Scientific Research Publishing

Journal of Computer and Communications, 2017, 5, 65-75
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

An Improved Genetic Algorithm for Flight Path Re-Routes with Reduced Passenger Impact

Babatope Samuel Ayo

School of Electrical Engineering and Computer Science, University of Bradford, Bradford, United Kingdom
Email: b.s.ayo@student.bradford.ac.uk

How to cite this paper: Ayo, B.S. (2017) An Improved Genetic Algorithm for Flight Path Re-Routes with Reduced Passenger Impact. *Journal of Computer and Communications*, **5**, 65-75.
https://doi.org/10.4236/jcc.2017.57007

Received: April 22, 2017
Accepted: May 16, 2017
Published: May 19, 2017

Abstract

Adverse weather has serious implications for flight timeliness, as well as passenger and aircraft safety. This often implies that alternative flight paths have to be used by aircraft to avoid adverse weather. To reduce the impact of such path re-routes, exact techniques such as artificial potential field model and Dijkstra's algorithms have been proposed. However, such approaches are often unsuitable for real time scenarios involving large number of waypoints and constraints. This has led to the use of metaheuristic techniques that give sub-optimal solutions in good time. In this work, an improved genetic algorithm-based technique has been proposed. The algorithm used an improved mutation operator, reduced passenger inconvenience and considered the schedules of aircraft.

Keywords

Flight, Weather, Shortest Path, Genetic Algorithm, Passenger Inconvenience

1. Introduction

Weather has considerable influence on air traffic. In fact, weather is one of the major causes of flight delays [1] and has serious implications for passenger and aircraft safety. Flights often have to be re-routed to avoid such adverse weather, which may come up at short notice. However, the weather conflict avoidance process has to be done as efficiently and optimally as possible, in order to minimize delays, distance travelled and other costs. A number of previous researchers have proposed solutions to the problem of obtaining optimal paths that avoid adverse weather. These include exact methods, such as dynamic programming [2], artificial potential field model [3] and Dijkstra's algorithm [2]. However, these methods can take long periods of time to produce solutions, if any. This is unacceptable for the short-term enroute scenario considered in this work. Metaheuristic techniques often provide suitable alternatives. Although the solutions they produce could be sub-optimal, such techniques are generally faster and able to handle larger number of waypoints and constraints. Metaheuristic techniques include simulated annealing (SA) [4], multi-objective genetic algorithm (GA) [5] [6] and ant colony optimization (ACO) [6] [7].

A simulated annealing (SA)-based approach was used in [4] [8] to obtain alternative routes for flights affected by adverse weather. SA [4] [8] is a heuristic inspired by the cooling process used to produce metals with desired properties. An initial flight solution is perturbed by randomly adding, removing, replacing fixes, or increasing/decreasing flight delays [4]. The change in operational costs after perturbation is calculated. If it is negative (meaning lower costs), the new solution x_p is accepted. Otherwise, x_p may still be accepted, depending on a calculated probability [4]. The process is stopped once the system is frozen or a user-defined criterion is reached. However, the work did not consider some important factors, such as the inconvenience experienced by passengers due to taking the alternative routes.

The ant colony system (ACS) algorithm [9] mimics the path-finding behaviour of ants attempting to reach a food source. For each iteration, the ant with the best tour updates the pheromone amount on the edges it visited. Preliminary work on the use of ACO to obtain weather-avoiding paths in a 4D grid has been described in [7]. The objectives of their approach were to minimize heading/altitude changes, minimize deviation from flight plan and avoid weather cells. Snapshots of the aircraft and weather cells positions were taken at regular intervals. More work needs to be done to compare the performance of the ACO algorithm with alternatives for weather avoidance. The work did not also consider effects of path re-routes on passengers.

Ahn and Ramakrishna [10] proposed a genetic algorithm for the shortest path routing problem. Tournament selection without replacement was used. Candidate solutions were encoded using variable length strings. Crossover was carried out by exchanging parts of chromosomes that started from a waypoint common to a pair of parents. They also proposed an improved equation for the sizing of population. The algorithm was applied by [11] to the navigation planning problem for dynamic situations, but used roulette wheel selection. Their aim was to reduce fuel consumption and altitude changes, while increasing flight security. However, weather avoidance and passenger inconvenience were not considered.

In this work, a GA-based method was used because it performs well for large-scale problems [12] [13] and could easily be integrated with existing scheduling algorithms. However, one problem of applying GA to the given scenario was the tendency of the algorithm to get trapped in local optima, leading to very sub-optimal solutions. To solve this problem, an improved mutation operator was used in this work. In addition, many existing solutions do not adequately support the passenger inconvenience factor as one of the costs of re-routing flights. This work improved the work of other researchers by considering this factor and the schedules of other aircrafts.

2. Problem Statement

The problem considered is to find a short-term trajectory that dynamically avoids adverse weather and minimizes costs, subject to constraints. Given the graph structure of the airspace under consideration, an optimal flight path is required from a start point to a destination airport. V represents the waypoints and airports (nodes) in the airspace. A flight leg l_{ab} (in set E of links) exists between two waypoints a and b if there exist at least one route that passes through the two waypoints. Candidate paths are created as sequences of waypoints through which the aircraft passes. Assumptions of the model include:

1) Change of speed between flight legs occurs instantaneously

2) Average delay on each flight leg is known, from historical data

3) Only the enroute flight phase is considered for each aircraft (landing and take-off times considered to be constant)

4) Air traffic sectors and their capacities are known

5) Original schedules of flights are known.

Derivation of the Passenger Inconvenience Objective

Most existing work on re-routing flight paths do not adequately consider the direct impact of adverse weather on passengers. Therefore, a new objective known as the passenger inconvenience factor has been defined to address this. The goal of the defined algorithm is to minimize this objective. The inconvenience experienced by passengers is caused by a number of factors such as number of missed connections, flight delay, intensity of weather phenomenon and the type of adverse weather. Hence, the passenger inconvenience factor (J_{PI}) is given by:

$$J_{PI} = \sum_{p=1}^{N_p} \left[w_\delta J_\delta + w_m J_m + w_{Mc} J_{Mc} \right] \tag{1}$$

where J_{Mc} = cost of number of missed connections, w_{Mc} = weight assigned to missed connection cost, w_m = weight of the weather impact cost, w_δ = weight associated with the delay cost, J_{Mc} = the weather impact cost, J_δ = flight delay costs and N_p = the total number of passengers. These costs are discussed below.

An objective of the defined algorithm is to minimize flight delays and associated costs. The flight delay cost is calculated as:

$$J_\delta = \delta \cdot C_\delta \tag{2}$$

where delay $\delta \geq 0$ and is defined by the difference between actual arrival time and expected time of arrival (ETA), and C_δ = unit cost of delay. The ETA is the sum of the departure time and estimated flight time. To obtain the estimated flight time for an aircraft i, the duration of each flight leg $T_l(a,b)$ is summed up, as shown in the following equation.

$$F_{Ti} = \sum_{a=s}^{d} \sum_{b=s,b\neq a}^{d} \left(c_{ab}^{T} \cdot T_l\left(a,b\right) \right) \tag{3}$$

C_{Tab} is a decision variable for flight duration, that is equal to 1 if the route of the aircraft passes through link (a, b) and 0 otherwise.

The weather impact cost J_m calculates the direct impact of the weather on passengers. It is dependent on the severity of the weather and the cost associated with the weather type, as shown in the equation below.

$$J_m = \sum_{m=1}^{N_m} \left(I_m \cdot C_m \right) \tag{4}$$

where N_m = total number of weather cells, I_m = intensity of the m^{th} weather cell and C_m = the cost associated with the type of weather cell. This cost is determined by how seriously the weather could affect an aircraft and its passengers. For example, severe turbulence has high impact on an aircraft and its passengers [14].

Missed connections happen when the sum of the actual arrival time of the preceding flight leg and the minimum connection time is later than the departure time of the connecting flight. Actual arrival time is obtained by adding flight time to the actual departure time of the flight. The inconvenience to passengers in this scenario is measured by missed connection costs, defined by:

$$J_{Mc} = N_{Mc} \cdot C_{Mc} \tag{5}$$

where J_{Mc} = total connection cost, N_{Mc} = number of missed connections and C_{Mc} = cost per missed connection. Substituting (2), (4) and (5) in (1) gives:

$$J_{PI} = \sum_{p=1}^{N_p} \left[w_\delta \cdot \delta \cdot C_\delta + w_m \cdot \sum_{m=1}^{N_m} \left(I_m \cdot C_m \right) + w_{Mc} \cdot N_{Mc} \cdot C_{Mc} \right] \tag{6}$$

In this work, the unit cost of passenger delay C_δ was estimated as $2 per passenger minute and reflected the cost of time lost by a passenger due to delay [15]. Similarly, the cost per missed connection C_{Mc} was taken as $50 [15]. To derive C_m, it was assumed that in the worst scenario, the weather was so severe that passengers were unable to fly, causing missed connections. In this case, $I_m \cdot C_m$ was taken to be $50. Setting the intensity in this maximal case as 100% (or 1.0), gave C_m as $50. In this work, the intensity of medium level weather was set as 0.5 and that of low level weather as 0.2. In addition, a minimum separation gap of five minutes was assumed for aircraft.

The constraints for the defined model are given below, and determine what solutions are feasible.

1) $D_p(i, j) \le D_{sep}$, where D_{sep} = minimum separation distance and $D_p(i, j)$ is the distance between aircraft i and j located at (x_i, y_i) and (x_j, y_j) respectively at time t, and is defined as:

$$D_p(i, j) = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{7}$$

This constraint implies that the distance among any two aircraft must be more than the minimum separation distance to keep aircraft safety.

2) $n_k(t) \le n_{\max}(t)$. The number of aircraft in a sector k should be less than the maximum allowed at that time.

3) $$\sum_{b=s,b\neq a}^{d} c_{ab}^{T} - \sum_{b=s,b\neq a}^{d} c_{ba}^{T} = 1 \text{ if } a = s; \quad -1 \text{ if } a = d; 0 \text{ otherwise} \tag{8}$$

This constraint specifies link (flight leg) characteristics for each way point [10], with respect to a given route of an aircraft. The first condition ($a = s$) indicates that only one link should leave the start node. Similarly, only one link should end at the destination node ($a = d$). For intermediate nodes, the difference between the number of links entering and leaving a waypoint should be zero.

4) $$\sum_{b=s,b\neq a}^{d} c_{ab}^{T} \leq 1 \quad \text{if } a \neq d\text{; and } 0 \text{ if } a = d \tag{9}$$

This constraint [10] indicates that the sum of links leaving the destination waypoint should be zero for a given route of an aircraft. For other nodes, the sum of links leaving the node should be less than or equal to one.

3. Improved Algorithm for Obtaining Alternative Flight Path

A genetic algorithm-based approach was used to derive a feasible re-route with the least cost. This was given by a multi-objective shortest path that minimized the above-mentioned costs. The shortest path algorithm (GATWR) proposed by Ahn and Ramakrishna [10] was improved and applied to obtaining the optimal flight path. The algorithm was enhanced by improving the mutation process to prevent premature convergence to local optima, a problem that was faced when the initial algorithm was directly applied to the considered problem. The goal of the algorithm was the minimization of the passenger inconvenience factor. The pseudocode of the improved algorithm (GA-IM) for a given aircraft is given below.

Weather Avoidance Algorithm with Improved Mutation (GA-IM)

```
Given: Initial start waypoint $s$, destination airport $d$, adjacency list $La$ for each waypoint $a$.
Locate adverse weather cells and intensity levels.
Get waypoints within range of adverse weather, i.e. distance between centre of weather and
waypoints < $Dsep$.
For each impacted flight leg
Update its cost to reflect intensity of weather
End For.
Get initial population of possible routes.
While (num_gen<max_num_gen)
       Get fitness of candidate paths, considering missed connections.
       Select route (roulette selection).
       Apply crossover.
       Mutate routes.
       Function repair_routes:
         Detect and remove disconnected paths and loops.
         If path crosses that of other aircraft
            Remove affected waypoints.
             Generate alternate path between points before and after crossing.
        End If.
        End Function.
End While.
```

The initial population was an initial list of candidate solutions (a set of possible routes). Starting from the start waypoint *s*, the next waypoint on a route was randomly chosen from its neighbouring waypoints (adjacency list) [10] [16]. This process was applied to each successive waypoint, until the destination was reached. Once a node was selected, it was deleted from the topology so as to prevent repetition of nodes. The random initialization was done to ensure diversity of the gene pool [10].

The fitness function was used to determine the extent to which a candidate chromosome was close to the actual solution. The inverse of the weighted sum of individual costs was used for the fitness function F_i, as shown in Equation (10).

$$F_i = 1/J_{PI} \tag{10}$$

3.1. Selection and Crossover

In our algorithm, a number of parents were selected using the roulette wheel method [12] [13]. In this method, the probability of selecting an individual was proportional to its fitness. Therefore, candidate solutions that are better were given more opportunity to reproduce, while retaining some diversity in the gene pool.

The crossover mechanism used was based on one-point crossover [10]. This method was chosen to reduce the number of invalid paths produced. For any two parents, a waypoint common to both was randomly chosen. Then, there was a swap of waypoints after the common waypoint. Crossover could lead to repeating waypoints, so a repair function was defined.

3.2. Mutation Process

The mutation operation used by [10] involved randomly selecting a mutation point on a candidate route and building a new path from that point to the destination node. To improve search ability of the algorithm, an improved mutation process was employed. The operation involved replacing a randomly-selected waypoint on the considered chromosome (route) with a node selected at random from the set of available waypoints in the given airspace section. In addition, another randomly-selected waypoint from available waypoints in the airspace was inserted at a random position in the route under consideration. The mutation process was modified to improve global exploration and ensure that there was diversity in generated solutions [17]. The rate at which mutation was done was determined by the mutation probability parameter. Whereas a high mutation probability ensures global exploration and diversity, it may lead to slow convergence rate and increase in invalid solutions.

3.3. Repair Function

One problem with the mutation and crossover procedures was that some infeasible chromosomes were generated. This could be due to repeating nodes or disconnected routes. To solve this challenge, a repair function including reconnection and repeated node elimination routines was developed. For the repeated node elimination routine, the algorithm deleted all waypoints between the repeated waypoints [10]. The later repeating waypoint was also deleted. The reconnection routine reconstructed disjoint waypoints that do not have direct connection with each other. It did this by using a subroutine similar to that used to build paths for the initial population. In this case, however, the start and destination waypoints were taken to be the two disjoint waypoints.

The repair function was also used to detect and avoid waypoints that would be concurrently used by other aircrafts at the instants of time that the aircraft under consideration would be using them. The planned paths of other aircraft were used to deduce the possible availability of each waypoint and alternative paths were searched after removing unavailable waypoints. The repair operator for GATWR could only eliminate repeating waypoints.

4. Results and Discussion

The implemented algorithms were applied to a simulated airspace scenario at constant altitude shown in Figure 1. The airspace under consideration was segmented into virtual waypoints in a grid arrangement [18] [19], with a time separation of 10 minutes between adjacent waypoints. The aircraft intended to navigate from waypoint 7 to waypoint 97. There were regions of adverse weather: one was centred on waypoint 36 and the other region affected waypoint 78 and nearby waypoints, as shown in the figure. The first weather region was of medium intensity (lower rectangular shape in Figure 1, outlined in dashes), whereas the second was of high intensity (rectangular shape in top-right corner and outlined in dashes).

The aircraft was to depart waypoint 7 at 08:00 HRS, and arrive at waypoint 7 at 9:45 HRS. The usual path was through waypoints 7-17-27-37-47-57-67-77-87-97, and the connection time for the next flight was 10:45 HRS. Another aircraft was simultaneously in the airspace during the time under consideration and had a path from waypoint 3 to waypoint 93.

The implemented algorithm was run with mutation probability of 0.05 and crossover probability of 0.70. The population size for the algorithm was 100,

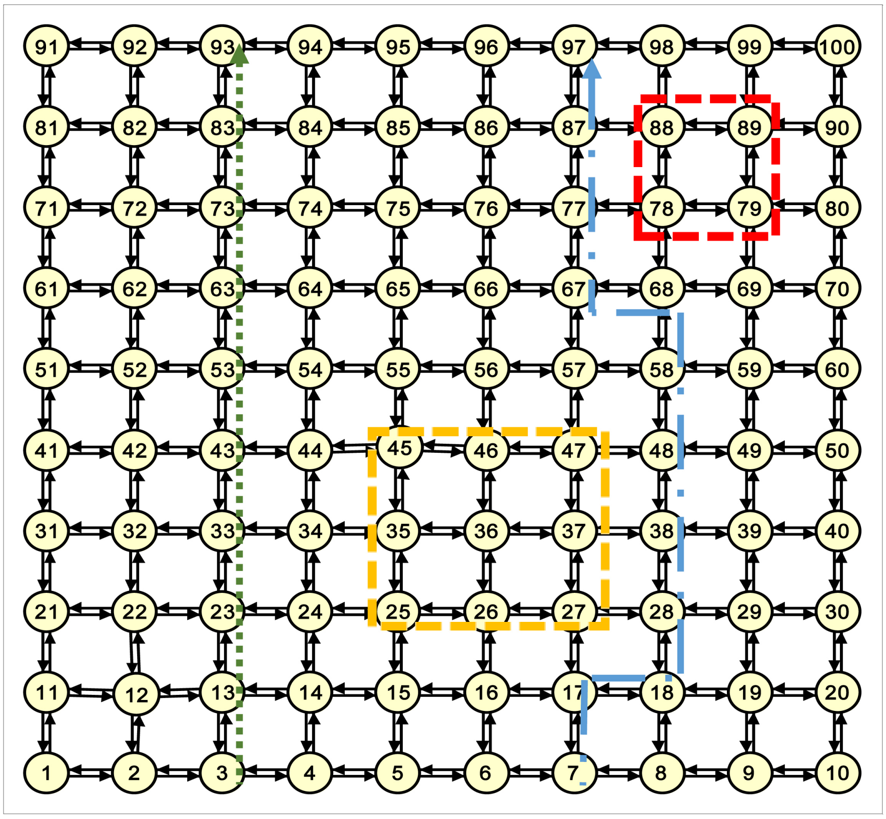

Figure 1. Best path obtained by proposed algorithm.

while the number of generations was 1000. The weights of component costs were kept constant at $w_\delta = 0.\ 2500$, $w_m = 0.5000$ and $w_{Mc} = 0.2500$. For flight legs in regions of high and medium intensity weather, I_m was set to 1.0 and 0.5 respectively. The test scenario was run five times. The algorithms were run using the MATLAB platform (R2015a version) [20].

Figure 2 shows the results of best total cost per passenger obtained at each generation, for the first 500 generations, averaged over five runs of the scenario. The scenario was solved with the algorithm implementing the proposed mutation operator (GA-IM) and with the GATWR algorithm [10]. Roulette wheel selection was used for both algorithms. GATWR converged to a best cost of \$28.50, averaged over the five runs. The incurred cost was made up of delay cost due to using the alternative path. It could also be observed that GATWR converged more quickly to a local optimal and was unable to obtain better cost values thereafter. On the other hand, the proposed algorithm was able to escape local optimal values to obtain a much lower cost value compared with GATWR. This could be attributed to the better search ability of the proposed algorithm using the improved mutation process.

The alternative path obtained by GATWR was 7-8-18-28-29-39-40-50-60-59-58-68-67-77-87-97. A much lower cost of \$2.50 was obtained by the proposed algorithm, also composed of delay cost. The alternative path obtained by the proposed algorithm was 7-17-18-28-38-48-58-68-67-77-87-97, depicted by an arrowed, dashed and dotted blue line in Figure 1. The obtained path avoided the adverse weather regions. For the first 200 generations, Figure 3 shows the mean

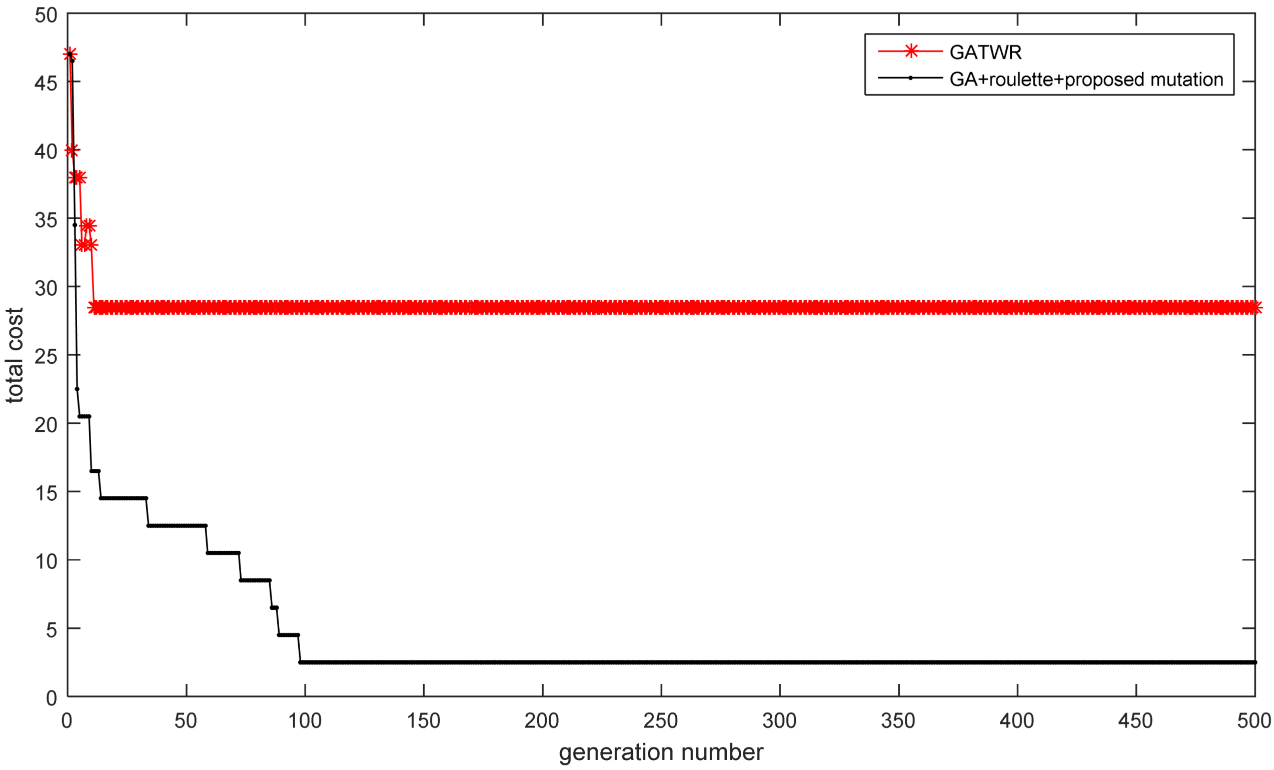

Figure 2. Variation of total costs with generation number.

of the cost values of chromosomes in each generation of the first test run. On the whole, the proposed algorithm produced chromosomes with lower cost values, compared with GATWR. Initially, there were few instances where the proposed algorithm had higher values. This was likely due to the mutation, which initially degraded the existing solutions but improved search capability. This can be confirmed by the lower fitness values eventually obtained by the algorithm. Table 1 gives further details for each run of the scenario, including the best cost value obtained.

As shown in Table 1, the highest cost value (least optimal) of $52.50 was obtained by GATWR. The average error in the cost values obtained by GATWR was $20.00. GATWR only obtained the best path in one of the runs. This gives a low probability of 0.25 for obtaining the best path. On the other hand, the proposed algorithm was able to obtain the better cost value of $2.50 in all the runs. This proves the improved performance of the proposed algorithm.

5. Conclusion

This work has proposed the use of an improved genetic algorithm for obtaining

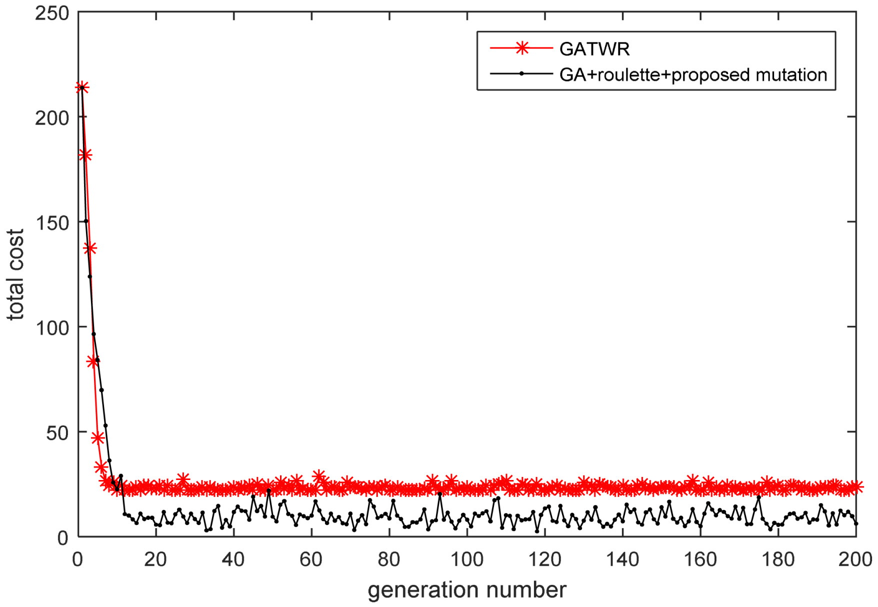

Figure 3. Variation of mean costs of chromosomes over generations.

Table 1. Cost values obtained for each run.

Test Run	Best Cost by GA-IM ($)	Best Cost by GATWR ($)	Error for GATWR ($)
1	2.50	12.50	10.00
2	2.50	52.50	50.00
3	2.50	2.50	0.00
4	2.50	22.50	20.00
5	2.50	22.50	20.00
Average	2.50	22.50	20.00

optimal paths for weather avoidance in aircraft. The algorithm used variable length strings and real encoding for candidate paths. An improved mutation operator has been used, along with roulette wheel selection. In addition, the passenger inconvenience factor was defined. The factor considered the effects of flight delay, weather impact and missed connections on passengers. Results showed that the proposed algorithm was able to produce solutions with lower total costs, while reducing impact on passengers and other aircraft.

Acknowledgements

The author would like to thank the Future Ubiquitous Networks Research Group, University of Bradford, UK for provision of research facilities and support during this research.

References

[1] The European Organisation for the Safety of Air Navigation (2016) Network Operations Report Analysis.

[2] Taylor, C. and Wanke, C. (2009) Dynamic Generation of Operationally Acceptable Reroutes. 9*th AIAA Aviation Technology, Integration, and Operations Conference*, Hilton Head, 21-23 September 2009, 7091. https://doi.org/10.2514/6.2009-7091

[3] Xu, X., Li, C. and Zhao, Y. (2010) Air Traffic Rerouting Planning Based on the Improved Artificial Potential Field Model. *Chinese Control and Decision Conference*, Xuzhou, 26-28 May 2010, 1444-1449.

[4] Taylor, C. and Wanke, C. (2010) Generating Operationally-Acceptable Reroutes Using Simulated Annealing. 10*th AIAA Aviation Technology, Integration, and Operations Conference*, Ft. Worth, 13-15 September 2010.

[5] Cai, K., Tang, Y. and Wang, W. (2015) An Evolutionary Multi-Objective Approach for Network-Wide Conflict-Free Flight Trajectories Planning. 34*th Digital Avionics Systems Conference*, Prague, 13-17 September 2015, 1D2-1-1D2-10.

[6] Alam, S., Bui, L.T., Abbass, H.A. and Barlow, M. (2006) Pareto Meta-Heuristics for Generating Safe Flight Trajectories under Weather Hazards. 6*th International Conference on Simulated Evolution and Learning*, Hefei, 15-18 October 2006, 829-836. https://doi.org/10.1007/11903697_104

[7] Nguyen, M.-H., Alam, S., Tang, J. and Abbass, H. (2007) Ants-Inspired Dynamic Weather Avoidance Trajectories in a Traffic Constrained En Route Airspace. *Proceedings of the* 6*th Eurocontrol Innovative Research Workshop and Exhibition*, Bretigny, 4-6 December 2007.

[8] Chaimatanan, S., Delahaye, D. and Mongeau, M. (2012) A Methodology for Strategic Planning of Aircraft Trajectories Using Simulated Annealing. 1*st International Conference on Interdisciplinary Science for Air traffic Management*, Daytona Beach, June 2012.

[9] Dorigo, M. and Gambardella, L.M. (1997) Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, **1**, 53-66. https://doi.org/10.1109/4235.585892

[10] Ahn, C.W. and Ramakrishna, R.S. (2002) A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations. *IEEE Transactions on Evolutionary Computation*, **6**, 566-579. https://doi.org/10.1109/TEVC.2002.804323

[11] Uçan, F. and Altilar, D.T. (2012) Using Genetic Algorithms for Navigation Planning

in Dynamic Environments. *Applied Computational Intelligence and Soft Computing*, **2012**, Article ID: 560184. https://doi.org/10.1155/2012/560184

[12] Lee, L.H., Lee, C.U. and Tan, Y.P. (2007) A Multi-Objective Genetic Algorithm for Robust Flight Scheduling Using Simulation. *European Journal of Operational Research*, **177**, 1948-1968.

[13] Sivanandam, S.N. and Deepa, S.N. (2008) Introduction to Genetic Algorithms. Springer Science & Business Media, Berlin.

[14] Sermi, F., Cuccoli, F., Mugnai, C. and Facheris, L. (2015) Aircraft Hazard Evaluation for Critical Weather Avoidance. 2015 *IEEE Metrology for Aerospace* (*MetroAeroSpace*), Benevento, 4-5 June 2015, 454-459. https://doi.org/10.1109/MetroAeroSpace.2015.7180700

[15] Arıkan, U., Gürel, S. and Aktürk, M.S. (2016) Integrated Aircraft and Passenger Recovery with Cruise Time Controllability. *Annals of Operations Research*, **236**, 295-317. https://doi.org/10.1007/s10479-013-1424-2

[16] MIT Strategic Engineering (2011) MATLAB Tools for Network Analysis.

[17] Köksoy, O. and Yalcinoz, T. (2008) Robust Design Using Pareto Type Optimization: A Genetic Algorithm with Arithmetic Crossover. *Computers & Industrial Engineering*, **55**, 208-218.

[18] Gleich, D. (2008) MatlabBGL. http://uk.mathworks.com/matlabcentral/fileexchange/10922-matlabbgl

[19] Dunham, M. (2010) graphviz4matlab. University of British Columbia, Vancouver.

[20] The Mathworks Inc. (2015) MATLAB. http://www.mathworks.com/