AIAA Guidance, Navigation, and Control Conference
13 - 16 August 2012, Minneapolis, Minnesota

AIAA 2012-4769

A Geometric Approach to Optimal Routing for Commercial Formation Flight

Thomas E. Kent* Dr. Arthur G. Richards†

This paper explores a geometric approach to finding optimal routes for commercial formation flight. A weighted extension of the classical Fermat point problem is used to develop an analytic solution to finding optimal routes, thereby reducing the complexity of the problem and enabling a quick evaluation. We then construct a method to decouple origin and destination nodes creating a vertex from which the route projects, along with loci of possible points of formation. This implementation enables us to take lists of routes and efficiently decompose them to find the optimal locations for flights to meet, fly in formation and then break away and continue on their solo paths. We look at a case study of creating formations from 210 transatlantic flights for a fleet size of up to 2, resulting in overall global approximate total fuel savings of 8.6%. Furthermore we explore heuristic methods to finding solutions when creating larger fleet size formations, indicating savings surpassing 10%

I. Introduction

SESAR (the Single European Sky ATM Research) projections indicate that from 2005, with an estimated increase in population and a growing reliance on high speed travel, the number of flights per day could double by the year 2020.[1] The implications of such a drastic rise are numerous; with a heavy dependence on fossil fuels,[2,3] increasing concerns for the environment[4,5] and an infrastructure not yet capable of such a demand,[6] alternatives to the way commercial flight operates today must be investigated. This paper explores one of these alternatives: formation flight, in an attempt to optimize current routes to decrease overall fuel burn.

A variety of research has been undertaken in a number of different areas affecting formation flight. The study of animal behavior, such as geese flying in a 'V' formation,[7,8] has always been of interest, while military aircraft have long flown in formation for defensive and communicative purposes.[9,10] However, more recent work into the aerodynamic effects of flying in close proximity[11] coupled with real-time flight tests[12,13] brings substance to the idea that flying in formation can reduce fuel burn. The ability to save fuel on long-haul flights, would not only save money, but could also increase performance factors such as range and speed.

One of the immediate benefits of flying in formation, over other proposed fuel saving methods,[14–16] is the relatively minimal change to the current infrastructure. The majority of today's commercial airliners can fundamentally observe a reduction in drag from formation flight.[17] Although the possibility of designing new planes in the future to take advantage of the aerodynamic benefits of this scenario would be a long term goal, in the short term it would not be a necessity.

While studies show a positive trade off between deviating routes in order to join formation and drag reduction benefits,[18–21] few have tackled the rather expansive problem of global routing for commercial formation flight. The inherent complexity of analysing an increasing number of flights means that we need to approach this in a clever way. Both a centralized and decentralized approach are explored in Ref. 22, wherein the computational complexity restricts the analysis to the two-aircraft case. The incorporation of 'proposal marriage' type algorithm explores the idea of joining formation in an ad-hoc fashion. Route optimization studied in Ref. 17, along with a case study, shows significant cost saving percentages. By using a more in depth optimizer, solutions obtained retain many of the restrictions imposed by todays infrastructure.

In order to evaluate completely a global optimum for formation flight within today's flight structure, hundreds of variables need to be considered. Moreover the increasing number of commercial flights and the

*PhD Student, Department of Aerospace Engineering Bristol University
†Aerospace Engineer, Department of Aerospace Engineering Bristol University

overtly combinatorial nature of the problem makes this task heavily computational. This paper approaches the problem in a simplified manner, in an attempt to gain insight into very large scale behavior without being computationally prohibitive. One of our most important assumptions is the removal of the dependence on time, gaining optimal solutions for flights when they are not required to take off or land in a specific time window. Although the impact of wind speed can have a large effect on route planning this paper does not try to address this but is rather left for future work. We also provide a cooperative solution, whereby aircraft join formations for the overall benefit of the fleet, not personal gain. However, results containing negative implications for certain aircraft can be easily removed. This essentially reduces the global problem to a weighted shortest-path problem. The idea being, if we know the absolute optimal point for formations to meet and break away, we can observe which variables would need adjusting to closely approximate this solution and its underlying cost reductions.

The following sections outline both an analytic method for finding the optimal routes and the best way to pair a list of them for a global optimum. A case study of 210 transatlantic routes and it's results are explored in section IV, which leads us to examine possible heuristic techniques for finding solutions for much larger problems.

II. A geometric method for finding optimal join points for formation flight

Abstracting the problem of formation routing to a simpler geometric approach, for two arbitrary routes wanting to join in formation, we can imagine three distinct airports as three distinct points on the plane, two departure airports A and B, say, and then by assuming they are flying to a common destination airport C, we reduce the problem to finding some point P (as in figure 1(a)) joining A, B and C together such that the sum of the arc lengths are minimal (*i.e.* the cost per unit distance is minimised).

A. The Fermat point problem

1. A Non-weighted solution

We first assume an equal distance cost for travelling along each of these arcs by looking at the 'Fermat point problem'.[23,24] This is a classical mathematical problem posed in the late 17th century, it states, given a triangle, ABC, on the plane, find a point P such that the sum of the distances $||\vec{AP}||$, $||\vec{BP}||$ and $||\vec{CP}||$ is minimal. Over the years Mathematicians have posed numerous ways of finding this point P, including derivative based methods, the use of mechanics and Fermat's elegant geometric solution.

This paper explores an adaption of the original approach, first proposed via a series of letters between the mathematicians Fermat and Torecelli,[23,24] creating a solution based on the geometric dualities of triangles and circles.

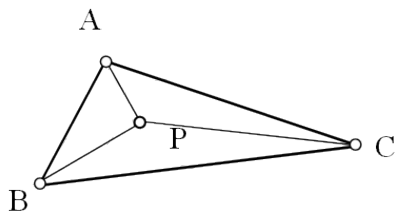
If we take a triangle ABC and construct outwardly three equilateral triangles along, and with side lengths corresponding to, the arcs AB, BC and CA. Then the lines subtending the outer vertex of each new triangle to its opposite vertex of the original intersect at a single point. This intersection is our desired point P (sufficiencies ensuring certain types of solution are explored in Ref. 25) which minimises the sum $||\vec{AP}|| + ||\vec{BP}|| + ||\vec{CP}||$. Similarly we can observe the same result by constructing the corresponding circumscribed circles of each of these three new equilateral triangles, creating a concurrency at a point P which is optimal.

One notable observation is the angle at which these arcs intersect.[26] For a 'weight-free' solution the angles of intersection $\angle APB$, $\angle BPC$ and $\angle CPA$ are all $120\,^\circ$. This result holds true with many studies of minimization observed in nature. For example the hexagonal structure of a honeycomb,[27] minimal surfaces in soap film experiments[28,29] and even molecular arrangements[30,31] all exhibit $120\,^\circ$ angles.

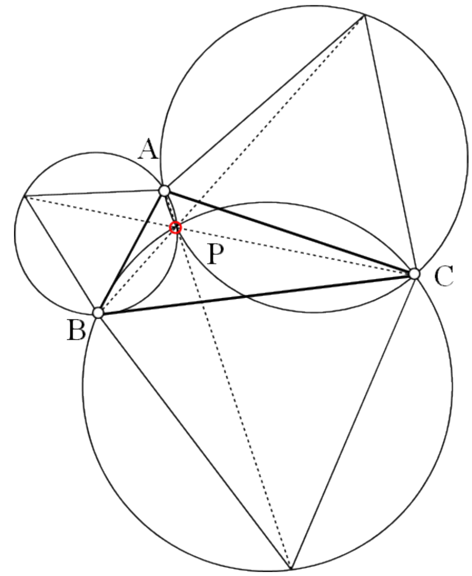
2. A weighted extension

In order for this geometric method to be practical for solving formation flight routing, we must first introduce a notion of weighting to represent differing cost per unit distance. Take three vertices A, B and C and their join point P. We then introduce scalar weights w_A, w_B and w_C corresponding to each of the arcs AP, BP and CP respectively, reducing our problem to minimising

$$w_A||\vec{AP}|| + w_B||\vec{BP}|| + w_C||\vec{CP}||. \qquad (1)$$



(a) Triangle ABC with possible join point P



(b) Circumscribed circles and subtending lines concurrent at an optimal point P

Figure 1. A Fermat-Torricelli geometric construction solution

If we imagine a table with three holes drilled representing the locations of the points A, B and C. Then at each of the holes a massless, frictionless string is passed through and the corresponding weight is tied to one end. We then tie the other ends of these three strings into a single knot. Letting this system settle will result in a natural mechanical equilibrium. This analogy coupled with the minimal energy principle[23] implies that the point the knot reaches on the table at mechanical equilibrium is identical to that which minimises the equation (1).

Therefore we develop an adaption of the figure 1(b) for weighted arcs. Moreover for the triangle ABC we seek a vectorial equilibrium about the point P so that,

$$w_A \frac{\vec{PA}}{||\vec{PA}||} + w_B \frac{\vec{PB}}{||\vec{PB}||} + w_C \frac{\vec{PC}}{||\vec{PC}||} = 0. \tag{2}$$

By applying the law of cosines to the three vectors in (2), we obtain expressions θ_A, θ_B and θ_C for the intercection angles $\angle BPC$, $\angle APC$ and $\angle APB$ respectively, based only on the input of the three scalar weight values w_A, w_B and w_C.[23]

$$\theta_A = \cos^{-1}\left(\frac{-w_B^2 - w_C^2 + w_A^2}{2w_B w_C}\right), \theta_B = \cos^{-1}\left(\frac{-w_A^2 - w_C^2 + w_B^2}{2w_A w_C}\right), \theta_C = \cos^{-1}\left(\frac{-w_A^2 - w_B^2 + w_C^2}{2w_A w_B}\right). \tag{3}$$

B. Application for formation flight

Incorporating a notion of weighted arcs permits us to more realistically assess the problem. We can see from equation (3), that using three equal weights we obtain the reassuring angles of $120\,°$. In the case of formation flight, however, these weights are not equal, in fact studies by Ref. 13 and Ref. 17 expect very reasonable drag savings (and thus a relative reduction in fuel burn) for aircraft flying in the up-wash of other formation members. For the purposes of this paper we have assumed average formation fuel burn figures in table 1 (estimates from Ref. 18–21 for varying fleet sizes) as our weights, however, we could easily introduce numerous other metrics in order to further assess the problem.

In terms of weighting this means that at the formation stage of the flight, on average, each fleet member uses the proportion $w_{f,n}$ of fuel. As there are n members in the fleet, the total estimated fuel burn on the formation arc is therefore $n \times w_{f,n}$. On the arc that corresponds to solo flight, each member experiences their normal fuel burn so are weighted as 1. This unitary weighting can however be adjusted to take into

Size of fleet (n)	1	2	3	4	5	6	7
Weight per fleet member ($w_{f,n}$)	1	0.9	0.85	0.82	0.8	0.785	0.775

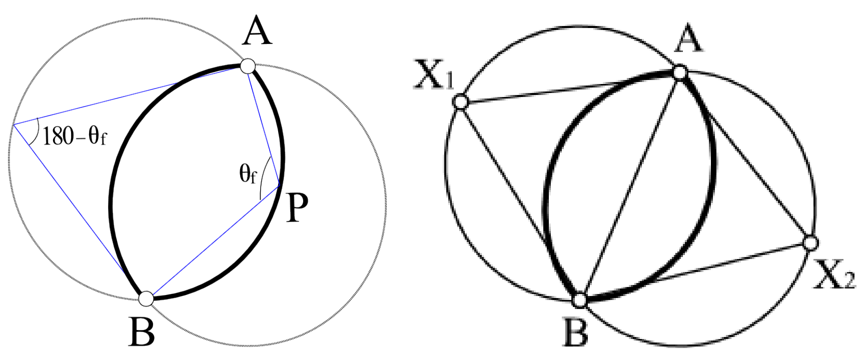
Table 1. Weight values to simulate proportional formation fuel distance for fleets of size n

account joining aircraft who do not burn fuel at the same rate. The three weights would need to be adjusted to accommodate proportional differences along both the solo paths and formation path.

For simplicity we use unitary weights for each flight now consider, for example, for two flights leaving airports A and B and travelling to a common destination C, who want to join in formation via point P. Here we have a fleet size of $n = 2$ and therefore the weights of the arcs AP and BP are 1 and for the formation arc PC it is $2 \times w_{f,2} = 1.8$. Using equation (3) we can then obtain the angles $\theta_A = \theta_B = 154.15\,^\circ$ and $\theta_C = 51.68\,^\circ$, at which they must meet to minimise their cost. The three points, A, B and C, coupled with the three angles, θ_A, θ_B and θ_C, define a single unique point that minimises the sum of the weighted distances connecting them. This point is the desired point P where the two flights meet for minimal fuel burn over their total journey.

1. *Loci of possible formation join points*

Knowing the three weights, and therefore the specific angles of interception θ_A, θ_B and what we will refer to as the formation angle, θ_f, enable us to abstract further and eliminate the need for a fixed destination vertex C. We have two fixed points A and B and an angle θ_f at which the trajectories meet. This knowledge enables us to create loci of possible formation points. In turn we can construct two corresponding inscribed circles with A and B on their perimeter. Each circle is comprised of two arcs, the first (the one we are interested in) contains, on its boundary, all the points P such that $\angle APB = \theta_f$, *i.e.* they meet at the angle required by equation (3), the other, all the points that meet at $180° - \theta_f$ as in figure 2(a).



(a) Inscribed loci of possible formation points given θ_f (b) Back vertices of optimal trajectory ensure all three intercept angles are satisfied

Figure 2. Possible solution points given an angle of interception

Analogous to the method of outwardly constructing equilateral triangles of the weight free geometric solution, we can outwardly construct two similar triangles ABX_1 and ABX_2 along the arc AB whose side lengths are in the same proportions as the weights[23] (i.e. the ratios $w_A : w_B : w_C$ and $||\vec{AX_i}|| : ||\vec{BX_i}|| : ||\vec{AB}||$ are equivalent) generating two back vertices X_i ($i \in \{1, 2\}$). Thus each of the inscribed circles is also a circumscribed circle for one of the triangles ABX_i (figure 2(b)). Therefore for any pair of nodes $\{A, B\}$ with three weights w_A, w_B and w_f, we can construct two inscribed circles each containing a back vertex. We can now reintroduce a destination node C and work out the optimal point to join formation. This is done simply by working out which arc X_iC crosses the boundary of its circle at the the required angle θ_f, this crossing point is the desired P. What this means is that for any destination point C the optimal route $\{APC, BPC\}$ is such that X_iPC is a straight line ($i \in \{1, 2\}$). The back vertex is the point at which the route appears to come and by projecting from this vertex we ensure not only that all three angles θ_A, θ_B and θ_f from (3) are satisfied but also that this join is optimal.

2. *Optimal join between loci*

As we know the loci of possible join points a priori of a destination. We can now assess the more realistic problem of having two routes with distinct departure and destination nodes. Then not only can we work out where we should join for optimal formation flight, but also where a formation should break away.

For two solo routes AC and BD (figure 3(a)) we first work out the circles and back vertices for each pair $\{A, B\}$ and $\{C, D\}$. Then we find the arc joining a back vertex X_i of $\{A, B\}$ to a back vertex Y_j of $\{C, D\}$ $(i, j \in \{1, 2\})$, such that it crosses both circles at the required formation angle w_f, (figure 3(b)) resulting in two crossing points P and Q which are the respective join and break points of the formation (figure 3(d)). If no single arc exists that satisfies w_f on both circles, then the optimal path is the shortest path between either X_i and C or D, or Y_i and A or B so that w_f is satisfied only once.

3. *Extension onto the sphere*

We should note here that the original problem, and above described adaption, is inherently planar. As such planar solutions for routing for formation flight will not necessarily be optimal on the globe. The properties of a curved surface mean it is impossible to find a $2D$ earth projection system which is isometric[32] (*i.e.* preserves both angles and distances). We can however, take the earth to be spherical and translate our method for use in spherical coordinates. We increase the dimension of each element of our method. Straight lines become planes, intersecting the earth through its centre creating great circle paths. Our inscribed circles become inscribed spheres, which (as we are only interested in its perimeter), intersect the globe along a planar surface defined as a spherical cap (or small circle). This enables us to take latitudes and longitudes of points and analogously solve our problem on the sphere.[33,34]

C. Verification

Figure 4 shows the difference in total formation distance of our geometric solution against that of an exhaustive search of 5000 random routes (evaluating possible join points at increments of 0.01 degrees of latitude and longitude). Figure 4(a) shows the frequency of a difference in solution. There are no instances of the geometric method giving a worse result and it is clear that the geometric method accurately finds the optimal point of formation, whilst taking a fraction of the time. Mathematical proofs for Fermat point problems of this type (both planar and spherical) are fairly abundant: for a deeper understanding of these available methods the author invites you to read Refs. 33–37.

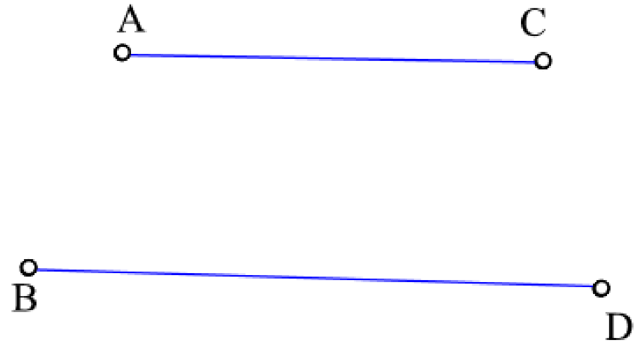
III. An extension for larger fleet sizes

A. Decoupling the problem

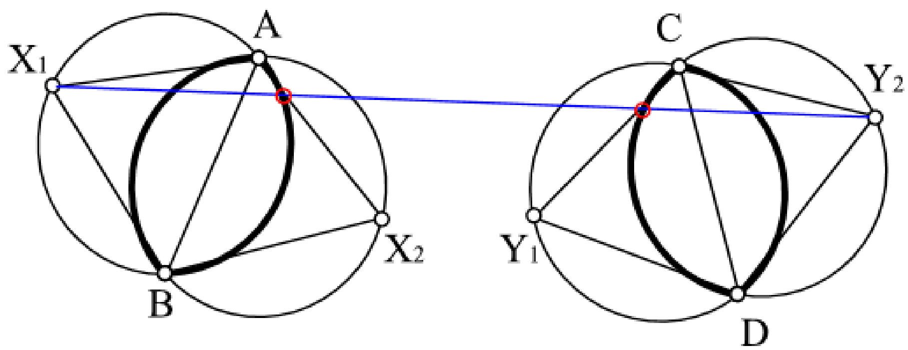
The above is a quite powerful result because it enables us to easily decouple the problem, reducing pairs of nodes to their back vertices and inscribed circles. This information is a priori of a destination, depending only on the relative weights and a fixed pair of nodes. Using this fairly simple method of projecting from a back vertex also means we can not only solve the 2 aircraft case, but theoretically the N aircraft case.

Take for example two routes *RouteA* and *RouteB*, by finding the back vertices X_A, Y_B who's arc crosses at the required angles, we can create a 'virtual route' *RouteAB*, whose projected route is going from X_A to Y_B. A third route *RouteC* can then be added. This is done just as before, only we have to adjust the arcs' weighting to take into account the new size of the formation at each stage of the route. We then solve the augmented problem where *RouteAB* and *RouteC* should join (P_{ABC}) and break away (Q_{ABC}), with weightings $w_{RAB} = 1.8$, $w_{RC} = 1$ and $w_{RABC} = w_{f,3} \times 3 = 2.55$. All that is left is to split *RouteAB* back to two separate routes and update P_{AB} and Q_{AB} based on their new destinations P_{ABC} and Q_{ABC}.

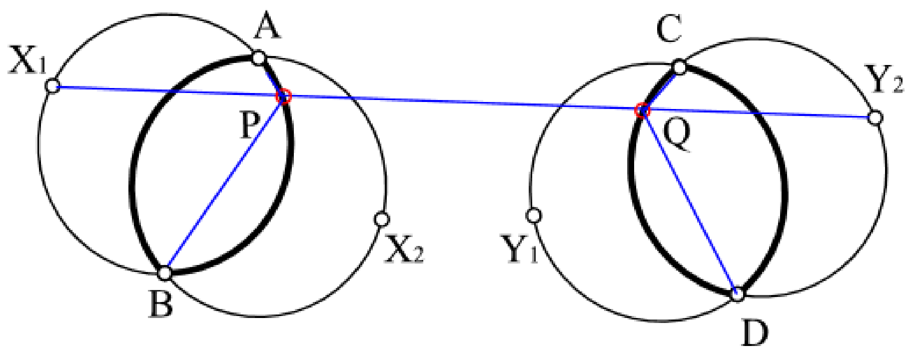
Figures 5(a)-5(d) depict the case where we assume *RouteA* joins *RouteB* then *RouteAB* joins *RouteC*, and breaks away in a similar way. However, realistically we want to find the order of joining that minimises total fuel burn for all flights. Therefore the various combinations of the order of joining formation, including scenarios whereby it might be optimal for only *RouteA* and *RouteC* to join, with a solo *RouteB*, must be computed.



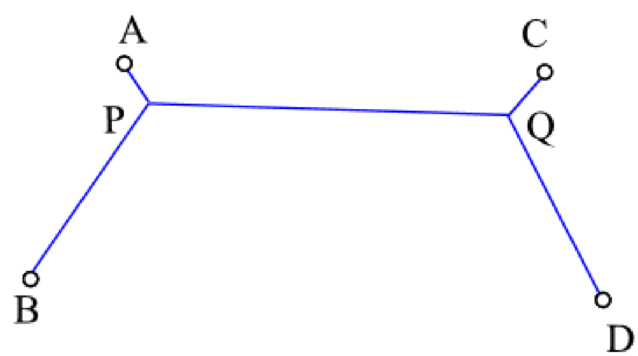
(a) Solo routes



(b) Trajectory between back vertices crossing loci of possible optimal points

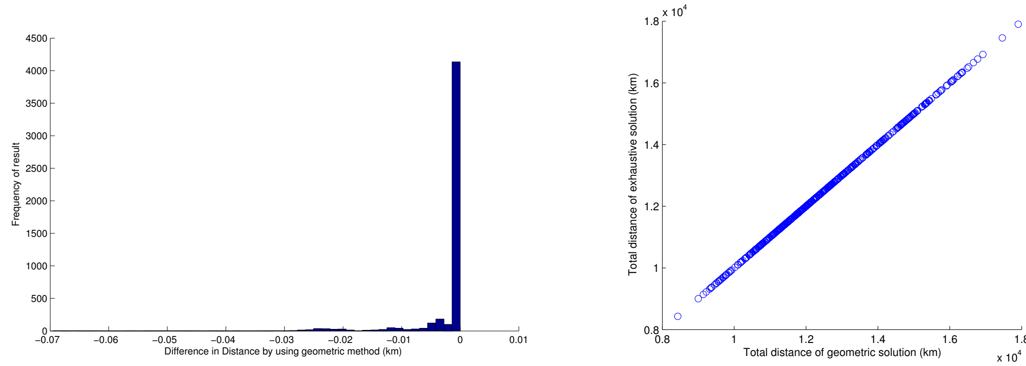


(c) Trajectory intercepet of loci



(d) Optimal route

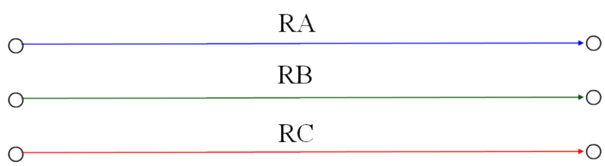
Figure 3. Join and break points for two distinct routes



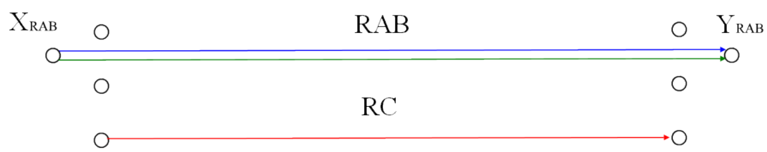
(a) Frequency in deviations between geometric solution and exhaustive search at 0.01 increments

(b) Distances of geometric solution against exhaustive search

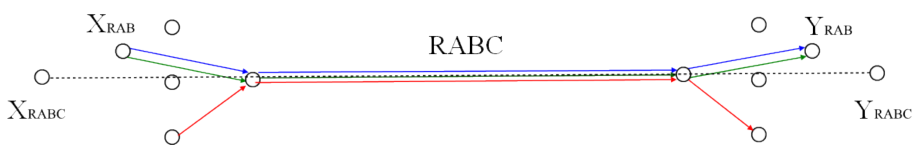
Figure 4. Deviations in results between geometric solution and exhaustive search at 0.01 increments



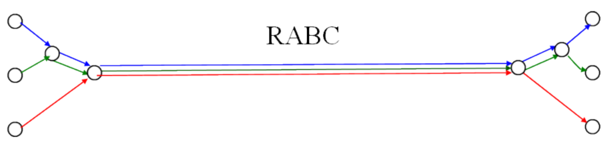
(a) Solo routes *RA*, *RB* and *RC*



(b) Create projected 'virtual-route' *RAB* between back vertices X_{RAB} and Y_{RAB}



(c) Join *RAB* with *RC* via new back vertices X_{RABC} and Y_{RABC}



(d) Update *RA* and *RB* join points given the future join

Figure 5. Join and break points for three distinct routes

B. Method for creating fleets of size 2 and 3

For any 3 distinct routes, for formations of size 2 there are four combinations. When trying to find fleets of up to 3 there is an additional nine combinations, which consist of 2 choices from 3 forms, one for the join up and one for the break away. If for example we take the two routes

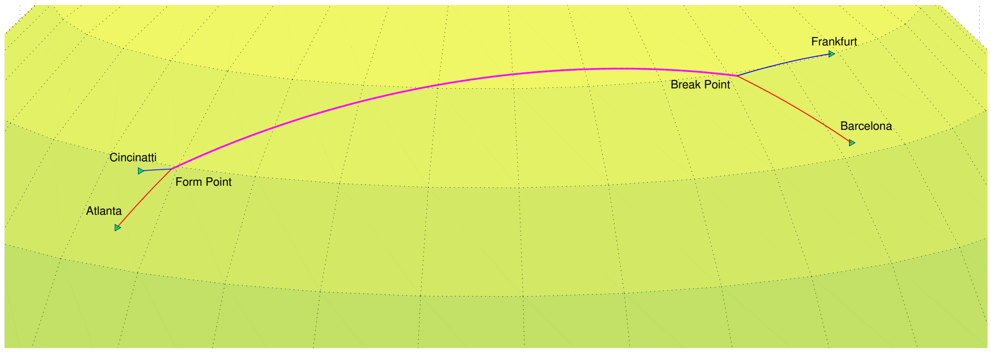
$$RouteA = \{\text{Atlanta},\ \text{Barcelona}\},\ RouteB = \{\text{Cincinnati}, \text{Frankfurt}\}, \tag{4}$$

and by using the above methodology with weight values from table 1 we can obtain the desired points for formation flight. Figure 6(a) shows the formation of RouteA and RouteB. The total great circle distance, and therefore fuel burn distance, for *RouteA* and *RouteB* as solo routes is 14359 km, when flown in formation the fuel burn distance reduces by 737 km to 13622 km. This equates to a rather significant saving of about

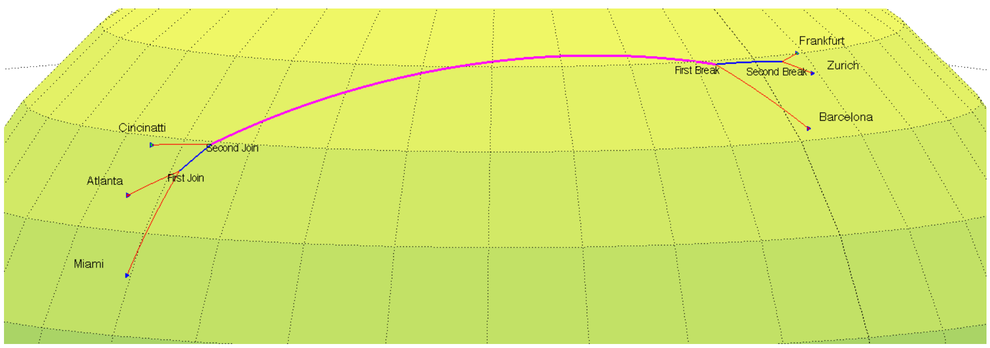
5.12%. By then adding a third route

$$RouteC = \{\text{Miami},\ \text{Zurich}\}, \tag{5}$$

we can run a similar implementation to find the optimal ordering of join and break points and their respective locations (as shown in figure 6(b)). The order of joining formation is $RouteA$ joins $RouteC$, then $RouteAC$ joins $RouteB$, followed by $RouteA$ breaking away from $RouteBC$, then $RouteB$ breaking from $RouteC$. The resulting saving is about 8.37%.



(a) Optimal formation route for the two routes $RouteA$ and $RouteB$



(b) Optimal formation route for the three routes $RouteA$, $RouteB$ and $RouteC$

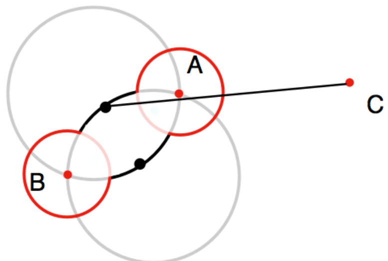
Figure 6. Optimal Join and Break points for fleet size 2 and 3

This outlines a simplistic framework for deciding the locations where fleets of size 2 and 3 should join up and break away in order to minimise total fuel burn. By decomposing larger lists of routes into combinations of subproblems of size 2 or 3 we can, in principle, solve for N routes.

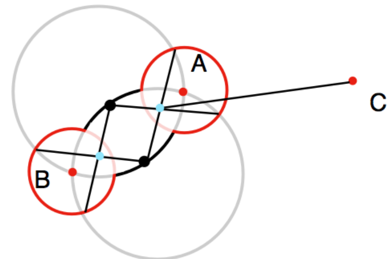
C. Incorporating a minimum distance to climb

Some of our early results indicated that the best savings to be had where when flights had a common departure or destination airport. Others tended to be between airports of close proximity. This meant that many join points were in fact at the airports themselves. Although this seems like a reasonable result, practicality issues could likely prohibit such a route. In this scenario flights would need to take off in formation (possibly on a parallel runway which would rule out many airports) and then engage in a series of step climbs in formation until they reached a cruising altitude. The implications of this along with the difficulty of achieving formation drag savings along the way directed us to look at only joining formations once they are at a cruising altitude.

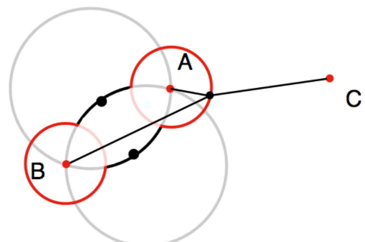
Implementing this into our current framework is fairly straightforward. First assume there is a circular region of a predefined radial distance (a horizontal change in distance between take off and an altitude at which formations can be joined) around each airport. If the optimal formation point lies within any of those regions then it cannot be used, however it can be moved onto its perimeter. To find the best point on this perimeter we project from a new point (simulating the equilibrium of a moment about an arm centred at the point A or B.) as shown in figure (7(b)), determined by intersections of the lines between the back vertices and the points where the perimeter intersects our arcs of optimality. Where this new projection intercepts the perimeter of the restricted region is our new desired meeting point (figure (7(c))).



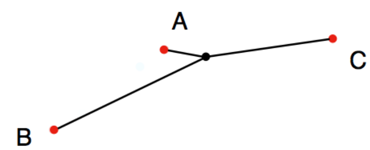
(a) Initial attempt too close to airport A



(b) Work out new projection point and new intercept



(c) Formation meets at new intercept point



(d) Formation meets at required distance away

Figure 7. New join point required to be at least a certain distance from each airport

This allows us to prescribe distances which the flights must be away from the airports before they can join a formation. Not only does this introduce a further aspect of realism, but could also be used to avoid joining formation around a busy airspace.

IV. Transatlantic flight case study for pairs

Using the methodology of the previous sections, we examine a list of 210 real transatlantic flights departing from 26 US airports and flying to 42 European airports. We analyse the potential savings of formation fleets of size 2. This study assumes that each plane is identical, that is, the reduction in drag and the relative fuel burn of each aircraft is assumed to be consistent; with formation weightings based on table 1 (Our simplistic weighting strategy can, however, be adjusted to incorporate any proportional differences). Furthermore we do not deal with the issue of cooperative fuel saving distribution between specific airlines, we treat the list as a single company. In keeping with our assumptions the results of table 2 are also time-free, based on the optimal positions for joining a fleet and breaking away. Lastly as discussed in section III(C) we include a minimum horizontal distance of 320km that the join points must lie away from each airport in order to allow each plane to reach a cruising altitude.

We firstly Computed the best formation route for all 22791 possible pairs of flights, taking a total of just 2.47 seconds on a 3Ghz desktop PC. Then using AMPL (A Mathematical Programming Language) we ran a MILP (Mixed Integer Linear Program) to select optimal combinations of formations, which solved in about 10 seconds, and resulted in all 210 routes being paired into 105 formations of size 2 (as in Figure 8)[†] .

Table 2 shows for the two-aircraft case the overall average deviation in distance is about 10km, which at Mach 0.84 is a difference of less than a minute. Therefore time implications from the non-direct routes are, on average, not too substantial. More significantly, the results suggest, on average, a significant saving of 8.643%. Crude cost estimates suggest a monetary saving of around \$1066.9 per aircraft (Based on a cost per gallon of fuel or \$3 at 1.65km per gallon[38]).

For the globally optimal combination, the majority of formations were made between routes which re-

[†]More maps available at `http://arthurrichards.blogspot.co.uk/2012/07/optimal-transatlantic-routes-for.html`

Figure 8. Transatlantic formation routes

	Average per aircraft
Solo distance (km)	6711.5
Formation distance (km)	6721.2
Deviation in route distance (km)	9.75
Effective fuel distance (km)	6124.7
Effective fuel distance saved (km)	586.8
Percentage saved (%)	8.643

Table 2. Per flight averaged results for joining transatlantic routes of fleet size 2

quired little deviation from their original path. That is, between routes whose departure airport, destination airport or flight path were in close proximity to each other. Even though distance to climb restrictions were implemented, many pairings found the best gain to be between other flights that shared either their departure or destination airport. If for example we look at Atlanta Hartsfield-Jackson (ATL) airport, then 18 of their flights make up 9 of the pairs. Even though they do not start to benefit from formation fuel savings until they are 320km away it is clear that they do not deviate greatly from their original paths. Although this paper does not attempt to asses how to schedule such flights, it is interesting to note that ATL airport has 5 parallel runways, which could be a viable option for such a scenario.

V. Methods for scaling up to larger formations

The methodology outlined in the previous sections enable us to rapidly enumerate potential formations between routes. We can therefore take a list of solo routes, and return a list of favourable formation routes (routes that benefit from joining in formation) in a relatively short amount of time. The combinatorial nature, however, of enumerating all possibilities means that as we increase the size of the route list the number of favourable combinations also increases. In fact the number of possible combinations of choosing k (our fleet size) from a set n (our route list size) is governed by binomial coefficients $n\mathbf{C}k = \frac{n!}{k!(n-k)!}$

This implies that, although the problem is relatively scalable, there are computational restrictions on what size of problem is feasible. Table 3 outlines the computational times of 6 different scenarios based on the number of possible combinations to be evaluated. It is therefore clear that for increasingly large lists of solo routes total computation time dramatically increases and this combined with increasing the fleet size can make it impractical to enumerate in a reasonable time. The increase in computation time for a single combination is reflected in the complexity of a particular join. For example for two solo routes joining together there is only one way for them to join and one way for them to break way. For three routes there 13, firstly 3 for them to join, with each of those having 3 different ways to break away, then there are further 3 possibilities where only two join and one flies solo, then lastly all three fly solo, totalling 13. If we consider a fleet size of 4 there are 1816 different possible formation topologies. In this scenario not only are we looking at more combinations, but each of those possibilities we need to evaluate dramatically more topologies. Enumerating combinations, however, is a highly parallelisable task, and so larger lists could easily be run on a computer-cluster and therefore mitigating much of the time dependance on the enumeration stage.

This realisation tells us that in order to examine a growing fleet size and a growing route list in a reasonable time we will need to consider possible heuristic methods to reduce the need to evaluate all combinations. Moreover, introducing further aspects of realism in to the problem such as time, will in fact benefit us computationally. The issue we must therefore next address is the impact the size of a problem has on the optimization.

	Fleet size 2		Fleet size 3	
Size of Route list	Combinations	Wall time	Combinations	Wall time
75	2,775	0.31s	67,525	6m 54s
210	21,945	2.47s	1,521,520	2h 35m 42s
3279	5,374,281	10m 7s	5,870,506,279	1y 51d 22h 28m*

Table 3. Possible combinations needing to be evaluated for varying sizes of route list and fleet (Times are based on a dual-core 3 GHz desktop with 2GB of RAM) (*) estimated

A. Optimisation techniques for finding a solution

Once a list of all favourable formations has been enumerated, we need to take that data and find the optimal choices to create our final solution. In the case study of section IV we used a generic MILP (Mixed Integer Linear Program), implemented in AMPL with CPLEX. The MILP solver within CPLEX calls upon a number of different algorithms (mainly branch and bound and cutting plane) and is arguably one of the best commercial solvers available for this.[39] Solving in such a way is highly effective for smaller problems. However, a MILP is NP-hard under certain conditions, such as number of variables, number of constraints and the convexity of the problem.[40] The non-convex nature of our problem, i.e. there are many possible local minimum, means that finding a global minimum is already a difficult task. Therefore as we increase the size of the problem (the number of variables) we also increase the resources needed to solve it. Due to this we could not solve effectively using a MILP for problems consisting of more than around 500,000 favourable combinations. If we took an example list of 3279 routes going between major cities worldwide, we can enumerate the 5 million possible pairings in around 10 minutes. The resulting favourable formations for this particular set of routes is around 70%, or some 3.5 million possibilities. Therefore the problem which we need our MILP to solve has 3.5 million variables and is infeasible.

There are a numerous ways to go forward with such a combinatorial problem. The two main directions are to either reduce the number of favourable combinations in order to reduce the number of variables in our problem or to find an approximate global solution. The first approach would involve introducing heuristics in an attempt to eliminate a large proportion of our variables so as to then find a globally optimal solution using a MILP. These may include minimal saving constraints, such that combinations with small overall savings would not be considered as impacts from time or weather could easily mitigate such savings, or maximum deviation constraints, so a flight's total distance does not impact too drastically on the duration of a flight. Although as explored later in this section we can take an already solved smaller sub-problem and build upon. The second approach would to use heuristic algorithms to find a 'good' solution in a feasible amount of time. We consider one such an approach in the following section by implementing a Simulated Annealing algorithm to find solutions to the problem of picking the best combinations from a list of favourable possibilities.

Ideally we would use a mixture of both approaches to find a reasonable solution. It would also be helpful to implement heuristic constraints in such a way as to increase the realism of the model rather than a simple pruning of viable solutions which match a criterion.

B. Simulated Annealing

The simulated annealing algorithm is a probabilistic metaheuristic for finding solutions which are 'acceptable' rather than necessarily globally optimum.[41–44] It is based on a controlled cooling technique used in metallurgy. At each step we consider a perturbed neighbouring state and probabilistically decide wether to move to this new state or stay with the previous one. The notion of a temperature means that while at higher temperatures we are more likely to accept a new state that is worse in the hope to increase the search space and to eventually result in a better solution. Then as the temperature lowers we converge towards the best local solution.

The algorithm, as described in Ref. 45, is a basic framework, which we will adapt to suit our particular problem. We firstly use an upper-triangular matrix of costs for each possible favourable combination, where a row i and column j (for $i \leq j$) correspond to the precalculated cost of flight i joining flight j (a solo flight is along the diagonal where $i = j$). We also use a symmetric state matrix s consisting of binary values. If the current state has flight i pairing with flight j in it then $s(i, j) = s(j, i) = 1$ otherwise it is zero. Setting it up in this way allows us to do two things. Firstly we can easily calculate the cost of a given state by element-wise matrix multiplication, as the cost Matrix is upper-triangular this means that even though there are two binary values in s for a distinct pair $\{i, j\}$, their cost is only added once. Secondly having a symmetric state matrix gives us an easy parity check, by summing along the rows or columns of s, we can see if any flights are not allocated either a formation or solo route.

The method used to decide on a neighbouring state is itself a random process upon the current one (outlined in algorithm 1). Firstly we produce a random number between 1 and the number of favourable formations. That number then represents either one of the formations or one of the solo routes. If the pairing is part of the current state, take it out, and if it is not, put it in (making sure any overlapping pairs are changed to solo flights).

1. *Simulated Annealing Results*

The Simulated Annealing algorithm is an effective way of finding 'good' solutions in a fixed amount of time. The time of each iteration is dependent on the size of the problem, however, once this is known we can set specific walltimes for the algorithm to run in order to find a solution. It is worthwhile to note here that some parameters of the basic algorithm should be adjusted on a problem to problem basis. Moreover the annealing schedule is very important, as it dictates how long we are able to try out worse solutions and in doing so we can attempt to mitigate some of the impact of non-convexity of the problem and avoid many of the local optimums. Preliminary tests carried out indicated that the best annealing schedule for our scenario

Algorithm 1 Neighbouring state algorithm

```
procedure NEIGHBOUR(s, costMatrix)                          ▷ A perturbation about the state s
    FF ← list of indices of favourable formations   ▷ Each corresponds to a binary value in s and a cost value in costMatrix
    NF ← length(FF)                                         ▷ Including solo routes
    R ← rand(1, NF)                                         ▷ Pick a random formation
    index ← FF(R)                                           ▷ Get the location of this random formation
    if s(index) = 1 then                                    ▷ Already in state
        s(index) ← 0                                        ▷ Remove it
    else
        s(index) ← 1                                        ▷ Add it
    end if
    for i = 1..length(s) do
        if sum(s(i, :)) = 0 then                            ▷ Aircraft not allocated to a formation or solo flight
            s(i, i) ← 1                                     ▷ Put it in solo flight
        end if
    end for
end procedure
```

was the one relating to the Boltzmann distribution $T = \frac{T_0}{\log k}$. This schedule spends significantly longer at 'higher' temperatures allowing us to more thoroughly explore our search space. However it may still be the case that as we come close to the specified iteration limit, the temperature is still high (i.e. more that 0.01), in such a case we can easily switch to a faster annealing schedule to rapidly descend to our final solution.

In order to benchmark the quality of the algorithm, we must return to our transatlantic case study from section IV and assess the difference in solutions between using a MILP (which gives us the global optimum for pairs) and the SA (Simulated Annealing) algorithm. We can see from table 4 and figure 9 that by using an SA for pairs of transatlantic flights we can obtain solutions with a cost decrease of around 7.6%, getting very close to the optimal solution of 8.6% after a 100,000 or so iterations. The SA for finding pairs runs at about 200,000 iterations a minute which allows us to easily reach a reasonable solution in a small amount of time. If we ran it for a much larger number of iterations we could expect reduce the gap between the global optimum but with a likely diminishing rate of return. It is also important to note here that, as it is a stochastic process, we cannot guarantee a specific result as every run of the algorithm runs on a different set of random numbers. We can however include a very large iteration count and an acceptability factor so that the algorithm ends only after it has achieved a certain cost decrease.

Iterations	Walltime	Minimum (%)	Average (%)	Maximum (%)
25,000	08s	7.278	7.371	7.459
50,000	16s	7.320	7.518	7.698
100,000	29s	7.537	7.639	7.818
200,000	01m 02s	7.616	7.728	7.819
5,000,000	23m 31s	7.935	8.037	8.162
MILP optimal pairings	10s	-	8.643	-

Table 4. Cost percentage saving using Simulated Annealing against a MILP

2. Simulated Annealing for formation fleet sizes up to 3

The SA algorithm for fleet sizes up to 3 is adapted from the methodology above by using 3-Dimensional matrices for both the state and costs. We also use an initial state corresponding to (an already found) 'good' solution for pairs. Again as we are dealing with a much larger problem the computation times are effected, but unlike the MILP we can easily predefine a duration or iteration count. The algorithm currently runs at about 200 iterations a minute, however, there is plenty of room to improve its efficiency. The SA for fleet sizes up to 3 does however show some promising results. We can see from table 5 and figure 10 that it is

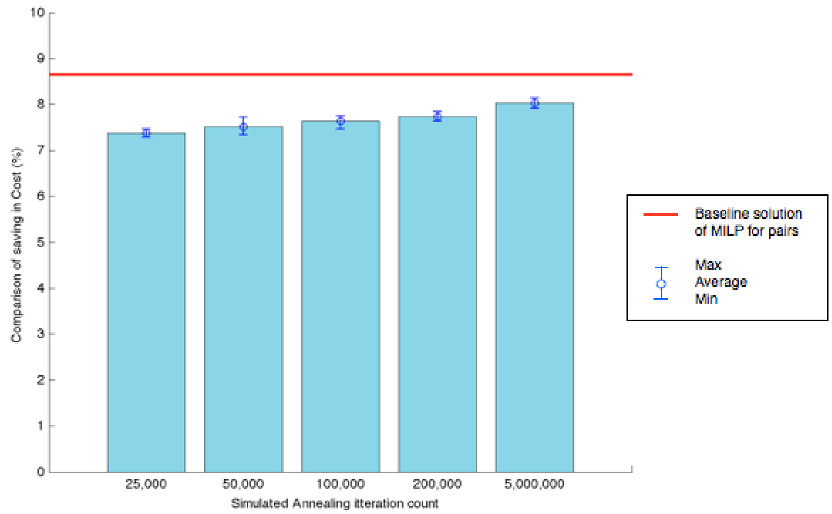


Figure 9. Simulated Annealing against a MILP for fleet size 2

possible to increase the total cost saving from 8.6% (from the MILP for pairs) to around 9.7% by letting fleet sizes be up to 3 bringing the overall average saving to almost 10%. This furthers the indication that it is worthwhile exploring larger fleet sizes.

iterations	Walltime	Minimum (%)	Average (%)	Maximum (%)
5,000	16m 21s	8.758	8.957	9.179
25,000	1h 21m 18s	9.050	9.340	9.563
50,000	2h 57m 58s	9.296	9.429	9.702
100,000	6h 01m 45s	9.403	9.556	9.803
200,000	10h 48m 46s	9.541	9.675	9.826
MILP optimal pairings	10s	-	8.643	-

Table 5. Effect of iteration count on percentage cost increase using Simulated Annealing against a MILP

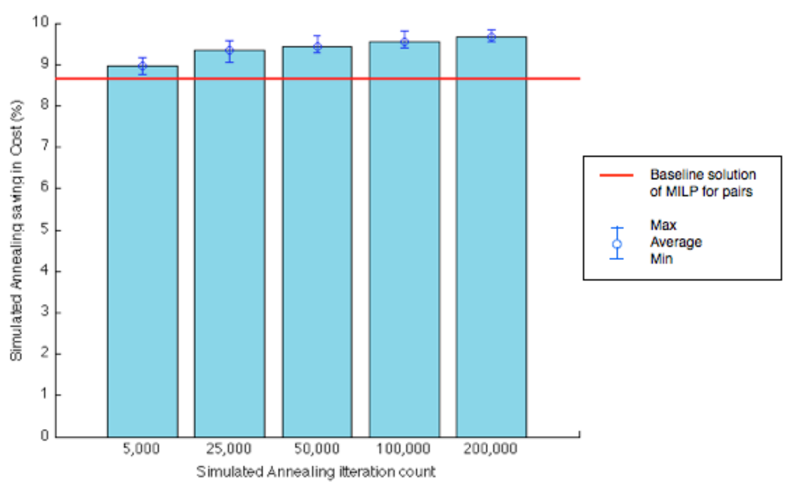


Figure 10. Percentage cost saving using formations of size three against solo flight

One of the requirements to using a SA algorithm is the having to define an initial state. This means however, that we can reuse previous results as initial states in a hope to better the result. A variety of research[46–49] has explored ways of using this to improve the quality of the solution. We could for example execute many parallel runs of the same algorithm systematically updating each one with the overall best solution found at given time intervals. These sorts of adaptions allow some scope for tailoring the algorithm to a specific problem in a hope to improve efficiency and results.

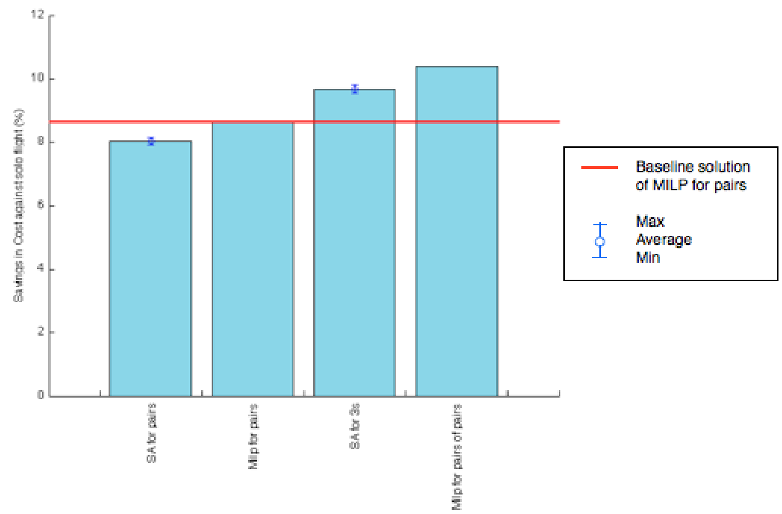


Figure 11. Comparison of savings in cost for different methods

C. Pairs of pairs

Finally we look at a heuristic way of building upon the optimal solution for pairs to create larger fleet sizes without the need to evaluate all combinations. We take the globally optimal set of 105 pairs for the transatlantic case study and try to pair each of those with another to make fleet sizes of 4. This is done much like before where we evaluate each combination trying to find the favourable ones. We use the approach outlined in section III of creating 'virtual' routes and then try to find the optimal points for these two virtual routes of size 2 to meet and break away (using updated path weightings based on table 1). We therefore only need to evaluate the ways of pairing 105 routes; some 5460 combinations rather than joining 210 routes into fleet sizes of up to 4 which results in close to 79 million combinations.

From the 5460 possible combinations evaluated (in less than a second) 1246 were favourable (around 20%). These were then optimised using a MILP in AMPL (as in section IV to find the optimal combination of fleets of size 4 or 2. The resulting solution consisted of 37 fleets of size 4 and 31 of size 2, equating to a total average saving against flying solo of 10.39%. This outlines a very simple method for creating larger fleet sizes using very little computational time but yielding an additional 1.75% above the optimal result for pairs (Figure 11) giving us a quick upper bound on future solutions.

VI. Conclusion

In this paper we have explored a geometric method for optimal routing for formation flight. An adaption of some basic geometric properties observed by Fermat have enabled us to decouple a complex problem, creating a possible method to find a global optimum for a list of N routes. The ability to create possible solution loci a priori of a destination reduces the degrees of freedom of the solution. The simplistic nature of the model enables us to find possible solutions to the routing problem very quickly. The method has been developed as such, in a hope to observe some of the principal factors affecting the locations of optimal formations. By running simulations for large lists of transatlantic flights it may be possible to gain insight into regions where formation flight is more beneficial.

One of the major assumptions of this model is that on the relaxation of a time dependency of flights. This may not be entirely realistic in the current airline infrastructure, however it can be used to gain an insight into what times different members of a fleet should leave in order to fly the optimal path. Moreover, results from section IV show that the average distance deviations, result in only a small increase to total flight time, while estimates show significant percentage savings. The impact of moving to a more dynamic problem as well as assessing uncertainty issues arising from factors such as weather are left for future work.

The combinatorial nature, and implied computational complexity, of the problem limits its potential for a global optimum for larger problems at this stage. We have however explored heuristic methods for finding 'good' solutions within reasonable time frames. Furthermore it is possible to incorporate a number of different constraints in an attempt to radically reduce the number of computations. For example we could reintroduce arrival and departure time dependence and only join routes which could do so in their required time frame. Our scalar method of arc-weighting could, in future, be adapted to introduce more of the decision variables which affect formation locations along with increasing the realism of our model. It would also be beneficial to incorporate a systematic method of building up to a global solution. By storing relevant information for each possible combination of routes in a fleet of size $n = 2$ or 3, we could increase n without having to recompute smaller fleets. Moreover, creating structured data for all fleet sizes allow us to easily extend formation sizes in a heuristic way in an attempt to reduce costs further.

References

[1]SESAR Definition phase 1: Deliverable 1. In *Eurocontrol*, pages 0–71. Tech. Rep, 2007.

[2]Shahriar Shafiee and Erkan Topal. An econometrics view of worldwide fossil fuel consumption and the role of US. *Energy Policy*, 36(2):775–786, February 2008.

[3]Shahriar Shafiee and Erkan Topal. A long-term view of worldwide fossil fuel prices. *Applied Energy*, 87(3):988–1000, March 2010.

[4]R Babikian. The historical fuel efficiency characteristics of regional aircraft from technological, operational, and cost perspectives. *Journal of Air Transport Management*, 8(6):389–400, November 2002.

[5]Julien Joseph Rojo. *Future Trends in Local Air Quality Impacts of Aviation*. PhD thesis, Massachusetts Institute of Technology, 2007.

[6]P Forsyth. The impacts of emerging aviation trends on airport infrastructure. *Journal of Air Transport Management*, 13(1):45–52, January 2007.

[7]Lisa Lofland Gould and Frank H Heppner. The vee formation of Canada geese. *The Auk*, 91:494–506, 1974.

[8]F Reed Hainsworth. Wing movements and positioning for aerodynamic benefit by Canada geese flying in formation. *Canadian Journal of Zoology*, 67(3):585–589, March 1989.

[9]C.M. Haissig. Military formation flight as a model for increased capacity in civilian airspace. In *The 23rd Digital Avionics Systems Conference (IEEE Cat. No.04CH37576)*, volume 1, pages 1.C.4–1.1–9. IEEE, 2004.

[10]Eric Klavins. Communication Complexity of Multi-Robot Systems. *Complexity*, (December):1–17, 2002.

[11]Andrew Ning. *Aircraft Drag Reduction Through Extended Formation Flight*. PhD thesis, Stanford University, 2011.

[12]M Jake Vachon, Ronald J Ray, Kevin R Walsh, and Kimberly Ennix. F / A-18 Performance Benefits Measured During the Autonomous Formation Flight Project. *Instrumentation*, (September), 2003.

[13]Ronald J Ray, Brent R Cobleigh, M Jake Vachon, and Clinton St John. Flight Test Techniques Used To Evaluate Performance Benefits During Formation Flight. 2002.

[14]R K Nangia. 'Greener' civil aviation using air-to-air refuelling relating aircraft design efficiency and tanker offload efficiency. *The Aeronautical Journal*, 2006(3186):589–592, 2007.

[15]Raj K Nangia. Way Forward to a Step Jump for Highly Efficient & Greener Civil Aviation An Opportunity for Present & A Vision for Future. *Fuel*, (December), 2009.

[16]N Qin, A Vavalle, A Lemoigne, M Laban, K Hackett, and P Weinerfetl. Aerodynamic considerations of blended wing body aircraft. *Progress in Aerospace Sciences*, 40(6):321–343, August 2004.

[17]Geoffrey C Bower, Tristan C Flanzer, and Ilan M Kroo. Formation Geometries and Route Optimization for Commercial Formation Flight. Number June, 2009.

[18]Rachel King and Ashok Gopalarathnam. Ideal Aerodynamics of Ground Effect and Formation Flight. *Journal of Aircraft*, 42(5):1188–1199, 2005.

[19]William Blake and David Gingras. Comparison of Predicted and Measured Formation Flight Interference Effects. *American Institute of Aeronautics and Astronautics*, 41(2):201–207, 2001.

[20]William Blake and Dieter Multhopp. Design, performance and modeling considerations for close formation flight. In *Proceedings of the AIAA Navigation Guidance and Control Conference*, number AIAA-98-4343, pages 476–486, 1998.

[21]P B S Lissaman and C A Shollenberger. Formation flight of birds. *Science*, 168(3934):1003–1005, 1970.

[22]Emilio Frazzoli. Efficient Coordination of Multiple-UAV systems of multiple-UAV systems. *Communications*, pages 1–30, 2003.

[23]Shay Gueron and Ran Tessler. The Fermat-Steiner Problem. *The American Mathematical Monthly*, 109(5):443, May 2002.

[24]Michael De Villiers. A generalization of the Fermat-Torricelli point. *The Mathematical Gazette*, 79(July):374–378, 1995.

[25]Yujin Shen and Juan Tolosa. The Weighted Fermat Triangle Problem. *International Journal of Mathematics and Mathematical Sciences*, 2008:1–16, 2008.

[26]Folke Eriksson. The Fermat-Torricelli Problem Once More. *The Mathematical Gazette*, 81(490):37, March 1997.

[27]L. Chen and The Catholic University of America. *Seismic performance of shear walls utilizing cellular material*. 2009.

[28]Jean E Taylor, The Annals, and No May. The structure of singularities in soap-bubble-like and soap-film-like. *Structure*, 103(3):489–539, 1976.

[29]Tobias H Colding and William P Minicozzi. Shapes of embedded minimal surfaces. *Proceedings of the National Academy of Sciences of the United States of America*, 103(30):11106–11, July 2006.

[30]Sara Fortuna, David L Cheung, and Alessandro Troisi. Hexagonal lattice model of the patterns formed by hydrogen-bonded molecules on the surface. *The journal of physical chemistry. B*, 114(5):1849–58, February 2010.

[31]Michael R Philpott and Yoshiyuki Kawazoe. Triplet states of zigzag edged hexagonal graphene molecules C(6m2)H(6m) (m = 1, 2, 3, ..., 10) and carbon based magnetism. *The Journal of chemical physics*, 134(12):124706, 2011.

[32]Waldo R. Tobler. A Classification Of Map Projections. *Annals of the Association of American Geographers*, 52(2):167–175, June 1962.

[33]Khuloud Ghalieh and Mowaffaq Hajja. 80.45 The Fermat Point of a Spherical Triangle. *The Mathematical Gazette*, 80(489):561, November 1996.

[34]E. J. Cockayne. On Fermat's Problem on the Surface of a Sphere. *Mathematics Magazine*, 45(4):216, September 1972.

[35]Anastasios Zachos and Athanase Cotsiolis. The weighted FermatTorricelli problem on a surface and an inverse problem. *Journal of Mathematical Analysis and Applications*, 373(1):44–58, January 2011.

[36]C Grob and T.-K. Strempel. On Generalizations of Conics and on a Generalization of the Fermat- Torricelli Problem. *The American Mathematical Monthly*, 105(8):732, October 1998.

[37]sadi Abu-saymeh and Mowaffaq Hajja. On the Fermat-TorricelIi Points of Tetrahedra and of Higher Dimensional Simplexes. *Mathematics Magazine*, 70(5):372–378, 2011.

[38]Saeed Pahlevan, Sharif Amir, Ranjbar Veeri, and Chettiar Arumugam. The Impact of higher Oil Prices on Airlines Share Price : the Case of Malaysian Airlines. In *The 10th International Conference on "U.N. Millennium Development Goals: Challenges and Perspectives"*, number 1, 2009.

[39]B Meindl and M Templ. Analysis of commercial and free and open source solvers for linear optimization problems. Technical report, University of Technology Vienna, Vienna, 2012.

[40]M R Garey and D S Johnson. *Computers and Intractability: A Guide to NP-Completeness*. Freeman, 1979.

[41]E Aarts and J Korst. *Simulated Annealing and Boltzmann Machines*. Interscience Series in Discrete Mathematics and Optimization. Wiley, 1989.

[42]L Davis. *Genetic Algorithms and Simulated Annealing*. Research Notes in Artificial Intelligence. Pitman, 1987.

[43]Adam Tauman Kalai and Santosh Vempala. Simulated Annealing for Convex Optimization. *Mathematics of Operations Research*, 31(2):253–266, 2006.

[44]Dimitris Bertsimas and John Tsitsiklis. Simulated Annealing. *Statistical Science*, 8(1):10–15, 1993.

[45]R W Eglese. Simulated annealing: A tool for operational research. *European Journal Of Operational Research*, 46(3):271–281, 1990.

[46]King-Wai Chu, Yuefan Deng, and John Reinitz. Parallel Simulated Annealing by Mixing of States. *Journal of Computational Physics*, 148(2):646–662, 1999.

[47]D Janaki Ram, T H Sreenivas, and K Ganapathy Subramaniam. Parallel Simulated Annealing Algorithms. *Journal of Parallel and Distributed Computing*, 37(2):207–212, 1996.

[48]M Miki, T Hiroyasu, and T Fushimi. Parallel simulated annealing with adaptive neighborhood determined by GA, 2003.

[49]H Chen, N S Flann, and D W Watson. Parallel genetic simulated annealing: a massively parallel SIMD algorithm. *IEEE Transactions on Parallel and Distributed Systems*, 9(2):126–136, 1998.