Solving the Airline Recovery Problem By Using Ant Colony Optimization

Seyed Hessameddin Zegordi & Niloofar Jafari

Seyed Hessameddin Zegordi, Associate Professor of Industrial Engineering, Tarbiat Modares University, I.R. IRAN, Niloofar Jafari, Ph.D. of Industrial Engineering, Tarbiat Modares University, I.R. IRAN.

KEYWORDS
Airline Scheduling, Disruption Management, Aircraft Recovery, Ant Colony Optimization Algorithm

ABSTRACT
In this paper an Ant Colony (ACO) algorithm is developed to solve aircraft recovery while considering disrupted passengers as part of objective function cost. By defining the recovery scope, the solution always guarantees a return to the original aircraft schedule as soon as possible which means least changes to the initial schedule and ensures that all downstream affects of the disruption are reflected. Defining visibility function based on both current and future disruptions is one of our contributions in ACO which aims to recover current disruptions in a way that cause less consequent disruptions. Using a real data set, the computational results indicate that the ACO can be successfully used to solve the airline recovery problem.

1. Introduction
Different events ranging from severe weather to the crew sickness inhibit airlines ability to always satisfy their planning and disrupt schedules. If such disruptions are not managed suitably and timely, they will severely affect the airlines performance. Occurring disruptions, airlines correct their flight operations by delaying flight departures, canceling flights, rerouting aircraft, reassigning crews or calling in new crews, and re-accommodating passengers. The goal is to get feasible, cost-minimizing plans that allow the airline to recover from the disruptions and their associated delays [1]. This problem has been studied since last three decades. Aircraft, crew and passengers are the most important aspects of airline disruption management [2]. Nowadays, in the airline industry, recovery planning is made in a primarily sequential manner, first recovering aircraft, then crew, and finally passengers [3, cited in 4].

Aircraft Recovery problem which is considered in this paper is to decide flight re-timings, flight cancellations, and revised routings for affected aircraft. These adjustments must satisfy maintenance requirements, station departure curfew restrictions and aircraft balance requirements, particularly at the beginning and end of the recovery period. At the end of the period, aircrafts should be positioned to resume operations as planned [1]. Complete explanation of concepts and models in aircraft recovery problem can be obtained in Clausen et al. [5], Yu and Qi [6], Kohl et al. [7], Anderson and Varbrand [8], Filar et al. [3] and Clarke [9]. When the aircraft recovery problem is solved, it can result in disrupted passengers. Canceling or delaying the departure of a flight will directly affect the passengers on that particular flight. It can also, indirectly, affect the passengers on the next flight in the route for the aircraft in question, if the planned ground time between the two flights is too short to cover for the delay [8]. A limitation of most of the existing aircraft recovery models is that passengers are not modeled clearly. Jafari and Zegordi [10, 11] have developed a mathematical model for airlines schedule recovery which recovers aircrafts and disrupted passengers simultaneously. But they have not
presented a solution methodology for their model especially for large scale cases. One important quality of a recovery solution strategy and algorithm is that it has to produce a solution fast. Love et al. [12] claim that it should be generated in less than 3 min, otherwise the recovery solution can become infeasible. During the last two decades, some powerful metaheuristics have been developed and successfully applied when solving many real-life problems [13]. Recently some of the metaheuristics have been applied for recovery problems. Andersson [14] has used a Tabu Search (TS) and a Simulated Annealing (SA) approach to the flight perturbation problem. Jingjie et al. [15] have designed a Tabu Search algorithm to solve an integer programming model for airport gate reassignment. In this paper, using ant colony optimization (ACO) method, a solution for the airline recovery problem is presented.

Defining the recovery scope, it will focuses on the disrupted aircrafts and flights instead of all of them. As a result problem size will be reduced and also guarantees a return to the original aircraft schedule. To the best of our knowledge, we are the first researchers to apply ACO in the field of aircraft recovery and our results demonstrate that this approach provides good solutions within a reasonable computation time. The rest of the paper is organized as follows: Section 2 explains the airline recovery problem considering disrupted passengers. In Section 3 we describe our solution methodology to solve this problem. The computational results are presented in Section 4. In the last Section, we conclude with directions for further research.

2. Airline Recovery Problem Considering Disrupted Passengers

In operation, the schedule frequently faces with disruptions. Then the object is to turn to the initial schedule fast and with the minimum cost. Disruption or its affects can disrupt flights and their resources as well as passengers, so it is required to generate recovery plan for disrupted flights, disrupted resources and disrupted passengers [10].

In order to do this, flights can be delayed or cancelled. New aircraft assignments, i.e. swaps, can be made, both within a fleet of aircraft as well as between different aircraft types. However, a flight can only be transferred from one aircraft type to another if the new aircraft has sufficient capacity to handle the passengers on the flight. Also passengers must be arrived to their destinations. Aircraft recovery formulation is as follows [10]:

\[
\text{Subject to:} \\
\begin{align*}
x_{kf,ro} + \sum_{f'} x_{kf',ro} & \leq 1 \\
f \in \text{first flight of } ro, \quad \forall ro \in RO_s, \\
f' \in \text{first flight of } ro \in RO_q, \quad \text{when } T_f' > T_f \\
T_{lastflight}(ro) & = L_{lastflight}(ro) + T_f * \text{Maximum Delayed Allowed}
\end{align*}
\]

\[
I - z_f = \sum_{k \in K_s} s_{kf} \quad \forall f \in F_s
\]

\[
x_{kf+l} - x_{kf} = 0 \quad \forall k \in K_s, \quad f, f_i \in ro \in RO_s
\]

\[
td_f \geq A_{kf} \cdot x_{kf} \\
\forall f \in \text{first flight of rotations in } F_s, \quad k \in K_s
\]

\[
td_{f \rightarrow ro} \geq t_{f \rightarrow ro} \cdot x_{kf} \cdot s_{kf} \cdot U
\]

\[
f \in \text{first flight of } ro \in RO_s, \\
f' \in \text{last flight of } ro' \in RO_q, \quad \text{when } T_f' + DT_f' \leq T_f + \text{Maximum Delayed Allowed}, \\
\forall k \in K_s
\]

\[
td_{f \rightarrow ro} \geq t_{f \rightarrow ro} \cdot U \quad \forall f \in \text{flights of } ro \in RO_s
\]

\[
td_f \geq T_f \\
\forall f \in F_s
\]

\[
ta_f = td_f + DT_f (I - z_f) \\
\forall f \in F_s
\]

\[
(td_{IT(p,z)} / (ta_{IT(p,I)} + u)) - 1 \leq y_p \\
(td_{IT(p,z)} / (ta_{IT(p,Z)} + u)) - 1 \leq y_p \\
\forall p \in P
\]

\[
y_p \geq z_f \\
\forall p \in P, \quad \forall f \in IT(p)
\]

\[
N_p (I - y_p) = \text{itt}_p \\
\forall p \in P
\]

\[
\sum_{p \in P} \text{itt}_p \leq \sum_{p \in P} N_p (I - y_r) \\
\forall r \in R(P)
\]

\[
\sum_{p \in P} \text{itt}_p \cdot trn_p = N_p \cdot y_p \\
\forall p \in P
\]

\[
\sum_{p \in P} \sum_{r \in R(p)} (I - \delta_{pr}) \cdot \delta_{pr} \cdot \text{itt}_p \leq \text{CAP}_p (I - z_f) \\
\forall f \in R(p)
\]

\[
x_{kf} , z_f , y_p = \{0,1\}, \quad \text{and}
\]

\[
td_f, ta_f \text{ are REAL and} \\
\text{itt}_p, \text{itt}_f, trn_p \text{ are integer}
\]
The decision variables common to the model are:

\( x_{kf} \): 1 if aircraft \( k \) is assigned to flight \( f \) and 0 otherwise;
\( z_{itp} \): 1 if flight \( f \) is cancelled and 0 otherwise;
\( t_{d} \): Actual departure time of flight \( f \);
\( t_{a} \): Actual arrival time of flight \( f \);
\( y_{itp} \): 1 if planned itinerary \( p \) is disrupted, and 0 otherwise.

\( i_{p}^{r} \): Number of passengers which were initially assigned to itinerary \( p \) and served on it;
\( i_{p}^{r} \): Number of passengers which were initially assigned to itinerary \( p \) but reassigned to itinerary \( r \).

\( n_{r}^{p} \): Number of passengers which were initially assigned to itinerary \( p \) but must be served on other airlines or other transportation mode.

The parameters common to the model are:
\( A_{k} \): Ready time of aircraft \( k \) to operate flight \( f \);
\( C_{k} : \) Number of remaining available seats on flight \( f \);
\( C_{f} \): Cost of canceling flight \( f \);
\( D_{f} \): Cost of 1-minute delay of flight \( f \);
\( C_{v} \): Cost of assigning aircraft \( k \) to flight \( f \);
\( D_{v} \): Expected trip (block-to-block) time of flight \( f \);
\( F \): The set of flights in recovery scope \( S \);
\( IT(P) \): The set of flight legs in itinerary \( p \);
\( IT(P,L) \): The last flight leg in itinerary \( p \);
\( IT(P,n) \): The \( n \)th flight leg in itinerary \( p \);
\( K_{k} \): The set of aircrafts to be used for recovery in scope \( S \);
\( N \): Number of passengers on itinerary \( p \);
\( P \): Number of passengers in flight \( f \);
\( P \): Number of passengers in flight \( f \);
\( R(p) \): The set of passenger itineraries (contains more than one flights);
\( R_{p} \): The set of candidate recovery itineraries for itinerary \( p \);
\( RO_{p} \): The set of aircraft rotations in recovery scope \( S \);
\( S \): Recovery scope index;
\( S_{p} \): Estimated cost per disrupted passenger which are not reassigned;
\( T_{c} \): The scheduled departure time of flight \( f \);
\( U \): Minimum connection time;
\( V \): The duty (usage) limit of aircraft \( k \);
\( \delta_{i} \): An indicator to represent whether flight leg \( f \) is in itinerary \( r \) or not, equal 1 if flight leg \( f \) is in itinerary \( r \) and equal 0 otherwise.

In Jafari and Zegordi model [10], for each disrupted aircraft the earliest disrupted flight and its departure airport is considered. The disrupted flight and the next flights which turn back the aircraft to this airport are included in the aircraft rotation. Other than the disrupted rotations, some undisrupted rotations (rotations with no aircraft disruptions) have been included. These rotations are used to check the possibility of swapping their aircrafts with the aircrafts of disrupted rotations. The originally assigned aircrafts to all rotations in the scope are considered to be used in the recovery process. As Guo [16], the length of the recovery period is not deterministic. It denotes the horizon to recover from a disruption. With other words the recovery horizon lasts until all changes caused by the disruption have been carried out. Jafari and Zegordi [10], have done pairwise comparison for rotations of all aircrafts which can be used for the recovery, to check in the case of exchanging their rotations, how long these changes last. This method ensures that all affects of each disruption are investigated. Finishing the comparison, flights and rotations which form the recovery scope are determined. The objective function of the model [10] minimizes the total cost associated to recovering all flights, aircrafts and passengers in the recovery scope. It is include cost of aircrafts assignment, total delay, cancellation and disrupted passengers. The first term in the objective function is to recover open positions of each disrupted flight by using the most efficient aircrafts in the system. The second and third terms promote reliable operations by minimizing flight delay and cancellation, respectively. The remaining terms recover disrupted passengers through reassigning them to the earliest available itinerary or transport them to the destination by any other way, other airlines or an alternative way of transportation. Constraints in (2) assign available aircraft to sequential rotations which are not overlapped. Constraints in (3) ensure that if a flight is canceled, no aircraft is assigned to it. Rotations aircraft usage is defined in Constraints (4). Departure time of each flight is determined in Constraints (5) to (8). Constraints in (9) guarantee that in the case of delay, there is no duty limit violation for that flight’s aircraft. Also satisfaction of maintenance requirements of aircrafts can be met here. Constraints in (10) relate the departure and arrival times for each flight. Constraints in (11) and (12) classify disrupted itineraries. Constraints in (13) to (16) reassign disrupted passengers. Finally, Constraints in (17) define the decision variables.

### 3. Ant Colony Optimization Algorithm

The ACO approach dates back to the pioneering work by Dorigo et al. [17 cited in 18]. ACO randomly constructs solutions in a step-by-step fashion. An ant is a conceptual unit performing a random construction of a solution. So-called pheromone values store information on how good a single construction step was in the past, whereas so-called visibility values evaluate the presumable goodness of that step in the present context [18]. Contrary to most applications of metaheuristics such as simulated annealing (SA) or genetic algorithms (GA), the ACO approach is usually applied to highly constrained problems [18]. Also ACO is one of the heuristic computational procedures that are designed to search and find answers to complex problems in situations where the number of possible alternative solutions is vast [19]. However, the application of ACO is restricted to optimization problems that can be described by graphs [20 cited in
expresses the pheromone evaporation based on Eq. (18) when on edge \((\text{flight } i)\) selects the next node \(\text{prob. function by Eqs. (18, 19). Ant next flights and rotations nodes to visit based on a}
\text{an aircraft. They must select the flights and}
\text{ants is as aircrafts. They must select the flights and}
\text{aircrafts and the other set represents flights and}
\text{previous sections. The aircraft recovery problem is}
\text{appropriate for solving the problem described in the}
\text{21]. Thus it is expected this technique to be especially}
\text{accurate.}

\[
\tau(i, j) = \tau(i, j) + \sum_{\text{flight } k \in J(i)} \min \left(1, \frac{\eta(i, j)}{\text{capacity}}, \text{visibility}(i, j) \right)
\]

\[
\text{disrupt }(j) = \left\{ \begin{array}{ll}
\text{delay}(j) \times \text{NP}(j) \times \text{CD}, & \text{or} \\
\text{NP}(j) \times \text{CC}, & \text{Otherwise}
\end{array} \right.
\]

\[
\text{Fig. 1. Graph representing the Aircraft Recovery problem with 2 aircrafts (AC1-AC2), 5 rotations (R1-R5) and 11 flights (F1-F11) solved by the ACO.}
\]

Each ant chooses aircraft nodes randomly but chooses next flights and rotations nodes to visit based on a
\text{probability function by Eqs. (18, 19). Ant } k \text{ at node } F_i \text{ (flight } i) \text{ selects the next node } F_j \text{ (flight } j) \text{ to move based on Eq. (18) when } q \ge q_0:

\[
\frac{[\tau(i, j)]^\alpha \eta(i, j)^\beta}{\sum_{F \in J(i)} [\tau(i, F)]^\alpha \eta(i, F)^\beta}, \text{ if } j \in J_k(i)
\]

\[
P_k(i, j) = \begin{cases} 
\eta(i, j), & \text{disrupt } (j), \text{ if } j \in J(k) \\
0, & \text{Otherwise}
\end{cases}
\]

\[
\eta(i, j) = \begin{cases} 
\text{disrupt } (j), & \text{if } j \in J_k(i) \\
0, & \text{Otherwise}
\end{cases}
\]

\[
\text{Flight } j \text{ is already disrupted as it can not fly according to its initial schedule. It means flight } j \text{ will be delayed}
\text{or canceled so cost of current disruption of the flight } j \text{ at time } t \text{ (disrupt } (j) \text{) is calculated based on Eq. (21).}
\]
Where $cost_k$ is the value of the objective function for each ant $k$ and $\text{maxcost}$ is the most value of the objective function in the iteration. ACO algorithm developed for aircraft recovery is illustrated in a flowchart in Fig.2.

![Flowchart of ACO for aircraft recovery problem](image)
According to the related studies, metaheuristics solution performance or quality is sensitive to certain control parameters. ACO control parameters are as follows: iterations number, number of ants in each iteration, relative importance of density and visibility ($\alpha, \beta$), initial density ($\tau_0(i, j)$), evaporation factor ($\rho$) and importance of exploitation versus exploration ($q_0$).

To set the parameters in the proposed ACO, we ran tests with factorial designing. The tests have been done with iteration numbers of 5, 7, 10, 15 and 20, number of ants of 10, 15, 20 and 25, the relative importance of the visibility parameter ($\beta$) of 0.5, 1, 1.5 and 2 and finally with the relative importance of the density parameter ($\alpha$) of 1, 2, 3, 4 and 5. Based on the results, we found that the following parameter values gave the average best results: Iteration number = 5, number of ants = 20, the relative importance of the visibility parameter ($\beta$) = 1 and the relative importance of the density parameter ($\alpha$) = 3. Pheromone is initialized by setting $\tau_0(i, j) = 1$ for all (i, j), evaporation factor ($\rho$) is set to 0.6 and $q_0$ to 0.7.

4. Computational Results

Our ACO algorithm is tested on a real problem secondary data set obtained from Andersson [14]. The data set (S1) contains short flights, between 15 and 125 minutes and is perturbed in two different ways, a and b. In “a” an aircraft is unavailable for some hours, and in “b” some flights are imposed with delays. Some passengers’ itineraries are extracted from data set S1 to be able to solve simultaneous aircraft and passenger recovery model in this study. Table 1 sums up the important characteristics for the data set. The minimum connecting time between two flights of each aircraft and also for passengers to walk between the arrival and departure gates of the consecutive flights is 10 min. The maximum delay allowed for a single flight is set to 60 min for all flights. The computer program of the ACO algorithm described in section 3 was written in Visual Basic 2008 and ran on a Sony Vaio Intel (R) Pentium (R) M, computer with 768 Mb of memory and 1.73 GHz processors.

First the problems are solved as aircraft recovery problem with out considering connecting passengers to show the ability of ACO to suggest good solutions comparing to the results of SA and TS algorithms done by Andersson [14]. The results for the S1a and S1b are presented in Table 2.

<table>
<thead>
<tr>
<th>n</th>
<th>Weights</th>
<th>Best Known Solution-BKS</th>
<th>Objective Function Value</th>
<th>ACO Result (% from BKS)</th>
<th>SA Result (% from BKS)</th>
<th>TS Result (% from BKS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CC</td>
<td>Ct</td>
<td>C</td>
<td>CD</td>
<td>c</td>
<td>st</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>100</td>
<td>10</td>
<td>1</td>
<td>30</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>1000</td>
<td>10</td>
<td>1</td>
<td>46</td>
<td>0</td>
</tr>
<tr>
<td>S1a</td>
<td>3</td>
<td>100</td>
<td>1000</td>
<td>10</td>
<td>1</td>
<td>32</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>100</td>
<td>10</td>
<td>1</td>
<td>25</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>100</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S1b</td>
<td>2</td>
<td>20</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>100</td>
<td>400</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

CC is the weight or cost for canceling a flight, Ct the weight for assigning a flight to an aircraft of a different type, C the weight for assigning a flight to an aircraft in the same fleet as originally planned, and CD is the weight for delaying a flight. The best known solutions (BKS) have been calculated by a Lagrangian Heuristic by Andersson and Varbrand [8]. c notes how many passengers that are affected by a cancelled flight, st and s how many swaps between aircraft types and regular swaps that are made, and finally d is the total amount of passenger delay in the solution. Objective Function Value is based on aircraft recovery formulation Eq. (1). In the last three columns of Table 2, meta-heuristic solutions are compared to the best known solution (BKS) for the particular problems. Results show that ACO is able to suggest good recovery solutions for aircraft recovery problem. ACO solutions are the same as BKS’s except in S1a3 and its difference is just about 0.004%. Running ACO for simultaneous aircraft and passenger recovery model, results and the comparison
with Integer Nonlinear Planning (INLP) method [10] are presented in Tables 3 &4. Jafari and Zegordi [10] have used LINGO 8.0. Except three cases (S1a-2-2, 4-1 & S1b-1-2) solutions obtained by ACO are as well as INLP’s. ACO CPU times are less than INLP’s.

5. Conclusion

In this paper while reviewing some related studies in airline recovery, a limitation in this field which is integrated recovery is emphasized. Most of the existing aircraft recovery models do not cover disrupted passengers clearly. We have developed an Ant Colony (ACO) algorithm to solve aircraft recovery while considering disrupted passengers as part of objective function cost. A heuristic function based on current and future disruptions is defined as visibility function in our ACO algorithm. So ants will feel a stronger attraction to choose flights which are already disrupted or cause less disruptions. Therefore it prevents propagation of disruptions.

Other metaheuristics like as TS and GA have been just applied to aircraft recovery but not integrated recovery. Although computational results and comparison between these three metaheuristics in aircraft recovery and between ACO and INLP in integrated recovery indicate that our ACO algorithm is able to successfully handle airline recovery. Future studies can be done on using our ACO algorithm for other similar recovery problems. Furthermore using other metaheuristics approaches for integrated airline recovery problem can be useful.

<table>
<thead>
<tr>
<th>No</th>
<th>CC</th>
<th>Ct</th>
<th>C</th>
<th>CD</th>
<th>S_p</th>
<th>Solution Method</th>
<th>Time(s)</th>
<th>Objective Function Value</th>
<th>c</th>
<th>st</th>
<th>s</th>
<th>d</th>
<th>(t_{r_c}^{\alpha})</th>
<th>(t_{r_p}^{\alpha})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>20</td>
<td>100</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>INLP</td>
<td>83</td>
<td>1080</td>
<td>30</td>
<td>2</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1-2</td>
<td>23</td>
<td>100</td>
<td>10</td>
<td>1</td>
<td>23</td>
<td>ACO</td>
<td>20</td>
<td>1080</td>
<td>30</td>
<td>2</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2-1</td>
<td>20</td>
<td>1000</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>INLP</td>
<td>67</td>
<td>1425</td>
<td>30</td>
<td>2</td>
<td>8</td>
<td>225</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>2-2</td>
<td>23</td>
<td>1000</td>
<td>10</td>
<td>1</td>
<td>23</td>
<td>ACO</td>
<td>22</td>
<td>1425</td>
<td>30</td>
<td>2</td>
<td>8</td>
<td>225</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>3-1</td>
<td>100</td>
<td>1000</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>INLP</td>
<td>53</td>
<td>1160</td>
<td>46</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>3-2</td>
<td>23</td>
<td>1000</td>
<td>10</td>
<td>1</td>
<td>100</td>
<td>ACO</td>
<td>24</td>
<td>1160</td>
<td>46</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>4-1</td>
<td>100</td>
<td>1000</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>INLP</td>
<td>23</td>
<td>1553</td>
<td>46</td>
<td>0</td>
<td>4</td>
<td>225</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>4-2</td>
<td>20</td>
<td>1000</td>
<td>10</td>
<td>1</td>
<td>100</td>
<td>ACO</td>
<td>20</td>
<td>1573</td>
<td>46</td>
<td>0</td>
<td>6</td>
<td>225</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No</th>
<th>CC</th>
<th>Ct</th>
<th>C</th>
<th>CD</th>
<th>S_p</th>
<th>Solution Method</th>
<th>Time(s)</th>
<th>Objective Function Value</th>
<th>c</th>
<th>st</th>
<th>s</th>
<th>d</th>
<th>(t_{r_c}^{\alpha})</th>
<th>(t_{r_p}^{\alpha})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>20</td>
<td>100</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>INLP</td>
<td>71</td>
<td>270</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>1-2</td>
<td>23</td>
<td>100</td>
<td>10</td>
<td>1</td>
<td>23</td>
<td>ACO</td>
<td>23</td>
<td>270</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>2-1</td>
<td>20</td>
<td>100</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>INLP</td>
<td>32</td>
<td>460</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>225</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>2-2</td>
<td>23</td>
<td>100</td>
<td>10</td>
<td>1</td>
<td>23</td>
<td>ACO</td>
<td>20</td>
<td>480</td>
<td>0</td>
<td>0</td>
<td>14</td>
<td>225</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>3-1</td>
<td>100</td>
<td>1000</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>INLP</td>
<td>77</td>
<td>800</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>250</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>3-2</td>
<td>23</td>
<td>1000</td>
<td>10</td>
<td>1</td>
<td>100</td>
<td>ACO</td>
<td>20</td>
<td>800</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>250</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>4-1</td>
<td>100</td>
<td>1000</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>INLP</td>
<td>77</td>
<td>990</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>475</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>4-2</td>
<td>23</td>
<td>1000</td>
<td>10</td>
<td>1</td>
<td>100</td>
<td>ACO</td>
<td>21</td>
<td>990</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>475</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>5-1</td>
<td>20</td>
<td>1000</td>
<td>10</td>
<td>1</td>
<td>100</td>
<td>INLP</td>
<td>20</td>
<td>1730</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1580</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>5-2</td>
<td>23</td>
<td>1000</td>
<td>10</td>
<td>1</td>
<td>100</td>
<td>ACO</td>
<td>18</td>
<td>1730</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1580</td>
<td>0</td>
<td>15</td>
</tr>
</tbody>
</table>

Tab. 4. Results for simultaneous aircraft and passenger recovery for S1b
References


