(3) route cancellation: It removes a sequence of flights from the source route and then puts it into a newly cancellation route (virtual route). (i) cancel_circuit: It removes a circuit from the route and drops the circuit in a virtual route. As shown in Fig.2 Route1/2 f, it cancels the circuit of flight 14 and flight15, and put them in virtual route (Route2f); (ii) cancel_tail: It removes a sequence of flights from start point to termination airport, as long as the start point differs from the termination airport. For example, flight 13, 14, and 15 do not form a circuit, so they can be cancelled and placed in virtual route (Route2g).

3.3 Searching strategy

After all feasible neighbouring solutions of each route pair are constructed, optimal neighbouring solutions will be chosen and deposited in the restricted candidate list (RCL) and the bad restricted candidate list (BRCL) respectively. The RCL is prepared for cost-reduced neighbours, compare to BRCL which contains cost-added neighbours. However, the original GRASP algorithm only go on along the cost-reduced direction. Therefore, to achieve the global optimal solution, accepting a temporary degraded solution is sustainable in a sense. To some extent, it can get an optimal solution rather than a local optimal solution within a certain range.

In the following, we introduce the procedures of the improved GRASP algorithm, and then the algorithm is explained step by step in flow chart for irregular flight recovery in Figure 2.

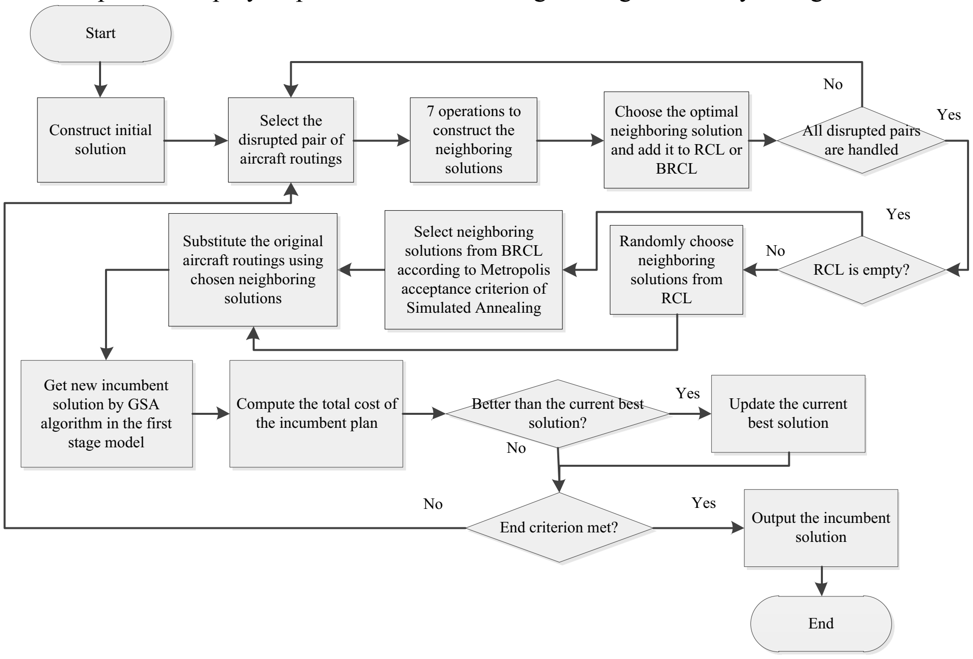
Start
Construct initial solution
Select the disrupted pair of aircraft routings
7 operations to construct the neighboring solutions
Choose the optimal neighboring solution and add it to RCL or BRCL
All disrupted pairs are handled
No
Yes
RCL is empty?
Yes
No
Randomly choose neighboring solutions from RCL
Select neighboring solutions from BRCL according to Metropolis acceptance criterion of Simulated Annealing
Substitute the original aircraft routings using chosen neighboring solutions
Get new incumbent solution by GSA algorithm in the first stage model
Compute the total cost of the incumbent plan
Better than the current best solution?
Yes
No
Update the current best solution
End criterion met?
No
Yes
Output the incumbent solution
End

Figure 2. Process of improved GRASP algorithm in recovery problem.

Step 1. Calculate the changed cost incurred by neighbours and determine whether it is cost –saving or cost-adding, and add the chosen neighbours in RCL or BRCL.

Step 2. If RCL is not empty, choose neighbours as much as possible to improve the current solution. If RCL is empty, randomly choose one neighbour from BRCL according to Metropolis acceptance criterion of simulated annealing algorithm.

Step 3. If the end criterion in simulated annealing algorithm is met, output the incumbent solution, otherwise, proceed to Step 4.

Step 4. Substitute the counterparts in the current solution using the chosen neighbours.

Step 5. Calculate the objective of the current solution and proceed to Step 1.

4 Computer Experiment

In order to evaluate this method, several types of scenarios are generated. All computations were performed on a ThinkPad T460s with Intel i7-6600 CPU and 7.4 GB of RAM. The improved GRASP algorithm was coded in C++ language on Microsoft Visual Studio, with calculating times limited to 30min.

The objective of this problem is to minimize the total weighted cost. The total weighted cost can be calculated by the number of cancelled flights, the number of terminal airport balance violated, the number of terminal aircraft positioning violated, the total flight delay time and the number of flight swaps multiplied by the weights, respectively.

The list of weights is a set of input parameters. An example of this list is given in Table 2 below.

Table 2. Weighted cost sample.

	Cancelled maintenance	Cancelled flights	Terminal airport balance violated	Terminal aircraft positioning violated	Total flight delay time
Cost	1000	800	500	10	1

Table 3 shows the solution with 4 scenarios. The first row is the number of aircrafts; the second row is the number of flights, the third row is the number of airport closures, and the last row is the number of maintenance.

Initial feasible solution rows show the delayed and cancelled flights without interference. This is divided into six rows. The first row is the number of cancelled flights, the second row is the number of cancelled maintenance, the third row is the number of terminal airport balance violated, the forth row is number of terminal airport positioning violated, the fifth row is total flight delay in minutes, and the last row is cost calculated by objective function.

The GRASP row is the solution obtained using the improved GRASP algorithm. This row is divided into eight rows. The first row is the number of cancelled flights, the second row is the number of cancelled maintenance, the third row is the number of terminal airport balance violated, the forth row is number of terminal airport positioning violated, the fifth row is total flight delay in minutes, the sixth row is the number of total flight swaps, the seventh row is the consuming CPU time (minute), and the last row is cost calculated by objective function.

Table 3. Solution comparison.

		Scenario1	Scenario2	Scenario3	Scenario4
#AIRCRAFT		12	44	85	16
#FLIGHTS		95	638	417	59
#AIRPORT CLOSURE		1	0	1	0
# MAINTENANCE		0	29	25	23
Initial feasible solution	cancelled flights	0	0	0	5
	cancelled maintenance	0	5	0	4
	terminal airport balance violated	0	0	0	2
	terminal airport positioning violated	0	0	0	1
	total flight delay in minutes	6600	0	7680	0
	cost	6600	5000	7680	22910
Solution by GRASP	cancelled flights	2	2	4	2
	cancelled maintenance	0	0	0	0
	terminal airport balance violated	0	0	0	0
	terminal airport positioning violated	0	0	2	0
	total flight delay in minutes	275	0	345	0
	total flight swaps	14	10	37	1
	consuming CPU time	5	30	30	0.5

	cost	4364	1100	4817	1010

5 Conclusion

In this paper, we propose an improved GRASP method to solve the aircraft routing recovery problem when flight schedules confronted by disruptions. An initial feasible solution is first obtained by postponing or temporally cancelling subsequent flights. Then, neighbourhood solutions are constructed through parts of disrupted routes being substituted by other disrupted routes or newly generated route. Three neighbouring solutions generation operations are applied to aircraft route pairs and single aircraft routes to enumerate feasible neighbour solutions. The optimal neighbour solutions are deposited in a restricted candidate list (RCL), compared to inferior neighbour deposited in a bad restricted candidate list (BRCL). This procedure is continuous before the end criteria is met.

Our contribution is to demonstrate that through the original GRASP algorithm to solve aircraft route recovery problem, combined with simulated annealing algorithm, there are three main points of improvement, the first is its capacity to reserve the inferior neighbourhood to BRCL instead of stopping the search process as original GRASP when no better solution exists currently. Another is limiting aircraft route pairs exchange in itineration between disrupted routes contributes to improvement of efficiency, which could be applied to solve large-scale programing problem. Besides, we take the consideration of aircraft availability, maintenance and airport closures into this method, the weights of which are alterable so that the method can adapt to the different situation and preferences in airlines.

Acknowledgement

This work is supported by “the Fundamental Research Funds for the Central Universities”, No.NZ2016109.

References

1. D. Teodorovic, S. Guberinic, E.J.O.R **15**, 178 (1984)
2. D. Teodorovic, G. Stojkovic, TRANSPORT. PLAN. TECHN **14**, 273 (1990)
3. D. Clarke, *The airline schedule recovery problem* (Massachusetts Institute of Technology, 1997)
4. N. Eggenberg, M. Salani, andM. Bierlaire, COMPUT OPER RES **37**, 1014 (2010)
5. Wei. X. T., Q. Gao, J.F. Zhu, Forecasting **29**, 66 (2010)
6. Tu. Y. Yan. S., E.J.O.R **103**,155 (1997)
7. J.F Bard. Thengvall B.G., Yu G., TRANSPORT RES A-POL **35**, 289 (2001)
8. J.F Bard. Thengvall B.G., Yu G., IIE TRANSACTIONS **32**, 181 (2000)
9. J. D. Petersen, G. S lveling, J.P. Clarke, E. L. Johnson, S.Shebalov., TRANSPORT SCI **46**, 482 (2012)
10. K. Sinclair, J.-F. Cordeau, and G. Laporte, E.J.O.R **233**, 234 (2014)
11. H-M. M. Vos, B. F. Santos, T. Omondi, TRPRO **10**, 931 (2015)
12. D. Mou, W. Zhao, J APPL MATH **2**, 485 (2013)
13. P. Arias, D. Guimarans, M. M. Mota, G. Boosten, *Proceedings of the 8th EUROSIM Congress on Modelling and Simulation* 265 (2013)
14. M.F. Argüello, J.F. Bard, Yu G, J COMB OPTIM **5**, 211 (1997)