Dynamic optimization of codeshare flight selection for an airline company

Marc-Alexandre LaRoche[1], Michel Gamache[1], Jerome Olivier-Ouellet[2]

[1]Département de mathématiques et de génie industriel, École Polytechnique de Montréal, Montréal, Québec, Canada H3C 3A7

[2]Operations research, Air Canada Inc., Montréal, Québec, Canada H4Y 1H4

Codesharing is a widespread practice that allows an airline to put one of its flight numbers on a flight operated by a partner airline and consider it in its own network. Even though it has been repeatedly demonstrated that this type of partnership can generate important additional revenue, it is increasingly complex for an airline to choose which flights to codeshare with which partners. We propose two heuristic methods that maximize the revenue of an airline over a selection of potential codeshare flights considering two aspects that were not fully considered in the literature: the impact of adding a codeshare flight in the network and the interactions that exist between flights that are chosen as codeshare. These methods were used to optimize the codeshare flight selection of Air Canada. In comparison with two methods currently used, the developed algorithms propose codeshare flight selections that generate greater revenue.

1. Introduction

Codesharing agreements have significantly transformed the industry in the last two decades up to a point where a great majority of airlines in the world have entered into this type of partnership allowing a marketing carrier to place its flight number on a flight operated by another carrier. It has become such a widespread practice that airline alliances have been formed to facilitate the negotiations between partners. As of 2008, 55%[1] of the world airline passengers were carried by an airline member of one of the three biggest world alliances.

From an airline's point of view, codesharing is as an opportunity to expand its network without adding airplanes to its fleet. It has been shown that carriers entering codesharing agreements can reap additional revenue earned at the expense of other competing companies (Hannegan and Mulvey, 1995; Park, 1997; Oum et al., 1996). However, Hannegan & Mulvey (1995) state that these profits largely depend on the degree of integration in operations and marketing that these companies reach. Park (1997) and Oum et al. (1996) also add that these profits exist when the partnerships are

[1] This percentage is based on the world traffic data available through the Air Transport Association of America and the traffic of the airline members of Star Alliance, OneWorld et SkyTeam.
The consulted websites are:
http://www.airlines.org/Economics/DataAnalysis/Pages/AnnualResultsWorldAirlines.aspx, consulted May 13th 2010.
http://www.staralliance.com, consulted January 24th 2009.
http://www.oneworld.com, consulted January 24th 2009.
http://www.skyteam.com, consulted January 24th 2009

complementary, that is, when the partners are not competing on the same routes. From a passenger's perspective, codesharing can also have its benefits. Brueckner (2001, 2003) and Bamberger et al. (2001) highlighted cases where the presence of codesharing significantly reduced the market fares. In addition, codesharing increases the number of possible itineraries to choose from for a passenger as partners combine their flights and show a better coordination in their flight schedules.

Even if it is now clear for an airline that there is a lot to gain from codesharing, it is anything but simple to choose which markets to enter, which partners to work with and to decide exactly which flights to codeshare under which conditions. Indeed, the possibilities are virtually endless. A carrier has to carefully evaluate potential partnerships and analyze the complementarity of the respective networks, the impacts on profitability for its own network, the ease of operation integration, the impact of a new partner on existing partners and alliances, the laws applicable in the various countries and the respective union agreements.

Many models have been developed over the years to guide airlines in their codeshare flight selection. Sivakumar (2003) uses a knapsack formulation with a binary variable equal to one or zero when a potential flight is chosen to be codeshare or not. With this model, the objective function consists of incrementing the revenue obtained with the codeshare flight selection. Srinivasan (2004) presents a similar model using game theory to set the prorate ratios during negotiations between two partners. O'Neal et al. (2007) propose a procedure that also uses binary variables for codeshare decisions but, in addition, they use a model that distributes the demand on Origin-Destination itineraries instead of flights as in Sivakumar (2003) and Srinivasan (2004). This procedure is used to optimize the codeshare flight selection of Delta Airlines and the results show an increase in revenue of up to 50M$ annually. Moreover, the mathematical formulation used considers overbooking. Abdelghany et al. (2009) present a nonlinear model that considers the possible revenue loss when new customers displace non-codeshare passengers. A genetic algorithm is used to solve this problem.

All these models use a mathematical formulation that optimizes the selection of codeshare flights with one or more partners. These approaches proceed in two steps. The first step is to estimate the number of passengers that will be on each route while the second one is to calculate the added revenue of each potential flight based on the passenger distribution obtained in the first step. Hence, the number of passengers on each route is calculated only once and doesn't change in the optimization phase.

This approach neglects that the addition of a codeshare flight in a network can significantly alter the flow of passengers on itineraries in which it belongs and on other competing routes. When airline A chooses to codeshare a flight operated by company B, all routes in company A using this new flight will automatically have a greater utility. Since the demand distribution directly depends on the total utility in the market, these new shared itineraries will attract more passengers at the expense of competing routes and the demand on many routes in the network will change. Therefore, the demand on each route should not be considered as fixed during the codeshare optimizing phase.

Given that adding a codeshare flight alters the demand distribution in the network, it is likely that a codeshare flight will also have impacts on other codeshare flights. Thus, the profitability of a codeshare flight can depend on other nearby codeshare flights and should not be studied independently.

In the light of the literature review, we propose a new codeshare optimization method that considers two aspects that were not previously considered: the impact of adding a codeshare flight in the network and the interactions that exist between flights that are chosen as codeshare.

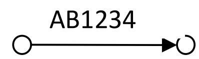
This paper describes the mathematical model built to estimate the distribution of passengers in a market and two slightly different heuristics developed to optimize codesharing. Furthermore, we present the results obtained with these heuristics when applied to Air Canada considering one and two partners.

2. General Approach

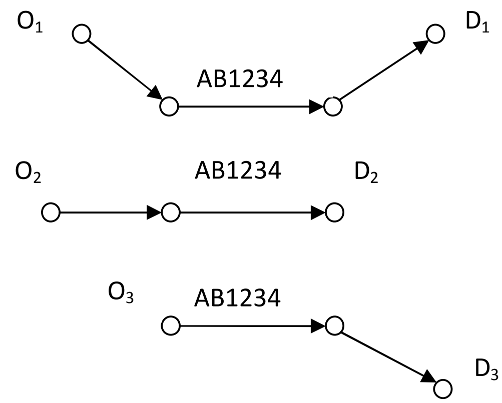
Because the demand in the network depends on the codeshare flight selection, the codeshare optimization problem cannot be solved in a single-step. Instead, the problem is modeled in two interrelated problems: one dealing with distribution of passengers and another with the maximization of revenue. Considering the large scale of this problem, we opted for a heuristic approach. At each iteration, the demand is distributed on the network considering a selection of codeshare flights. Based on the resulting network flow, this selection is modified in order to increase the profitability of the carrier.

Nowadays, it takes several minutes to estimate the demand on the complete network of a medium size airline like Air Canada. Therefore, instead of using a full network, reduced graphs built around the potential codeshare flights are used to evaluate the effect of codesharing a flight on its neighborhood. The reduced graph can be defined as all the flights on which the inclusion of a codeshare flight may have a significant effect. For example, here are the steps to build a reduced graph in the hypothetical case of flight AB1234 which would be used in only three itineraries:

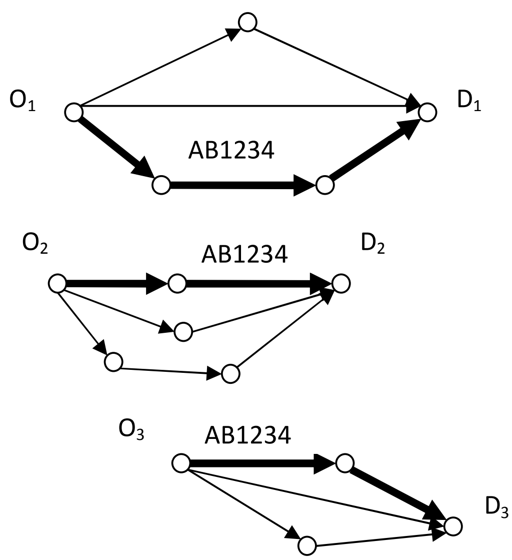
1. **Potential codeshare flight**



2. **List all the itineraries in which the potential codeshare flight is used**



3. **List all the competing itineraries of those listed at step 2**



4. **Integrate all the itineraries of step 3 in the same graph**

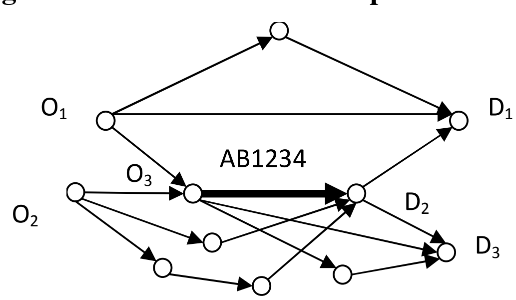


Figure 1: Reduced graph construction method

By using this construction method, we make the assumption that all changes in demand due to AB1234 becoming codeshare are contained in this reduced graph. In reality, flights in the reduced graph have influence on flights in the global network and vice versa. However, we ignore this influence as it should be negligible. Plus, this influence is partially offset by the heuristic approach proposed.

To study the impact of the inclusion of a codeshare flight on the network, two distributions of demand are made: one when the flight AB1234 is not considered codeshare and the other one when the flight AB1234 is codeshare. The indicator used is the difference in total revenue generated throughout the reduced graph for all the flights operated by the carrier. This difference in revenue, denoted Δ_p, represents the change in revenue between the first and second resolution. Following this step, it is possible to list all the potential codeshare flights and compute for each a Δ_p value. This positive or negative value estimates the benefits for an airline of codesharing each of these flights independently in their network.

The proposed heuristic constructs an initial solution by using these Δ_p values to select the flights that seem the most profitable. Considering the addition of these new codeshare flights, the demand is redistributed on the reduced graphs associated with each of the chosen codeshare flights. By recalculating the Δ_p values, the interactions between codeshare flights become clear because some flights seem more profitable and some less. At this point, the challenge is to modify the selection of codeshare flights in order to increase the overall revenue of the airline based on the Δ_p values. The proposed heuristic is explicitly described in section 4. However, before giving more details on the heuristic, Section 3 will first present the demand distribution model used within the heuristic.

3. Demand Distribution Model

The proposed heuristic requires distributing demand on several reduced graphs for each iteration. In order to do so, the model used must be reliable and easy to solve because its precision and computational time will be crucial in the performance of the heuristic.

The distribution model is based on the preferences of passengers. Each itinerary in the network is assigned a utility value that is calculated considering many factors such as time of departure, time of arrival, duration and connection type. Then, for each itinerary in this network, we calculate the market share that is equal to the utility value of the itinerary divided by the sum of utility values of all the itineraries in the network. Following that, the market share of each itinerary is multiplied by the weekly demand in that market to determine the number of passengers that should normally be assigned to this itinerary. Finally, the excess demand assigned to itineraries that have already reached their full capacity is redistributed on the network.

For the distribution of passengers, we propose a linear programming model. In that model, f, i and m designate respectively the flights, the itineraries and the origin-destination markets. The index p is used to designate a potential codeshare flight around which we construct a reduced graph. The sets F, I and M stand for all flights, itineraries and origin-destination markets in the entire network while F_p, I_p et M_p represent the subsets that define the reduced graph of p. Thus, each potential codeshare flight p is studied on its specific reduced graph built with the method described in Section 2.

The parameters are defined as follows:

D_m : Weekly demand of passengers traveling in the market m

C_f : Capacity of flight f

U_i : Utility of itinerary i

Two binary parameters are used. Let

$$S_{(i,f)} = \begin{cases} 1, & \text{if flight } f \text{ is in itinerary } i \\ 0, & \text{otherwise} \end{cases}$$

and

$$T_{(m,i)} = \begin{cases} 1, & \text{if itinerary } i \text{ is in market } m \\ 0, & \text{otherwise} \end{cases}$$

In addition, the two decision variables are:

X_i : Number of passengers assigned to each itinerary by the solver

e_i : Gap between the number of passengers assigned to each itinerary and the market share, only when the number of passengers assigned is smaller than the market share

Based on the statements above, the linear programming model is as follows:

$$Max\ Z = \left(\sum_{i \in I_p} X_i * U_i - \sum_{i \in I_p} e_i \right)$$

$s.c.$

(1) $$\sum_{i \in I_p} X_i * S_{(i,f)} \leq C_f, \qquad \forall\, f \in F_p$$

(2) $$\sum_{i \in I_p} X_i * T_{(m,i)} \leq D_m, \qquad \forall\, m \in M_p$$

(3) $$X_i + e_i \geq \sum_{m \in M_p} \left(\frac{U_i}{\sum_{j \in I_p} (T_{(m,j)} \times U_j)} \times D_m \times T_{(m,i)} \right), \qquad \forall\, i \in I_p$$

The first term of the objective function is the sum of the utility value of all itineraries multiplied by the number of passengers assigned to each itinerary. The second one represents the sum of penalties imposed when the number of passengers assigned is less than the market share. This second term is directly related to the third block of constraints that will be described below. It is not necessary here to add a coefficient greater than 1 to e_i since the coefficients of variables X_i (utility values) are in the order of decimal and e_i already is a very large number.

The first block of constraints states that a flight cannot receive more demand than its capacity. The second block states that the demand distributed in a market cannot exceed the weekly demand estimated in that market. In the third block of constraints, the following expression represents the market share of the itinerary i:

$$market\ share_i = \sum_{M_p} (\frac{U_i}{\sum_{j \in I_p} (T_{(m,j)} * U_j)} \times D_m \ \times T_{(m,i)}$$

The third block of constraint states that the solver must assign to an itinerary, when it is possible, a flow of passengers equivalent to its market share. However, if it is impossible, the solver breaks this rule and assigns a positive value to e_i. In doing so, a penalty in the second term of the objective function accounts for it. This occurs when the sum of the market shares of all itineraries using a specific flight is greater than the capacity of this flight. At that point, the solver will attempt to use the full capacity of the flight and will be penalized for the market shares that have not been assigned. The use of e_i is essential to insure that a solution always exist for the model.

This model tries to emulate the iterative spill and recapture methods that are widely used in the airline industry. In order to accelerate the resolution of the model, the third block of constraints simply redistributes the excess demand of certain itineraries to the other most attractive itineraries where seats are still available. This linear programming model gives, in most cases, results that are very close to those obtained with the spill and recapture method.

To ensure the realism of the distribution model in the heuristic, the linear program only distributes the demand of passengers traveling within the reduced graph and blocks the demand that interferes with the reduced graph but whose origin or destination is outside of the studied reduced graph.

With this model, it takes 5 to 10 seconds to distribute the demand on a typical reduced graph.

4. Heuristic Method

The heuristic proposed in this section is applied to Air Canada's network but can be adapted to any airline company.

The first step of the heuristic is to determine a list of flights, called the list of potential flights, that can be codeshare with a partner airline. Codeshare flights will be selected within this list to optimize the revenue of Air Canada. From this point, a vital component of the proposed heuristic is to determine, in each reduced graph, the itineraries that will become shared when certain flights become codeshare. To do this, a list of potential itineraries is created. This list includes all the itineraries containing at least one Air Canada flight and at least one flight on the list of potential flights. Hence, this list contains all the itineraries that can have their utility value modified when flights from the list of potential flight are selected to become codeshare. When studying a reduced graph, a potential itinerary is considered shared when all partner airline flights that compose this itinerary are selected to be codeshare. When an itinerary becomes shared, its connection type becomes more attractive to passengers because the flights composing it can be purchased from a single airline in one transaction and the connection between these flights is now guaranteed. Therefore, the total utility of the itinerary increases.

It is important to note that the flights AC* that are marketed by Air Canada but operated by a partner airline are considered as Air Canada flights in this step. In the following example, an Air Canada flight is followed by an Air Canada flight operated by company XX and then a flight of partner company ZZ.

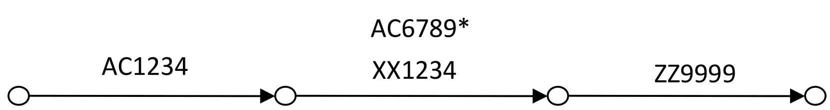


Figure 2. An itinerary AC-AC*-ZZ

In this case, the whole itinerary would become shared when selecting ZZ9999 as codeshare.

In what follows, the fictitious flight AB1234, which would belong to the potential flight list, is used to describe the procedure for assessing the effects of codesharing a certain flight while considering the flights that are already codeshare.

The first step to calculate the effect of codesharing AB1234 is to build the reduced graph around AB1234 as previously described. Then a utility value is assigned to all the itineraries in this graph considering the potential itineraries that are shared as per actual codeshare flight selection. In this first resolution, AB1234 is not considered as a codeshare flight and, consequently, all the potential itineraries containing this flight are not initially considered. At this point, the demand is distributed and the revenue of Air Canada on this reduced graph is estimated.

Secondly, AB1234 is selected as codeshare. Considering this addition in the codeshare flight selection, more potential itineraries become shared. Again, the demand is distributed and the revenue of Air Canada on this reduced graph is estimated. This revenue value therefore takes into account all the itineraries that have become shared and the effects resulting from these changes on the reduced graph.

The difference between the revenue calculated in the first and in the second resolution, denoted Δ_p, represents the effect on the revenue of codesharing AB1234 while considering the flights that are already codeshare.

An initial solution is built by calculating Δ_p for all the potential flights and selecting the flights that seem the most profitable. Considering these new codeshare flights, the demand is redistributed on the reduced graph of each chosen flight to verify if they still seem as profitable. It is possible that a codeshare flight "X" belongs to the reduced graph of another chosen flight "Y". At this point, it is possible that the flight X influences the distribution of demand in the reduced graph of Y. Ultimately, it is possible that these two flights compete for the same passengers as it is also possible that these two flights are complementary and generate even more revenue. This effect is evaluated by recalculating Δ_p for all the reduced graphs of the selected flights. Then, a number of flights that become less profitable are dropped. With the same logic, Δ_p values are calculated for flights that have not been chosen considering the actual codeshare flight selection and the ones which seem the most profitable are selected. In doing this loop over and over, the selection of codeshare flights is modified to gradually increase the revenue of Air Canada.

To calculate the revenue of Air Canada at each resolution, the demand on the itineraries containing at least one flight operated by Air Canada is multiplied by the average price of their origin-destination market. Air Canada's revenue is then the sum of the revenue on each reduced graph. For shared itineraries, Air Canada's revenue is proportional to the distance flown by a flight operated by Air Canada. For example, in a market where the average income is $ 1,200 per seat, if an itinerary is composed of an Air Canada flight that travels 400 km and then a partner flight which travels 600 km, Air Canada's revenue is estimated at $ 480. In reality, codesharing agreements can be much more complex regarding revenue sharing but we use a simplified revenue calculation.

Two versions of the heuristic are proposed. But before presenting them, the following sets and parameters must be defined:

$\mathcal{L}$ List of potential codeshare flights

$\mathcal{S}$ List of the selected codeshare flights(i.e $\mathcal{S} \subseteq \mathcal{L}$)

$\mathcal{T}_k$ List of potential flights that cannot be considered during the iterations k to k+4.

p Potential codeshare flight

$\mathcal{G}_p^E$ Reduced graph associated to flight p that considers the codeshare flights $p \epsilon E$ where E represents one of these three sets: $\emptyset$ if no flight is codeshare, $\{p\}$ when only p is codeshare or $\mathcal{S}$ when all the flights in $\mathcal{S}$ are selected as codeshare.

LP_p^E Linear program associated to $\mathcal{G}_p^E$

R_p^E Revenue associated to the demand distribution made by LP_p^E

j City

$\mathcal{N}_j$ Set of cities

$\mathcal{N}_p$ Set of cities present in reduced graph $\mathcal{G}_p^E$

Δ_p Revenue difference if flight p is considered as codeshare or not

$\Delta_{\mathcal{S}}$ Sum of the revenue differences when flights in $\mathcal{S}$ are selected as codeshare

In order to simplify the presentation of the two versions of the heuristic, we will first describe a subroutine that will be called upon to estimate the impact of codesharing flight p in the reduced graph $\mathcal{G}_{\mathrm{p}}^{\mathrm{E}}$.

Subroutine $\boldsymbol{Impact}(p, E)$:

1. Build $\mathcal{G}_p^E$
2. Resolve LP_p^E and calculate R_p^E
3. Build $\mathcal{G}_p^{E \cup \{p\}}$
4. Resolve $LP_p^{E \cup \{p\}}$ and calculate $R_p^{E \cup \{p\}}$
5. $\Delta_p = R_p^{E \cup \{p\}} - R_p^E$

The two versions of the heuristic, named algorithms 1 and 2, are described here:

<u>Algorithm 1</u>:

1. Initialization of list $\mathcal{L}$ et $k = 1$
2. $\mathcal{N} = \emptyset$
3. $\forall\, p \in \mathcal{L}$
    a. Call $\boldsymbol{Impact}(p, E = \emptyset)$
    b. $\forall\, j \in \mathcal{N}_p$
        i. If $j \in \mathcal{N}$ then go back to step 3.
    c. Si $\Delta_p > 100$
        i. $\mathcal{L} \leftarrow \mathcal{L} \setminus \{p\}$

ii. $\mathcal{S} \leftarrow \mathcal{S} \cup \{p\}$

iii. $\mathcal{N} \leftarrow \mathcal{N}_p$

4. $k = k + 1$
5. $\mathcal{S}^- = \emptyset$
6. $\forall\, p \in \mathcal{S}$
   a. Call $\boldsymbol{Impact}(p, E = \mathcal{S} \setminus \{p\})$
   b. If $\Delta_p < 0$ then $\mathcal{S}^- \leftarrow \mathcal{S}^- \cup \{p\}$
7. $\mathcal{L} \leftarrow \mathcal{L} \setminus \mathcal{S}^-$
8. $\mathcal{S} \leftarrow \mathcal{S} \setminus \mathcal{S}^-$
9. $\mathcal{T}_k = \mathcal{S}^-$
10. $w = k - 5$
11. If $w \geq 2$ then
    a. $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{T}_w$
12. $\mathcal{S}^+ = \emptyset$
13. $\mathcal{N} = \emptyset$
14. $\forall\, p \in \mathcal{L}$
    a. Call $\boldsymbol{Impact}(p, E = \mathcal{S})$
    b. $\forall\, j \in \mathcal{N}_p$
       i. If $j \in \mathcal{N}$ then go back to step 14;
    c. If $\Delta_p > 25$
       i. $\mathcal{S}^+ \leftarrow \mathcal{S}^+ \cup \{p\}$
       ii. $\mathcal{L} \leftarrow \mathcal{L} \setminus \{p\}$
       iii. $\mathcal{N} \leftarrow \mathcal{N}_p$
15. If $\mathcal{S}^+ = \emptyset$ then $\Delta_s = \sum_{p \in \mathcal{S}} \Delta_p$ and STOP; otherwise $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}^+$ and go back to step 4;

Algorithm 1 can be summarized as follows: step 1 establishes the list of potential flights that will be studied. Steps 2 and 3 select all the flights with a Δ_p greater than 100$ and creates the initial solution. Steps 4-8 drop flights that have a negative Δ_p after the flights from the initial solution have become codeshare. Steps 9, 10 and 11 manage the tabu list ensuring that flights that have been dropped are not considered for next 5 iterations. Steps 12-15 add the new flights that seem profitable. The algorithm iterates from step 4 to step 15 until a stable solution is obtained. In addition, the algorithm checks that the flights selected at each iteration have no common cities in their reduced graph. Thus, before adding flights, steps 2 and 13 establish that the set $\mathcal{N}$ is empty. Then, in steps 3-b and 14-b, if the cities in the reduced graph of p are not present in $\mathcal{N}$, flight p is selected and the cities of its reduced graph are added to $\mathcal{N}$, otherwise flight p is simply not selected.

This constraint was added to select flights that are distanced in the global network and that logically don't influence one another. By doing that, it is possible to choose more than one flight per iteration and thus accelerate the resolution without neglecting the effects that two flights might have between them.

Another particularity of this algorithm is that it uses the criterion Δ_p > 100$ for the construction of the initial solution and Δ_p > 25$ afterwards. This approach chooses flights that seem very profitable at first and analyzes the flights a little less attractive in a second step considering the flights already included in the initial solution.

We will now describe the second version of the heuristic.

<u>Algorithm 2</u> :

1. Initialization of list $\mathcal{L}$ et $k = 1$
2. $\forall\, p \in \mathcal{L}$
   a. Call $\boldsymbol{Impact}(p, E = \emptyset)$
   b. Si $\Delta_p > 100$
      i. $\mathcal{L} \leftarrow \mathcal{L} \setminus \{p\}$
      ii. $\mathcal{S} \leftarrow \mathcal{S} \cup \{p\}$
3. $k = k + 1$
4. $\mathcal{S}^- = \emptyset$
5. $\forall\, p \in \mathcal{S}$
   a. Call $\boldsymbol{Impact}(p, E = \mathcal{S} \setminus \{p\})$
   b. If $\Delta_p < 0$ then $\mathcal{S}^- \leftarrow \mathcal{S}^- \cup \{p\}$
6. $\mathcal{L} \leftarrow \mathcal{L} \setminus \mathcal{S}^-$
7. $\mathcal{S} \leftarrow \mathcal{S} \setminus \mathcal{S}^-$
8. $\mathcal{S}^+ = \emptyset$
9. $\Delta_p^* = -\infty$
10. $\forall\, p \in \mathcal{L}$
    a. Call $\boldsymbol{Impact}(p, E = \mathcal{S})$
    b. If $\Delta_p > 25$ then $\mathcal{S}^+ \leftarrow \mathcal{S}^+ \cup \{p\}$
    c. If $\Delta_p > \Delta_p^*$ then set $\Delta_p^* = \Delta_p$ and $p^1 = p$
11. If $\mathcal{S}^+ = \emptyset$ then $\Delta_s = \sum_{p \in S} \Delta_p$ and STOP; otherwise :
    a. $\mathcal{S} \leftarrow \mathcal{S} \cup \{p^1\}$
    b. $\mathcal{L} \leftarrow \mathcal{L} \setminus \{p^1\}$
12. $\Delta_s = \sum_{p \in S} \Delta_p$

13. $\mathcal{S}^{+} = \emptyset$

14. $\Delta_p^* = -\infty$

15. $\forall\, p \in \mathcal{L}$

    a. Call $\boldsymbol{Impact}(p, E = \mathcal{S})$

    b. If $\Delta_p > 25$ then $\mathcal{S}^{+} \leftarrow \mathcal{S}^{+} \cup \{p\}$

    c. If $\Delta_p > \Delta_p^*$ then set $\Delta_p^* = \Delta_p$ and $p^2 = p$

16. $\Delta_{\mathcal{S}\cup\{p^1\}} = \sum_{p\in S} \Delta_p$

17. If $\Delta_{\mathcal{S}\cup\{p^1\}} < \Delta_s$ then $\mathcal{S} \leftarrow \mathcal{S}\backslash\{p^1\}$ and go back to step 8;

18. If $\mathcal{S}^{+} = \emptyset$ then $\Delta_s = \Delta_s^2$ and STOP; otherwise :

    a. $\Delta_s = \Delta_{\mathcal{S}\cup\{p^1\}}$,

    b. $\mathcal{S} \leftarrow \mathcal{S} \cup \{p^2\}$

    c. $\mathcal{L} \leftarrow \mathcal{L} \setminus \{p^2\}$

    d. $p^1 = p^2$

    e. go back to step 13;

Algorithm 2 can be summarized as follows. Step 1 establishes the list of potential flights that will be studied. Step 2 establishes the initial solution by choosing all the flights that have a Δ_{p} greater than 100\$. Steps 3-7 drop flights that have a negative Δ_p after the flights from the initial solution have become codeshare. Steps 8-12 select the next flight p^1 which seems the most profitable. Steps 13-16 recalculate the Δ_p values and select the flight p^2 that seems the most profitable among the flights that are not selected. The step 17 checks if the flight p^1 actually increased the estimated revenue Δ_s. If this is the case, the algorithm returns to step 13 to study the flight p^2. Otherwise, the flight p^1 is definitely dropped and the algorithm tries to find another flight of interest starting at step 8. The algorithm iterates from step 13 to step 18 as long as Δ_s increases. When Δ_s decreases after adding a codeshare flight, the algorithm returns to Step 8 and continues its exploration.

Algorithm 2 is made of three main steps. First, the initial solution performs an initial screening by selecting the potential codeshare flights that seem the most profitable. Then, the first iteration makes several adjustments to take into account the interactions that exist between chosen flights and select other flights that did not seem interesting originally. Finally, starting from the second iteration, the algorithm studies one flight at a time and includes it in the codeshare flight selection only if it adds to Air Canada's revenue.

Rather than doing many iterations to find a global maximum, this algorithm aims at finding a local maximum quickly. Algorithm 2 attempts to perform few iterations that provide a good solution that is not necessarily optimal. In addition, this method prevents any form of cycling by removing the flights that do not increase the overall revenue starting from the second iteration.

5. Results

The implementation of the heuristics has been written in C++ and the demand distribution was solved through CPLEX version 11.2.1. ILOG Concert Technology version 2.7 was used to link C++ and CPLEX. The heuristics have been tested on an Intel Xeon Quad-Core 2.66 Ghz system with 3.5 GB of RAM.

Two reference algorithms are used to evaluate the performance of the new algorithms. Algorithm 3 consists in simply choosing all the potential codeshare flights and algorithm 4 chooses all the flights that present a Δ_p greater to 25$ when studied independently. These two algorithms represent two methods that are used, at different levels, in the airline industry when choosing codeshare flights.

The following two tables present the results of the four algorithms optimizing Air Canada's selection of codeshare flights with one partner.

Table 1. Optimization of the codeshare flight selection with one partner at Air Canada

	Algorithm			
	1	2	3	4
Best Δ_s (dollars/week)	158 471$	160 658$	130 573$	98 779$
Computational time (hours)	31,0	7,2	2,1	2,0
Number of flights selected	134	136	209	135

Table 2. Performance of algorithms 1 and 2 compared to algorithms 3 and 4 with one partner

	Algorithm 1	Algorithm 2
Difference in Δ_s compared to algorithm 3	+21,4%	+23,0%,
Difference in Δ_s compared to algorithm 4	+60,4%	+62,6%

These results clearly demonstrate that algorithms 1 and 2 make a more profitable codeshare flight selection than algorithm 3, which simply selects all flights connecting to Air Canada's network and is also more profitable than algorithm 4 which selects all the flights that seem profitable when studied independently.

The results are obviously very specific to the case studied. It is therefore not possible to decide whether an algorithm is better than the other in the airline industry at large. However, in this case, we can say that, when compared to algorithm 1, algorithm 2 obtains a selection of flights that is 1.4% more profitable in 7.2 hours instead of 31.0. However, algorithm 1 is more robust by allowing many more changes from one iteration to another. As explained above, algorithm 2 goes directly to a local maximum after the second iteration. This characteristic is the strong point and weak point of this algorithm because if the local maximum is very profitable, a good solution will be found very quickly. However, if the local maximum is not very profitable, the algorithm will provide a poor solution. Algorithm 1 explores in more depth the solution space and allows to go beyond a local maximum if not profitable.

On the other hand, it is interesting to note that algorithms 1 and 2 are more profitable than algorithm 3 even if they select fewer flights. This shows that there are clear interactions between selected flights. Furthermore, some Δ_p values that were positive in the reduced graph at one iteration became negative in the next iteration and vice-versa. This also confirms that the profitability of a codeshare flight can depend on the other flights already codeshare.

It is important to add that the Δ_s value does not directly represent the revenue of Air Canada if all selected flights became codeshare. This value represents the sum of the Δ_p values of the selected flights and no complete resolution of the network is made to confirm if Δ_s applies to the entire network. However, it is clear that Δ_s is directly related to the profitability of a codeshare flight selection. It is important to be aware of this nuance in the analysis of results.

With this point in mind, the annual revenue, when multiplying the revenue of this typical week by 52, is 8,354,216 $ while the revenue generated by algorithm 3 and algorithm 4 is 6,789,796 $ and 5,136,508 $, respectively. This is an increase of $ 1,564,420 from algorithm 3 and $ 3,217,708 from algorithm 4.

It is pertinent to add that even though algorithms 1, 2 and 4 select roughly the same number of flights, the selection is very different from one algorithm because their construction methods vary significantly.

To test the new heuristic in a different context, the study was extended to the case of two partners of Air Canada. Given the long computational times, only algorithms 2, 3 and 4 were performed. The results are presented in the following two tables:

Table 3. Optimization of the codeshare flight selection with two partners at Air Canada

	Algorithm		
	2	3	4
Best Δ_s (dollars/week)	306 321 $	272 318 $	163 710 $
Computational time (hours)	49,6	5,9	5,8
Number of selected flights	498	747	491

Table 4. Performance of algorithm 2 compared to algorithms 3 and 4 with two partners

	Algorithm 2
Difference in Δ_s compared to algorithm 3	+ 12,5%
Difference in Δ_s compared to algorithm 4	+ 87,1%

These results show that algorithm 2 generates much more revenue than algorithms 3 and 4 while requiring a selection of fewer flights. This confirms the results obtained when analyzing with one partner. In addition, 747 potential flights are considered in this case compared to 209 in the previous section, which demonstrates the ability of this heuristic to solve very complex cases.

However, the computational time in this case increases drastically. Algorithm 2 takes 49,6 hours to solve compared to 7,2 hours in the previous analysis with one partner. With the network considered being very large, the computational time becomes almost unrealistic in the business context. The solution space in which the heuristic seeks a solution is so vast that the only possibility is to find a local maximum as quickly as possible, which is obviously not optimal.

That being said, the annual revenue generated by algorithm 2 is 15,928,692 $ compared to 14,160,536 $ for algorithm 3 and 8,512,920 $ for algorithm 4. This represents a difference of 1,768,156 $ compared to algorithm 3 and $ 7,415,772 compared to algorithm 4. Despite the constraints of time, resources and assumptions that have been asked to simplify the problem, it remains that these results are very interesting.

6. Hypotheses list

The codeshare flight selection is a process that includes some subjectivity. Indeed, the dynamics of negotiation with partners, the political relations between countries and the behavior of passengers, among others, are all aspects that are very difficult to model accurately. To be able to construct an objective mathematical method, several hypotheses have been made.

1. All Air Canada flights and its partners' are considered fixed and cannot be modified to enhance the connectivity of certain itineraries. The codeshare flight selection is optimized considering the network as it is.

2. Air Canada optimizes the selection of the AC* flight numbers placed on flights operated by partners. The proposed method does not consider the OA* flight numbers that partners place on flights operated by Air Canada.

3. By choosing a codeshare flight, Air Canada must codeshare all the flights of the week that uses this flight number. It is not possible to codeshare a flight number for specific days of the week.

4. The revenue made from the prorate ratios when Air Canada sells seats on partner flights is not taken into account. The revenue considered is directly generated by the increase of traffic on flights operated by Air Canada.

5. The revenue made from codesharing is based on the average prices observed on each market that do not change with time.

6. The cost related to the selection of a codeshare flight is considered as null. In reality, there exists a cost related to the management of a codeshare flight. However, it is difficult to attribute a cost to a specific codeshare flight and it is legitimate to think that this cost is negligible.

7. The operating costs of each flight, which vary depending on the airplane type, are not considered. Consequently, the codeshare flight selection is optimized based on the revenue and not on the profits. Added to the fact that the operating costs are hard to evaluate, the cost per seat per kilometer flown does not seem to differ much between flights.

8. The codeshare flight selection is made on directional flights and does not imply any symmetry. In other words, Air Canada can codeshare a flight on market A→B without codesharing a flight on B→A.

9. There are no restrictions on the number of codeshare flights Air Canada can select or the number of passengers that Air Canada can add on a partner flight or on certain classes of a partner flight.
10. There are no restrictions imposed by unions or governments on the codeshare flight selection.
11. Overbooking is not considered. The demand distribution model only looks at the capacity of each airplane.

7. Conclusion

It is certainly not easy to optimize the selection of codeshare flights in the already very complex worl of the airline industry. Yet the potential profits associated with codesharing are so great that most airlines now move towards this direction. The objective of this paper is to present a tool that proposes a selection of codeshare flight maximizing the revenue for an airline considering two aspects which were not explicitly taken into account in existing models: the impact of codesharing a flight in the network and the interactions that exist between the flights chosen to become codeshare. To achieve this, a heuristic was developed to make several demand distributions on reduced graphs built around potential flights to study the codesharing effects and to modify the codeshare flight selection at each iteration.

By formulating a number of assumptions and using various methods to simplify the problem while retaining a maximum of realism, two different algorithms were proposed. To evaluate their performance, two methods currently used in the industry are simulated with algorithms to obtain reference values. The results show that the two heuristics developed in this project offer codeshare flight selections that generate more revenue than the two reference algorithms. Although several precautions must be taken in the analysis of these results, they indicate that the heuristic method developed is promising. To our knowledge, the heuristic presented is the first method optimizing the codesharing revenue that considers the interactions between the selected codeshare flights.

Despite these interesting results, the codeshare management problem is far from being resolved. This paper only examines the case of Air Canada and has several limitations. Since the available computing power was limited, the proposed heuristic does not resolve the entire network to evaluate the profitability of a codeshare flight selection. Furthermore, our mathematical model would gain precision in using a "Spill and Recapture" module. Another weakness of the proposed heuristic is its very long computation time. Research should be conducted to try to accelerate the calculation methods. Moreover, the quality of the heuristic would be greatly improved if more iterations were performed in the solution space. On the other hand, the heuristic should also consider the effects of codesharing on the revenue of partnering companies. In addition, the dynamics of negotiation should be taken into account in the codesharing optimization. Finally, codesharing should not be treated as an isolated problem but as part of the network building process to optimize the overall profitability of an airline.

References

Abdelghany A., & Sattayalekha, W. (2009). On airlines code-share optimization: a modelling framework and analysis. *International Journal of Revenue Management 3*(3) 307-330.

Bamberger, G.E., Carlton, D.W., & Neumann, L.R. (2004). An Empirical Investigation of the Competitive Effects of Domestic Airline Alliances. *Journal of Law and Economics, 47*(1) 195-222.

Brueckner, J.K. (2003). International Airfares in the Age of Alliances: The Effects of Codesharing and Antitrust Immunity. *The Review of Economics and Statistics, 85*(1) 105-118.

Brueckner, J.K. (2001). The economics of international codesharing: an analysis of airline alliances. *International Journal of Industrial Organization, 19*(10) 1475-1498.

Hannegan, T.F., & Mulvey, F.P. (1995). International airline alliances: An analysis of code-sharing's impact on airlines and consumers. *Journal of Air Transport Management, 2*(2) 131-137.

O'Neal, J.W., Jacob, M.S., Farmer, A.K., & Martin, K.G. (2007). Development of a Codeshare Flight-Profitability System at Delta Air Lines. *Interfaces, 37*(5), 436-444.

Oum, T.H., Park, J.H. & Zhang, Anming (1996). The Effects of Airline Codesharing Agreements on Firm Conduct and International Air Fares. *Journal of Transport Economics and Policy, 30*(2), 187-202.

Park, J.H. (1997). The effects of airline alliances on markets and economic welfare. *Transportation Research Part E, 33*(3) 181-195.

Sivakumar, R. (2003). Codeshare Optimizer – Maximizing Codeshare Revenues. *AGIFORS Schedule and Strategic Planning 2003.* Toulouse, France.

Srinivasan, A. (2004). Optimizing Codeshare Revenues. *AGIFORS Annual Symposium 2004.* Singapore.

Wen, Y.H., & Hsu, C.I. (2006). Interactive multiobjective programming in airline network design for international airline code-share alliance. *European Journal of Operation Research, 174*(1) 404-426.