Network Robustness Improvement Based on Alternative Paths Consideration

Jean-Claude Lebègue[1(✉)], Daniel Delahaye[2], and Jacco Hoekstra[3]

[1] Sopra Steria, Colomiers, France
jean-claude.lebegue@soprasteria.com
[2] Ecole Nationale de L'Aviation Civile, Toulouse, France
daniel@recherche.enac.fr
[3] TU Delft, Delft, The Netherlands
J.M.Hoekstra@tudelft.nl

Abstract. Many transportation networks have complex infrastructures (road, rail, airspace, etc.). The quality of service in air transportation depends on weather conditions. Technical failures of the aircraft, bad weather conditions, strike of the company's staff cause delays and disrupt traffic. How can the robustness of such networks be improved? Improving the robustness of air transportation would reduce the cascading delays between airports and improve the passenger journey. Many studies have been done to find critical links and nodes, but not so many analyze the paths. In this paper, we propose a new method to measure network robustness based on alternative paths. Besides improving the robustness of the French (respectively Turkish Airlines and European) low-cost flight network by 19% (respectively 16% and 6.6%), the method attempts to show the relevance of analyzing the network vulnerability from a path-based approach.

Keywords: Robust network · Topology · Passenger-centric model · Floyd-Warshall algorithm · Simulated annealing · Transport

1 Introduction

The European high-level vision aims for a 4-h door-to-door complete journey for 90% of travelers within Europe by 2050. It also forecasts that the number of commercial flights will reach 25 million. The topology of the airspace network should change to accommodate these new flights. As the number of flights on the network increases, the number of disruptions and delays will also increase if the current network structure remains unchanged. In 2022, the number of flight cancellations in the U.S. has reached 2.69% [1]. This is the highest cancellation rate in a decade, according to the Bureau of Transportation Statistics, not including 2020. These disruptions cause an economic loss every year. By making the airspace more robust, it will be possible to absorb some of the delays and reduce the recovery time. Although robustness comes at a price, taking it into account will improve the quality of service provided by airlines.

A. L. Martins et al. (Eds.): INTSYS 2023, LNICST 540, pp. 179–193, 2024.
https://doi.org/10.1007/978-3-031-49379-9_10

Every year, the world's population grows, and with it the size of our transportation networks (road, rail, airspace, etc.) to meet the increasing demand. At the beginning of air transportation, the Benoist Airboat Model XIV, no. 43 [2] could accommodate two people: the pilot and a passenger. Now, the Airbus A380 has a capacity of more than 500 travelers [3]. The transportation network is not only growing according to a horizontal axis but also to a vertical one. Horizontal growth is related to the increase in the size of the network associated with a transportation mode (number of vehicles and their capacities) whereas vertical growth refers to the number of transportation means. For instance, based on the European forecast air transportation is mainly concerned with horizontal growth. The bigger the network, the more complex it is.

This complexity brings a lot of new challenges to deal with. The quality of service should remain at least the same, no matter the size of the network. People expect to spend the same amount of time doing what they used to do before the network gets bigger. One of the most important challenges is the load, especially during peak hours. The network is supposed to offer an acceptable service even when capacity is reached. In a society where competition between services is a norm, it can be a reason to shift from one service to another. Finally, the network must be robust to disruptions. At a time like ours, time is a precious resource. Therefore, it is inconceivable to spend a lot of time in a traffic jam because of a disruption nor to wait for hours for the airline operators to set up a solution to make the passengers reach their final destinations.

A branch of the research in transportation theory has been devoted to the study of robustness. Researchers in this community have developed several models for analyzing networks. These models try to quantify the extent to which infrastructure elements are vulnerable to failures, congestion, attacks, etc. The methods to address this problem are numerous, but can be divided into two groups: topological vulnerability analysis and system-based vulnerability analysis. The metrics based on the former come from complex network theory and usually use graph properties without considering the dynamics of the transportation networks. The models from the latter method overcome this aspect by integrating notions from transportation theory. Vulnerability is usually quantified by the difference in cost between a nominal state and a disrupted state.

Robustness is not limited to the identification of critical elements, but also defines methods to improve the robustness of the network. The strategies to improve the robustness are numerous. The rewiring strategy gets a lot of attention lately thanks to its network properties conservation aspect, in particular, the node degree conservation. Sometimes the choice of a rewiring strategy seems to be justified by economic benefits however this kind of argument is not as evident as one may think. Firstly, it depends on the characterization of the network. On transportation networks, rewiring between non-directly connected elements involves adding a new connection. Secondly, establishing new connections at the strategic level may be a possible and interesting option. By strategic level, we mean that the flight schedule for a day is known several months in advance so the airspace topology of this day can be analyzed and improved.

The following paper presents a new topological method to improve network robustness based on alternative paths. The strategy chosen here is the addition of new links in a static state of the air network. It is well known that robustness is correlated with the number of links in a network and hence with the number of paths. However, enumerating path is a time-consuming task. Instead of using classical k shortest path algorithms, the authors propose a modified Floyd-Warshall algorithm to quantify this property and use a simulated annealing algorithm to find the set of links whose addition improves the network robustness. Knowing the future flight schedule, we search for flights to add to an air network in order to increase the number of alternative routes and make it more robust to potential disruptions or major delays. By doing so, we expect the operators to use fewer resources to deal with disruptions when they occur.

The next section presents state-of-the-art methods for improving robustness. Section 3 details the model and an algorithm for adding links that maximizes robustness. Section 4 describes the methodology. Section 5 presents the results obtained with such a method and discusses the advantages and shortcomings of our approach and the last section highlights future work.

2 Related Work

Robustness analysis is a large domain that gathers research topics such as analyzing the impact of random and targeted attacks on networks, identifying important nodes and links, defining robustness measures [4]. All these topics lead to different definitions of robustness and resilience in air traffic management (ATM). [5] define the robustness as the ability of a system to experience no stress during a time horizon. Stress refers to the deviation of a system from its reference state. For [6], robustness is the capacity of a system maintain its connectivity following random node isolation. According to [7], a system is robust if it can maintain its performance when facing a disruption.

In transportation networks and specifically in ATM, robustness is related to network connectivity and passenger delays.

In [8], it is the difference between schedule and execution efficiency over an observation period. For [9], it is difference in the global traffic travel time before and after a link disruption. [10] use graph signal processing methods to identify and quantify the abnormal distribution of delays across US airports. As [11] remark, ATM politics want to improve passenger mobility but lack passengers-oriented metrics. The paper proposes a set of passengers-centric measures to complement the already flight-centric measures to better evaluate air transportation network performance.

A lot of robustness measures in the literature model focus on network connectivity. Some models characterize robustness by the size of the largest connected component after the removal of the graph's elements (nodes or links) [12–14]. There are other robustness models based on complex network theory [15], and among them are the centrality indices [16,17]. Degree, betweenness, closeness, and eigenvector centrality are the most famous. Another attempt uses the network spectrum especially the algebraic connectivity [18,19]. This measure is a well-known qualitative index to compare OD pair's connectivity but lacks precision. The previous enumerated approaches are purely topological and therefore do not capture transportation features.

Some robustness models are based on the shortest path between OD pairs. The network efficiency [7] is a measure of the proximity of nodes in the network. Betweenness centrality [16] measures the number of shortest paths through a node. This metric can identify important nodes in a network. Both previous models consider only the shortest path. However, passengers do not necessarily choose this path for their trips. In [20], robustness is defined as the number of rerouted passengers in a disrupted network. The shortest path may not be sufficient to reroute passengers, and this alternative path may not have an acceptable travel time, which can lead to trip cancellation.

Improving network robustness is a complex task because of the size of the state of space. Different approaches were developed to address the problem, they are based on a random approach. Several papers have shown that a topological change of a network structure can significantly improve its robustness [12,14,21]. The most acknowledged strategies consist of adding and/or removing elements from a network [22–24] or rewiring existing connections to form new connections [12,25].

Floyd-Warshall's algorithm has received a lot of attention from the research community. Several studies were done to improve the computation time by different means such as GPUs and parallelism [26], cache optimization [27], matrix multiplication optimization [28]. The algorithm was also generalized to compute the k shortest paths [29]. However, it seems that nothing has been done on attempting to assess the number of paths connecting all node pairs.

The main contribution of this paper is a new passenger-centric robustness model based on alternative paths. The model differs from the literature in that it does not consider only the shortest path. From the passenger's perspective, there are several options where they can be rerouted if something unexpected happens on their original trip. These potential alternatives have a travel time that does not deviate too much from the shortest path travel time. Even if the passenger is rerouted, his total travel time (time spent in the aircraft) remains acceptable.

3 Model

Path-based models are extremely rare. Those that take this approach only focus on one path: the least-cost path. From the passenger's perspective, the cost is significant, but so are the alternatives in case of disruptions. If they are few, the re-routing solution can be stressful and tiring.

In this paper, we consider a static transportation network (Table 1) so there is no waiting time between connections. By connection, we mean a generalization of the flight leg. We focus only on passenger travel times. We are looking for a set of connections to add to this network to improve the number of alternatives, and thus, the passenger travel time in case of a disruption. We assume that the connections are not full and that there are always seats available to reassign passengers to all alternatives.

Table 1. Main variables used to describe the robustness model

Variables	Description
$\mathcal{T}$	Transportation network
$\mathcal{G}$	Strongly connected digraph
$\mathcal{N}$, $\mathcal{A}$	Set of nodes and links
$\mathcal{P}$, $\mathcal{Q}$	Set of origins, destinations
$\mathcal{L}$	Set of non existing links
$\mathcal{G}^{\mathcal{L}}$	$\mathcal{G}$ where $\mathcal{L}$ was added to $\mathcal{A}$
$\mathcal{R}_{pq}^{\mathcal{L}}$	Routes connecting node p to node q in $\mathcal{G}^{\mathcal{L}}$
$\mathbf{d} = (d_{pq})$	Fixed demand
$\mathbf{t} = (t_a)$	Link cost
$\mathbf{c} = (c_{pqr})$	Route cost

3.1 Problem Formulation

Let us consider a transportation network $\mathcal{T} = (\mathcal{G}, \mathbf{t}, \mathbf{d})$ [30,31] where:

- $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ is a strongly connected digraph with $\mathcal{N}$ being the set of nodes and $\mathcal{A}$ the set of links.
- $\mathbf{t}$ is the vector of the costs of the different links of the transportation network.
- $\mathbf{d}$ represents the demand between two nodes of the network. It is the number of passengers that want to travel from an origin node p to a destination node q. Moreover, $\mathcal{P} \subseteq \mathcal{N}$ and $\mathcal{Q} \subseteq \mathcal{N}$ are respectively the sets of origin nodes and destination nodes.

In this paper, the transportation network robustness is related to the richness of alternatives. The more paths connect OD pairs, the more there are alternatives for rerouting the passengers when a disruption happens. The diversity of paths helps to maintain network connection, which is one of the most fundamental criteria when it comes to analyzing robustness because it ensures the existence of a path between all pairs of nodes.

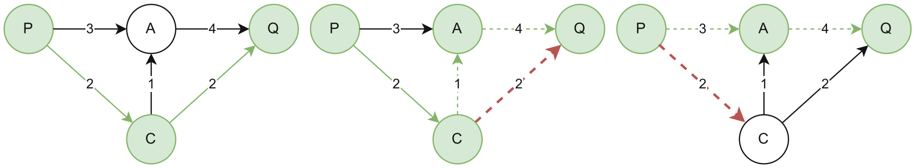


Fig. 1. Yen algorithm 3-shortest-paths (green paths) computation process between OD pair PQ based on the disconnection (red-dashed links) of each of the links forming the shortest path (green-lined path) (Color figure online)

We are interested in a robust transportation network model based on alternative paths between an OD pair. Let us consider a path $r = (p, c, a, q)$ connecting an OD pair pq and v_{pqc}, v_{pqa} being the respective robustness of the nodes c and a. These robustness values are characterized by their number of alternative paths between an internal node $i \in (c, a)$ and the destination node q. We are looking for a path model that combines these v_{pqi} values. The goal behind this definition is to capture the alternatives at each node i along the path r. A robust path always possesses an alternative to the destination in case of a disruption on the nominal path r. The model is applicable to large transportation networks, but it requires efficient computation of alternative paths for all paths connecting all OD pairs, and this operation is very time-consuming and requires a lot of memory resources.

To fully analyze the robustness of a transportation network based on the previous model, one needs a loaded network and the paths on which passengers are assigned. However, these data are not easily accessible, so instead of the model described above, we focus on a less restrictive model. In general, the shortest paths connecting two nodes are quite similar. They slightly differ from the shortest path by very few different nodes. It is based on this principle that the Yen algorithm [32] computes the k shortest paths between two nodes. As illustrated by Fig. 1, the shortest path (green-line path) between P and Q is (P, C, Q) with a cost of 4. To compute the two other paths (green-dashed paths), the algorithm is going to iteratively remove (red-dashed links) the links (C, Q) and (P, C) from the network to respectively find the paths (P, C, A, Q) and (P, A, Q). Both paths have a cost of 7. The main drawback of this algorithm is the network modification at each iteration (disconnecting and reconnecting links). Unlike the Yen algorithm, the Floyd-Warshall algorithm does not perform this operation on the network at each iteration because the paths are built progressively. Moreover, the latter algorithm is known to be efficient in computing all shortest paths between all pairs.

To make the running time acceptable, we limit the method to the computation of the paths connecting an OD pair. The main benefit of this approach is to combine the computation of the distinct paths connecting the OD pairs and the computation of the alternatives along them. The robustness of an OD pair is quantified by the paths connecting its origin to its destination.

The model proposed by the authors can be defined by the following optimization problem:

$$\begin{cases} \max\limits_{\mathcal{L} \in \mathcal{P}(\mathcal{N}^2 \setminus \mathcal{A})} f(\mathcal{L}) = \sum\limits_{p \in \mathcal{P}} \sum\limits_{q \in \mathcal{Q}} |\mathcal{R}_{pq}^{\mathcal{L}}| & (1) \\ c_{pqr} = \sum\limits_{a \in r} t_a \quad \forall r \in \mathcal{R}_{pq}^{\mathcal{L}} & (2) \\ \mu < t_a < \eta \quad \forall a \in \mathcal{L} & (3) \\ |\mathcal{L}| < \kappa & (4) \\ c_{pqr} < \pi \quad \forall r \in \mathcal{R}_{pq}^{\mathcal{L}} & (5) \end{cases}$$

The strategy adopted here is to add elements. The model searches for the set of links $\mathcal{L}$ that will increase the number of paths from all origin nodes to all destination nodes. $\mathcal{R}_{pq}^{\mathcal{L}}$ is the set of paths connecting nodes p and q in the graph $\mathcal{G}^{\mathcal{L}}$ where $\mathcal{L}$ has been added to the link set $\mathcal{A}$. Constraint 2 is the definition of the cost of a path r based on the cost of its links a. Constraint 3 imposes lower and upper bounds on the travel time of the newly added links. We don't want to connect airports that are too close together or too far apart. Constraint 4 fixes the maximum number of links to add to the network and finally constraint 5 fixes the maximum travel time spent by the passengers on the path r connecting p to q.

3.2 Modified Floyd-Warshall Algorithm

The heuristic chosen to solve the problem is the simulated annealing [33]. This metaheuristic is a well-known algorithm in the operations research community to deal with optimization problems. This approach can reach the quasi-global optimum solution in a reasonable computation time.

The main principle of the simulated annealing algorithm (Fig. 2) consists in generating a neighbor $\mathcal{M}$ and comparing its objective evaluation $f_{\mathcal{M}}$ with the current decision evaluation f. If $f_{\mathcal{M}}$ is better, f and $\mathcal{L}$ are updated. However, these variables have a non-null probability to be updated even if $f_{\mathcal{M}}$ is worst so that the algorithm does not stay in a local maximum. In our implementation, we used the well-known metropolis rule [33] to accept from time to time a bad solution.

The core of the simulated annealing algorithm is the neighboring operator because it is the process that is going to modify the taken decision (Fig. 3). The neighboring operator implemented is threefold: twenty-five percent of the time, the size of the decision is increased as illustrated in Fig. 3. The decision goes from one link to two. Another twenty-five percent is used to reduce the size of

Data: adjacency matrix of a strongly directed digraph
Result: set $\mathcal{L}$ of links that best improve the robustness of the network
$\mathcal{L} \leftarrow$ random set of non-existing links
$f_{\mathcal{L}} \leftarrow$ objective function evaluation for $\mathcal{L}$
$T \leftarrow$ initial temperature
repeat
    for $k <$ *number of iterations* **do**
        generate a neighbor $\mathcal{M}$ of $\mathcal{L}$
        evaluate the objective function $f_{\mathcal{M}}$ for $\mathcal{M}$
        $r \leftarrow$ random number in [0,1]
        if $f_{\mathcal{M}} > f_{\mathcal{L}}$ *or* $r < e^{\frac{f_{\mathcal{M}} - f_{\mathcal{L}}}{T}}$ **then**
            $f_{\mathcal{L}} \leftarrow f_{\mathcal{M}}$
            $\mathcal{L} \leftarrow \mathcal{M}$
        end
    end
    decrease the temperature T
until *temperature is sufficiently low*;

Fig. 2. Simulated annealing pseudo-code to solve the optimization problem (Eq. 1).

the decision by one element. The rest of the time is spent swapping the elements in the decision with the same number of other random elements from the absent links set.

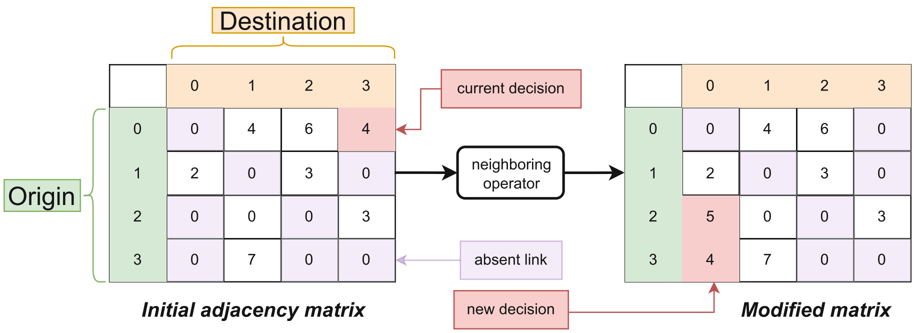


Fig. 3. Simulated annealing neighboring operator functioning. The operator modifies the current decision to a new decision according to three operations: adding, removing, and swapping.

Now that the method to solve the optimization problem has been presented, the last thing to do is to find a way to evaluate the objective function (Eq. 1). It is a tough task to do it fast, due to the number of possibilities. Here, the authors propose a modified Floyd-Warshall [34] algorithm (Fig. 4) to tackle the problem. The approach proposed here does not focus on computing the shortest paths between all pairs of nodes instead it estimates the number of alternative paths.

Data: strongly directed digraph $\mathcal{G}(\mathcal{N}, \mathcal{A})$
Result: alternative paths matrix P
$P \leftarrow$ paths set matrix
for $k \in \mathcal{N}$ **do**
    for $i \in \mathcal{N} \setminus \{k\}$ **do**
        for $j \in \mathcal{N} \setminus \{k, i\}$ **do**
            $P_{ij} \leftarrow P[i][k] \oplus P[k][j]$
            filter the paths in P_{ij}
            $P[i][j] \leftarrow P[i][j] \cup P_{ij}$
        end
    end
end

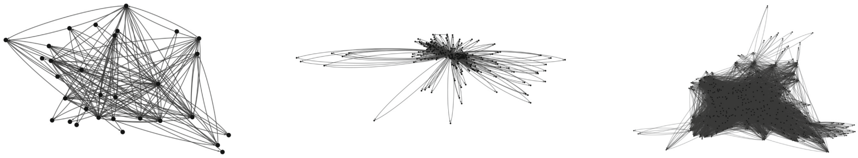
Fig. 4. Modified Floyd-Warshall algorithm to assess the number of alternative paths between all pairs of nodes without looking for the shortest paths.

Like the classical Floyd-Warshall algorithm, the version proposed in this paper builds the paths connecting OD pairs (i, j) by passing through a transit node k. The paths P_{ij} are the result of the concatenation of the paths $P[i][k]$ (connecting OD pair (i, k)) and the paths $P[k][j]$ (connecting OD pair (k, j)). Then, the set P_{ij} is filtered to remove cycles and expensive paths. Finally, the set P_{ij} of paths going through k is added to $P[i][j]$ which contains the other paths connecting (i, j).

4 Numerical Examples

The robustness improvement method presented in the previous section was tested on the French (Fig. 5a), Turkish Airlines (THY, Fig. 5b), and European (Fig. 5c) low-cost flight networks on 01/06/2018. The network data come from the Eurocontrol database. The main features and topologies of the studied networks are summarized in Fig. 5. Each node is an airport. Two nodes are connected by a link if there is at least one flight between the two airports in the data. The travel time of the leg is the difference between the departure airport's off-block time and the arrival airport's in-block time.

No information about the actual demand was available, so both sets $\mathcal{P}$ and $\mathcal{Q}$ were considered to be equal to $\mathcal{N}$ so it contains all possible pairs of the network because it is the worst-case scenario. This means that there is always at least



(a) French network: 31 nodes and 144 links

(b) THY network: 216 nodes and 610 links

(c) European network: 330 nodes and 5194 links

Fig. 5. The topology of the three strongly connected low-cost flights networks of June 1st, 2018: France, THY, Europe

one passenger who wants to go from any airport to any other airport in the air transportation network. We assume that the newly added flight was performed by an A320. We define a linear regression model to compute the new leg travel time based on the distance between the two airports of the flight. The model was fitted to the low-cost flights performed by an A320 (Fig. 6). The accuracy score of the model is 0.98. In the simulated annealing run, if the travel time between two airports was less than 20 min (roughly equivalent to a two-hour drive) or greater than 7 h and 30 min (the maximum travel time performed in the data), the travel time was set to infinity.

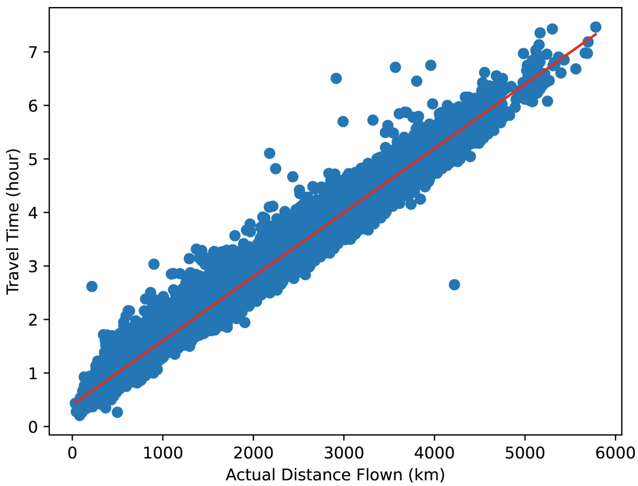
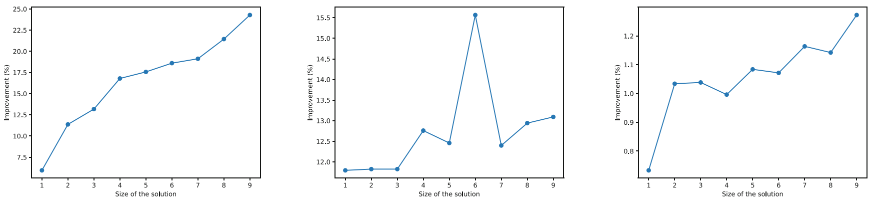


Fig. 6. A320 flight travel time model based on the distance between the two airports of the flight by doing a linear regression of the low-cost flight data done the 01/06/2018

Furthermore, the number of iterations per temperature level was set to 100 and the decreasing cooling coefficient was set to 0.97 to allow a reasonable state of space exploration. This means that from one temperature level to its lower level, the temperature is decreased by 3%. This process is repeated until the temperature reaches 0.01% of its initial value. Finally, the maximum travel time on the French (resp. THY and European) network was set to 7200 s (resp. 10800 s and 12600 s), which is about two hours, and the number of legs to 4. These upper bound were fixed based on the data. In fact, 75% of the French (resp. THY and European) flights have a travel time less than 4916 s (resp. 7380 s and 8322 s). The upper bound was determined by increasing the travel time values by 50%.

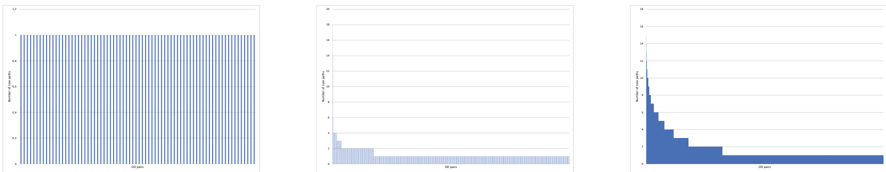
The raw data set contained internal and external flight data. In this study, we focus only on internal flights. All the external flights were removed from the data set.

5 Results



(a) French network (b) THY network (c) European network

Fig. 7. Improvement (percentage) of the number of alternative paths between all the node pairs for different sizes of solution (ranging from one to nine) on the French, THY, and European low-cost flight networks



(a) French network (b) THY network (c) European network

Fig. 8. Distribution of the paths (sorted in decreasing order) per OD pair after adding the solution links (5% of the network links) to the low-cost flight networks

5.1 Improving the Robustness of Air Transportation Networks

Firstly, we analyze the influence of the solution size on the robustness. For the three networks, we compute the best solution for sizes ranging from one link to nine links (Fig. 7).

On the french network (Fig. 7a), the improvement is linear with the size of the solution and goes from 7.5% to 25%. This network is small (31 nodes) so adding one link represents a 0.7% improvement in the number of links whereas adding nine links represents a 6% change compared to the initial topology. The results show this structure modification has a great impact on the number of alternative paths. Unlike the two other transportation networks, the state of space is smaller therefore it is less complex to find the best solution. As we can see and expect, the larger the solution, the greater the improvement.

On the THY network, we notice a difference between the first three solutions whose improvement is less than 12%, then it increases to 15.5%. It decreases to 12.5% before increasing again to 13% (Fig. 7b). This time, the network is larger, and the addition of nine links only represents a 1% increase in the total number of links. However, this small change in structure is enough to improve the robustness of the network. The decreasing behavior of the curve at size 7 is unexpected, since the program is supposed to find at least the same set of links as a smaller-size solution. This behavior is due to the state of space exploration. In fact, it is so large that the program has not tested the links found for the size 6 solution.

Finally, we analyze the influence of the solution size on the European network. The results vary very little around 1% improvement (Fig. 7c). However, we can observe a slight increase from size one to nine.

5.2 Analysis of Flight Paths Distribution in Air Transportation Networks

The results showed that the methodology we introduce in this article can globally improve the number of alternative paths between the OD pairs in an air transportation network. We were curious about the shape of the distribution of the newly added path for each OD pair. Figure 8 plots the number of added paths per OD pair only for the OD pair that have been improved. We wanted to compare the three networks when the same proportion of links have been respectively added and analyze their new paths distribution according to their OD pairs.

In the French network, we added 7 links which are (LFML, LFMT), (LFLL, LFMP), (LFBO, LFBT), (LFBH, LFBP), (LFRB, LFBI), (LFSB, LFLL), (LFST, LFSL). These links increasing the number of alternative path by 19%. The improvement concerns 74 OD pairs over 930 (Fig. 8a). The paths are uniformly distributed among the improved OD pairs: they all received one new path.

In the THY network, we added 31 links. In Fig. 8b, we can distinguish three groups: OD pairs that receive one new path, two new paths and more than three.

This latter group is a minority. Together, 837 over 46440 OD pairs have been improved. It represents 16% new paths in the network. The OD pair (LTAC, LTFJ) obtains the maximum number of alternatives with 18 new paths. This link connects Esenboğa airport with Sabiha-Gökçen airport.

Finally, the distribution of European paths (Fig. 8c) looks similar to the THY distribution. The number of paths is increased by 6.6%. 7833 over 108570 OD pairs are improved when the solution links are integrated into the European network: 68% get one new path, 14% get two paths, and the remaining 18% get more than three paths. OD pair (EGCC, EGCN) receives the maximum number of new paths for this transportation network with 16. This link connects Manchester airport with Doncaster-Sheffield airport.

Flights on the French and European networks are operated by several different airlines. Adding the set of solutions that can increase robustness is complex for at least two reasons: firstly, how the airlines would be chosen to perform these new flights and secondly, these new routes would entail a loss of customers for other airlines. However, the process is much simpler for THY because the flights on its network are only operated by its own aircraft. Improving their network would only benefit them, with no loss of customers.

6 Conclusion

The paper has presented a new model of transportation network robustness based on alternative paths between OD pairs. We also developed a method to improve it. The method is the combination of a simulated annealing metaheuristic and a modified Floyd-Warshall algorithm. The latter is used to assess the number of paths connecting the OD pairs. The method has been tested on three static study cases: the French, THY, and European low-cost flight networks. The number of alternatives in the French (resp. THY and European) network has been improved by 19% (resp. 16% and 6.6%), and these alternatives concern 8% (resp. 2% and 3%) of the OD pairs. The difference in results between these studies is due to the size and the topology of the networks. By slightly changing the network topology (5% new links have been added in each network), the studies have shown it is possible to greatly improve the number of alternatives among the passenger OD pairs. In this study, the passenger can move from any airport to any other airport. Having several alternatives is interesting during a disruption because it reduces the costs generated and the resources to mobilize to reassign the passengers. The most interesting benefit is that if a disruption happens on an improved OD pair, there is at least one solution for the passengers to reach their destination with little delays. For a company, robustness is a guarantee of the quality of service and insurance to reroute the passengers if a disturbance happens on its network.

Although we consider the OD pairs, we did not simulate disruptions nor manage the potential stranded passengers. Our interest here was to globally improve the robustness of a transportation network from a passenger perspective. The next step to validate this work is to ensure that there are always enough seats on the alternative paths for all the passengers and to reassign the stranded passengers to them.

The current version of the model only maximizes the global number of alternative paths in the network. However, it is not relevant to find more alternatives than the number needed to reassign all passengers. By considering this constraint, we can maximize the number of improved OD pairs.

The model is also static. We did not take into account the waiting time and focused only on the travel time of the passengers. However, the latter variable should be considered so that all aspects of the disruption are captured in the simulation. By extending the model to a dynamic version, we will take into account all these important features and make the model more realistic. Considering schedules, load factors, and operational constraints on aircraft can help analyze the quality of the solution. On the other hand, it can reduce the state of the decision space and the number of alternatives to be computed.

References

1. Bureau of Transportation Statistics: Airline On-Time Statistics and Delay Causes. U.S. Department of Transportation (2023). https://www.transtats.bts.gov/ot_delay/OT_DelayCause1.asp?20=E
2. Michaels, W.: The story of the world's first airline (2012). https://www.iata.org/en/about/history/flying-100-years/firstairline-story/
3. Airbus. A380. https://www.airbus.com/en/products-services/commercial-aircraft/passenger-aircraft/a380
4. Sun, X., Wandelt, S.: Robustness of air transportation as complex networks: systematic review of 15 years of research and outlook into the future. Sustainability **13**, 6446 (2021)
5. Gluchshenko, O., Foerster, P.: Performance based approach to investigate resilience and robustness of an ATM system. In: Tenth USA/Europe Air Traffic Management Research and Development Seminar, Chicago (2013)
6. Lordan, O., Sallan, J.M., Simo, P., Gonzalez-Prieto, D.: Robustness of the air transport network. Transp. Res. Part E Logist. Transp. Rev. **68**, 155–163 (2014)
7. Zhou, Y., Wang, J., Huang, G.Q.: Efficiency and robustness of weighted air transport networks. Transp. Res. Part E Logist. Transp. Rev. **122**, 14–26 (2019)
8. Zhou, Y., Wang, Y.: Measuring the resilience of airlines operation networks (2021)
9. Cai, Q., Ma, C., Alam, S., Duong, V.N., Sridhar, B.: Airway network flow management using Braess's paradox. In: 13th USA/Europe Air Traffic Management Research and Development Seminar, Vienna, Austria (2019)
10. Li, M.Z., Gopalakrishnan, K., Balakrishnan, H., Pantoja, K.: A spectral approach towards analyzing air traffic network disruptions. In: 13th USA/Europe Air Traffic Management Research and Development Seminar (2019)
11. Cook, A., Tanner, G., Cristobal, S., Zanin, M.: Delay propagation-new metrics, new insights. In: Eleventh USA/Europe Air Traffic Management Research and Development Seminar. EUROCONTROL/FAA (2015)
12. Schneider, A.A.M.C.M., Andrade, S.H.J.S., Herrmann, H.J.: Mitigation of malicious attacks on networks. Proc. Natl. Acad. Sci. **108**, 3838–3841 (2011)
13. Cai, M.P.Q., Alam, S.: Interdependency and vulnerability of multipartite networks under target node attacks. Complexity **2019**, 1–16 (2019)

14. Zeng, A., Liu, W.: Enhancing network robustness for malicious attacks. Phys. Rev. E Stat. Nonlinear Soft Matter Phys. **85**(6), 066130 (2012)
15. Sydney, M.Y.A., Scoglio, C.M., Schumm, P.: Characterizing the robustness of complex networks. arXiv Networking and Internet Architecture (2008)
16. Bröhl, T., Lehnertz, K.: Centrality-based identification of important edges in complex networks. Chaos **29**(3), 033115 (2019)
17. Yu, D.C.E., Zhao, J.: Identifying critical edges in complex networks. Sci. Rep. **8**, 14469 (2018)
18. Sydney, C.M.S.A., Gruenbacher, D.M.: Optimizing algebraic connectivity by edge rewiring. Appl. Math. Comput. **219**, 5465–5479 (2013)
19. Fiedler, M.: Algebraic connectivity of graphs. Czechoslov. Math. J. **23**, 298–305 (1973)
20. Wandelt, S., Sun, X., Cao, X.: Computationally efficient attack design for robustness analysis of air transportation networks. Transportmetrica A Transp. Sci. **11**(10), 939–966 (2015)
21. Wang, J.W.: Robustness of complex networks with the local protection strategy against cascading failures. Saf. Sci. **53**, 219–225 (2013)
22. Beygelzimer, R.L.A., Grinstein, G., Rish, I.: Improving network robustness by edge modification. Phys. A Stat. Mech. Appl. **357**, 593–612 (2005)
23. Abbas, W., Shabbir, M., Jaleel, H., Koutsoukos, X.: Improving network robustness through edge augmentation while preserving strong structural controllability. In: 2020 American Control Conference (ACC), pp. 2544–2549. IEEE (2020)
24. Van Mieghem, P., et al.: Decreasing the spectral radius of a graph by link removals. Phys. Rev. E **84**(1), 016101 (2011)
25. Chan, H., Akoglu, L.: Optimizing network robustness by edge rewiring: a general framework. Data Min. Knowl. Disc. **30**, 1395–1425 (2015)
26. Djidjev, H.N., Thulasidasan, S., Chapuis, G., Andonov, R., Lavenier, D.: Efficient multi-GPU computation of all-pairs shortest paths. In: 2014 IEEE 28th International Parallel and Distributed Processing Symposium, pp. 360–369 (2014)
27. Park, J.S., Penner, M., Prasanna, V.K.: Optimizing graph algorithms for improved cache performance. IEEE Trans. Parallel Distrib. Syst. **15**, 769–782 (2004)
28. Han, S.C., Franchetti, F., Püschel, M.: Program generation for the all-pairs shortest path problem. In: 2006 International Conference on Parallel Architectures and Compilation Techniques (PACT), pp. 222–232 (2006)
29. Minieka, E.: On computing sets of shortest paths in a graph. Commun. ACM **17**, 351–353 (1974)
30. Delahaye, D.: Optimisation de la sectorisation de l'espace aérien par algorithmes génétiques, Ph.D. dissertation, ENAC, Toulouse, France (1995)
31. Patriksson, M.: The Traffic Assignment Problem: Models and Methods. Dover Books on Mathematics. Dover Publications (2015)
32. Yen, J.Y.: Finding the k shortest loopless paths in a network. Manage. Sci. **17**(11), 712–716 (1971). https://doi.org/10.1287/mnsc.17.11.712
33. Delahaye, S.C.D., Mongeau, M.: Simulated annealing: from basics to applications. In: Handbook of Metaheuristics (2018)
34. Floyd, R.W.: Algorithm 97: shortest path. Commun. ACM **5**, 345 (1962)