Argumentation in the Re-accommodation of Airline Passengers using Mobile Devices

Jorge Lima[1], Ana Paula Rocha[1,2], Antonio J. M. Castro[2]

[1] DEI, Faculty of Engineering, University of Porto, Portugal,
[2] LIACC, University of Porto, Portugal

Abstract. In an airline company, one of the most important tasks is to control the operational plan, i. e., making sure that flights are executed according to the scheduled plan. When the normal functioning is affected by an unexpected event, disruption management appears in order to solve all possible issues. From aircraft to passenger, operation control centers have to effectively fix all the disrupted parts in the fastest time possible minimizing at the same time further costs. In this paper a mobile application that uses argumentation-based negotiation is introduced, allowing disrupted passengers to actively participate in the re-accommodation process, by interacting with the airline computerized system and without having to contact the airline customer service. This results in higher passenger satisfaction and less operational costs for the airlines.

Keywords: Argumentation, Multi-agent Systems, Transportation

1 Introduction

In an airline company, the planning and subsequent monitoring of flights is a complex problem because it involves the consideration of multiple and costly resources and their dependencies. Unexpected events force a change in the previously envisaged plans, making it necessary for some entity to be responsible for the resolution of possible problems that occasional irregularities might cause.

The Operational Control Center (OCC) is the entity that manages the operations of an airline company and its primary objective is to recover from flights delays, minimizing as much as possible their impact in cost. Three major consequences of these irregularities are the delays and cancellation of flights and the loss of transfers of passengers in transit to other destinations.

The resolution of an irregularity usually has three major dimensions: the aircraft and flight schedule, the passengers itinerary and the crew schedule. If one of the affected parts is the passenger, currently he is only informed by the company of an alternative route. If eventually he disagrees with the given solution he must head for the irregularities desk or call the company in order to find another solution.

A balance between the need of minimizing the costs and the importance of increasing the passenger satisfaction, is very important. In the work of Bratu [1] two models were developed that optimize the balance between airline operating costs and passenger costs by identifying flight departure times and cancellation decisions. The first model, named Disrupted Passenger Metric, minimized the sum of operating and disrupted passenger costs. The second model, named Passenger Delay Metric, the delay costs are more accurately computed by explicitly modeling passenger disruptions, recovery options, and delay costs.

Another work on the passenger recovery problem was made by Zhang and Hansen [5]. The authors introduce ground transportation modes as an alternative to the passenger recovery by air during disruptions in hub-and-spoke networks. An integer model with a nonlinear objective function allows to substitute flight legs with other forms of transportation, respecting the ground transportation times. The objectives of the model are aimed at minimizing passenger costs due to delay, cancellation or substitution, as well as minimizing the operating cost of the transportation.

The MASDIMA system (Multi-Agent System for Disruption Management) [2] aims to manage the operation of airlines, monitoring unexpected events that may affect and cause flight delays. It uses software agents where each agent represents a part of the problem including its preference and goals: aircraft, crew and passenger. Another software agent (supervisor) represents the global view of the airline company. Through an automated negotiation process, called Generic Q-Negotiation (GQN), the best integrated solution is chosen according to the global interest of the airline and complying with the time available for arriving to a solution.

These models however do not take into account the passengers personal interests. When offered a proposal the passenger might agree or disagree with it. In case of disagreement the disrupted passenger is obliged to go to the company's irregularities desk in order to see his problems solved in another way. The lack of a personalized solution might decrease the customer satisfaction and therefore the loyalty to the airline company.

As stated by Maher [4] the passenger recovery is generally considered as the final stage in the resolution process, and hence passengers experience unnecessarily large impacts resulting from the referred disruptions making this approaches far from optimal in the disrupted passenger point-of-view.

This paper proposes an approach where it is possible for an active participation of the passenger in the resolution of the problem through a mobile device, arguing with the airline system (e.g. MASDIMA), thereby trying to increase both the satisfaction and the commodity of the customer, minimizing the inherently existing drawbacks, by obtaining a more personalized solution. This approach uses argument-based negotiation [3] as a mechanism for achieving cooperation and agreement between the airline (represented by the computerized system) and the passenger.

On section 2 the passenger problem and resolution is described, including the argumentation algorithm used. On section 3 the experimentation performed using a prototype of the mobile application and results obtained are presented and, finally, in section 4 the conclusions and future work are presented.

2 The Passenger Problem and Resolution

In order to solve the passenger disruption, we propose for an active participation of the passenger in the resolution process through an argumentative process with the airline company. In face of a disruption, the airline disruption management system calculates several possible solutions for the disrupted passenger, each one with possible different costs for the airline. The airline will try to use the solution that minimizes its cost, but at the same time that maximizes the passenger satisfaction. For that, an argumentative process occurs between the two parts, airline company and passenger. Moreover, the passenger can participate in this process using a mobile device, what makes it possible to carry out it remotely and avoids the stress and waste of time of look for the irregularities desk. In order to ensure proper communication between the human passenger (hereinafter referred to simply as Passenger) and the company without the need of human supervision, a software agent representing the airline point of view (hereinafter APA - Airline Passenger Agent) was defined and implemented.

Figure 1 presents the system architecture. In this proposal, we use MAS-DIMA [2] as the airline disruption management system. It is important to mention that our proposed approach can be integrated with current OCC practices with or without a computerized system.

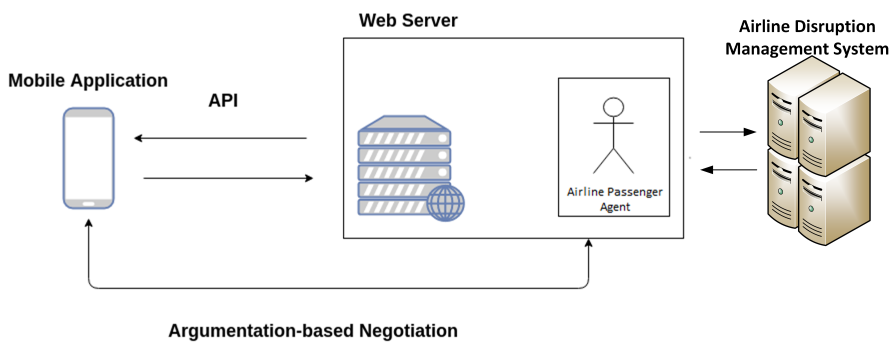

Fig. 1: System Architecture

The passenger uses a *mobile application* to interact with the system (Figure 2), that enables him to start a discussion with the company regarding its problem without the need of dislocations or any kind of effort besides some clicks on his smartphone. This application will present the passenger with a simple yet intuitive interface allowing this interaction.

To ensure proper communication channels between the passenger mobile application and the APA, a *web server* was developed. The web server wraps an agent container which contains the APA. This way it will be able to exchange messages through the exposure of an API (Application Programming Interface).

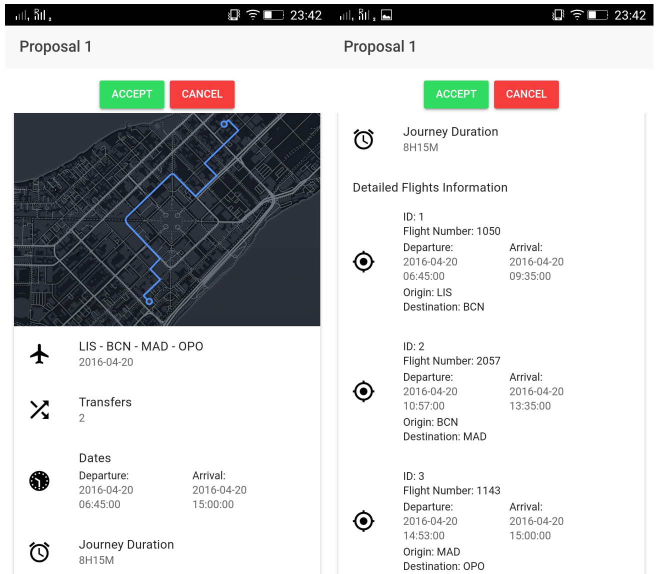

Fig. 2: Mobile Application GUI Example

After deciding which message to send, it will be converted to an API request according to its characteristics, providing an adequate communication environment.

The *APA* is the software agent that represents the airline point of view in the resolution process. It will be assigned to each disrupted passenger in order to try to solve his particular problem.

The passenger will exchange messages with APA through a *negotiation protocol based on arguments* such that the resulting solution is according to his preferences. Through the use of argumentation and based on the information the APA has according both to the passenger and the environment he is inserted, it is allowed, in addition to justify more verbally and clearly the choice of a specific proposed solution, find the closest possible solution to the needs and criteria of each affected passenger with minimal cost to the airline. APA will continuously offer new alternatives in case of disagreement with the passenger if the latter presents valid arguments. The passenger will also communicate with APA, agreeing or disagreeing with it through the use of argumentation techniques. Next subsections will describe in more detail each one of these components.

2.1 Airline Passenger Agent

The Airline Passenger Agent is a critical piece of the system since it is the one who will represent the airline company while communicating with the disrupted passenger.

Interaction with the Gateway Agent

The proposed system contains two types of software agents: a Gateway Agent and an Airline Passenger Agent. A diagram showing the interaction between these two agents is depicted in Figure 3.

The gateway agent is a simple mediator agent, that allows the communication between the airline disruption management system and the server where the APA is located. Its only function is to wait for messages from the passenger and send them to the correct APA. When the gateway agent receives a message from the server, it translates it into an ACL (Agent Communication Language) message defining each performative according to the message received, that is then sent to the specific APA. Table 1 refers to the connection between the received messages from the server and the correspondent ACL message.

Table 1: Correspondence between received messages from API and ACLMessage

From Server (Gateway Agent)	ACLMessage (Airline Passenger Agent)	
	Performative	Content
Connection Intent	INFORM	Connection message
Request Alternatives Intent	CFP	Request message
Reject Alternatives Intent	REJECT-PROPOSAL	Arguments message
Accept Alternative Intent	ACCEPT-PROPOSAL	Selected alternative
Revert Claims Intent	REQUEST	Claims to remove

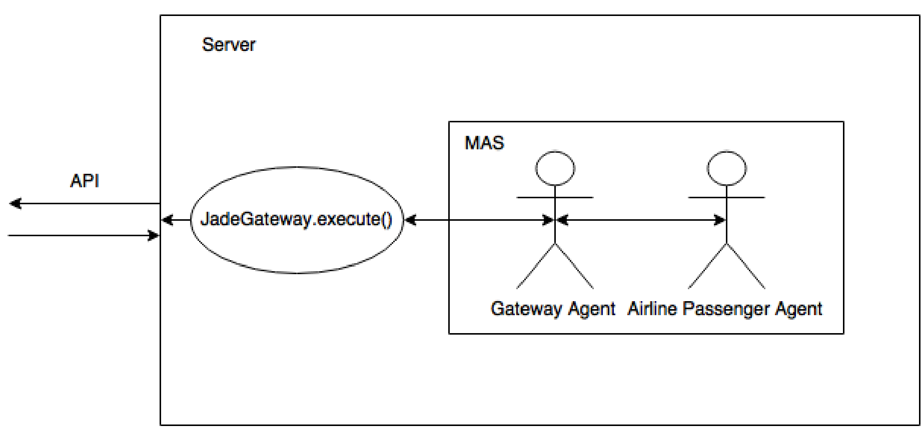

Fig. 3: Gateway Agent and Airline Passenger Agent interaction

Airline Passenger Agent Behaviours

The Airline Passenger Agent is implemented using the JADE framework, that uses a **behaviour** abstraction that allows the developer to personalize the agent to suit his needs. Taking advantage of this, the APA was equipped with four main behaviours: ConnectionBehaviour, ProposeBehaviour, ArgumentBehaviour and RevertClaimBehaviour.

The APA also makes use of a single cyclic behaviour to wait for messages from the Gateway Agent. After evaluating the received message the APA decides which of the main behaviours it should launch to achieve the desired effect. These are described in next paragraphs.

Connection Behaviour The APA will launch this behaviour when an intent of connection is sent by the passenger (*Connection Intent* in table 1). It then verifies if there is any disruption affecting that passenger and in a positive case sends him the disruption details finishing afterwards.

Propose Behaviour This behaviour is triggered when the passenger intends to start the negotiation process (*Request Alternatives Intent* in table 1). APA sends to the passenger the first four alternatives with the least cost for the airline company. No reasoning is made while obtaining these first alternatives since the passenger has not yet argued.

Argument Behaviour This behaviour is triggered when APA receives an argument from the passenger justifying his rejection of previous alternatives (*Reject Alternatives Intent* in table 1). APA analyses the passenger arguments and decides whether or not to send new alternatives. A proper algorithm was developed for that purpose, and it is described in next section.

Revert Claim Behaviour This behaviour is triggered when the passenger intends to void an already sent claim (*Revert Claims Intent* in table 1). APA sends to the passenger the best proposals that match the desired claims.

2.2 Argumentation based Resolution

As stated previously, the use of argumentation will allow a verbose justification for each proposal and counter-proposal made by each participant, either the passenger or the APA. A claim and a reason will compose an argument that will be exchanged along with the rejection of alternatives, allowing the passenger to engage on an argumentation with the APA.

It is through argumentation that the APA and the passenger will negotiate and stand by their point of view until a mutually acceptable conclusion is reached. The APA will start by presenting the passenger with four initial alternatives to his disrupted flight.

After analyzing those alternatives the passenger should decide whether or not to start an argumentation process with the APA by rejecting the presented proposals. In case of acceptance of any of the four initial proposed alternatives, the negotiation process terminates successfully. If not, an iterative argumentative process occurs between the passenger and the APA, until a mutually acceptable conclusion is reached. This last step will be detailed in the remaining of this section. It is important to point out that this process can end up unsuccessfully meaning that no agreement was achieved and the disrupted passenger will have to use the traditional methods to change the final outcome, if he so wishes. A representative diagram of the overall interaction between the passenger and the APA is presented in fig 4.

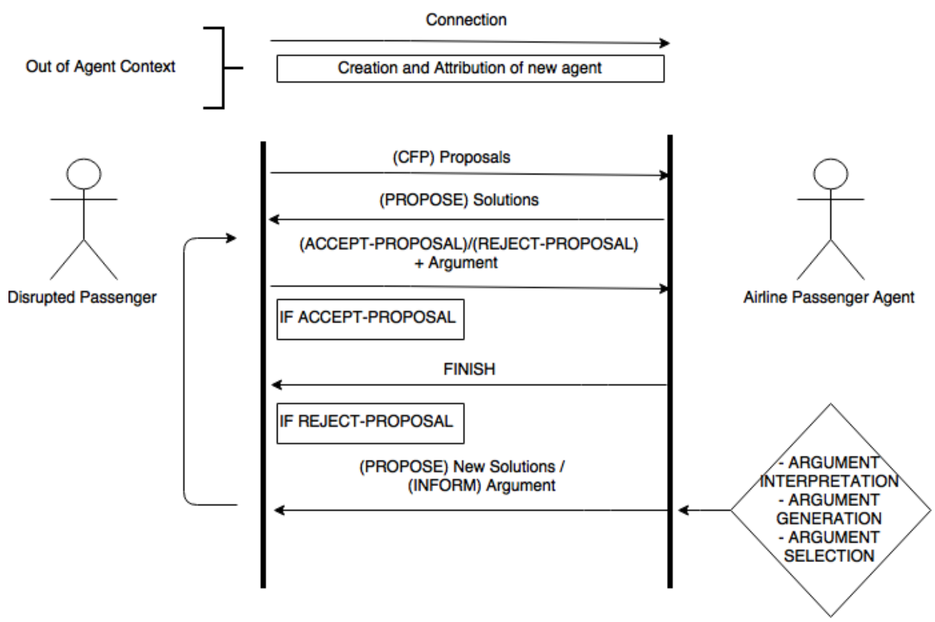

Fig. 4: Argumentation Resolution Process

Before starting the description of the argumentation process itself, the argument structure should be defined. The argument is composed by two elements, a claim and a reason, and is represented as a pair $A =< Claim, Reason >$.

In this scope, the *Claim* represents how and what the disrupted passenger wishes to improve in the previously analyzed alternatives. For instance, the time of arrival being before some date, or the number of transfers being inferior to some number. The *Reason* element represents why the passenger thinks his claims should be approved. It is this reason that should be convincing or persuading enough to change others mental state. The claims and reasons the disrupted passenger can use are limited, since natural language processing is not part of this system at the moment. The defined Claims and Reasons are the following:

– Claims: Time of Arrival, Waiting Time on Transfers, Transfer Location, Number of Transfers.
– Reasons: Personal, Professional, Health.

In this particular scenario, the disrupted passenger will argument about parameters such as those referred above, and the APA will argument on the rejection of the latter. Now that the argument was defined, it is necessary to develop an algorithm that analyzes the arguments received and decides what to do.

The argumentation reasoning by the disrupted passenger will be made solely by himself so the focus in the argumentation process will be on the APA reasoning.

When the disrupted passenger decides to start a negotiation process, a message containing his argument is sent to the server, redirecting this message to the APA assigned to that passenger. This APA will trigger an argument behaviour as referred above and start reasoning about the presented situation. This will not only contain the arguments the disrupted passenger has sent, but also knowledge the APA might have about the latter. The argumentation algorithm is described in listing Algorithm 1.

At first, the APA verifies the arguments veracity by calculating how many times the passenger has sent that argument in that context. If that value is higher than a minimum limit, it then proceeds with the negotiation, rejecting otherwise. Next the APA will check for previous alternatives that match the claims sent so far and add them, if available, to the output array (attack by rebut). Next it will verify if any argument that attacks the received one can be found, returning it if found (attack by undercut). If none of the latter steps is triggered the APA can now move on and find new alternatives that match the desired claims before sending them back to the passenger. An explanation about other methods in the Argument Algorithm follows.

Check Previously Sent Alternatives This method will verify from all the already sent alternatives all that still match the passenger requirements. The designed algorithm is presented in listing Algorithm 2.

Finding Attacking Arguments This method will verify if the APA can refute the passenger argument by attacking it. The APA will search in available data for elements that could verify the passenger argument. For instance, searching in the passengers special requests for health and/or professional related requests. The passenger importance is a relevant variable since it will define the range of acceptability of the argument as the APA will soften its aggressiveness based on that importance. The algorithm is presented in listing Algorithm 3.

Algorithm 1: Argumentation Algorithm

```
Input: Argument
Output: Response
best = null;
if argumentVeracity >= veracityLimit then
    previousAlternatives = checkPreviousAlternatives();
    if previousAlternatives not empty then
        best += previousAlternatives;
    end
    args = findAttackingArguments();
    if args is empty then
        for Round in roundsSoFar do
            currentClaims = getCurrentClaims();
            bAlternatives = findBestAlternatives(currentClaims);
            best += bAlternatives;
            Response = best;
        end
    else
        Response = args;
    end
else
    Response = error message;
end
```

Algorithm 2: Check Previously Sent Alternatives Algorithm

```
Input: Argument
Output: Response
matches = [];
if numberOfRounds > 1 then
    for Round in roundsSoFar do
        for Alternative in AlternativesSentInRound do
            for Argument in ArgumentsToClaim do
                if alternativeMatchClaim then
                    matches += true;
                else
                    matches += false;
                end
            end
            if matches not contains false then
                Response += Alternative;
            end
        end
    end
end
```

Algorithm 3: Find Attacking Arguments Algorithm

```
Input: Argument
Output: Response
paxIntel = retrievePaxHistory();
args = verifyIntel(paxIntel);
if args not null then
    Response = null;
else
    paxImportance = paxIntel.importance;
    if paxImportance greaterThan importanceLimit then
        Result = null;
    else
        Result = args;
    end
end
```

Find Best Alternatives The APA will find all the possible alternatives that match the current claims, and add them to the output array in order to send them to the passenger. The alternatives are increasingly ordered by cost for the company, so the first ones are the least cost, and so minimizing the loss of the company at each step. After finishing its reasoning, APA will whether send to the disrupted passenger new alternatives, or a message containing the rejection message and a new round will start.

Reverting Claims During the negotiation process the disrupted passenger might have the need to rollback some claim in order to see his preferences matched. In this case he could use the revert claim feature that would remove the selected claim from the claims used so far, allowing the disrupted passenger to review his selections in each round of the negotiation.

3 Experiments and Results

To validate our approach, we asked ten passengers to use the mobile application developed during our work and test three different scenarios designed based on real examples provided by an European airline. The idea was to see how many interactions (rounds) were necessary to achieve a solution that suits better the disruptive passenger interest, finishing with the acceptance (or not) of an alternative flight. The scenarios were as follows:

- **Scenario 1**: Passenger with flights AMS-LIS-OPO. Flight AMS-LIS delayed 80 minutes making the passenger to miss the LIS-OPO flight. In other two previous disruptions the passenger has used personal reasons as an argument to achieve better alternative flights.

– **Scenario 2**: Passenger with flights FRA-LIS-FNC. Flight FRA-LIS delayed 42 minutes making the passenger miss the LIS-FNC flight. During the check-in the passenger declared health related issues and in previous disruptions the passenger has argued with professional reasons.
– **Scenario 3**: Passenger with flights FCO-LIS-GIG. Flight FCO-LIS delayed 35 minutes making the passenger to miss the LIS-GIG flight. The passenger is a frequent flyer to Rio de Janeiro (GIG) and in previous disruptions has argued with professional reasons to achieve better alternative flights.

At the end, we asked them to fill a three questions form, evaluating from 1 (Very Bad) to 5 (Very Good) their satisfaction about the solution (alternative flight) and the use of the application. The questions were:

– **Q1**: How good was the final solution proposed?
– **Q2**: How do you classify the use of the application in a daily basis?
– **Q3**: How was the flow of the argumentation process?

Regarding the argument acceptance and number of rounds, figure 5 shows the results. As it is possible to see 30% of the proposed solutions were accepted at the first round, without the need for the passenger to argue for a different solution and 63,3% accepted after the second round, having the passengers used professional reason argument in 20% times, personal reasons on 26,7% and health reasons on 16,7% of the times. Only 6,3% required a third round to reach a better alternative flight. Also, the fact that in all experiments the passenger accepted the solution in, at most, the third round, reflects the good performance of the system.

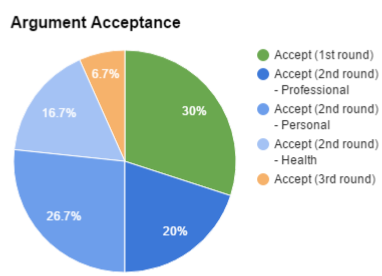

Fig. 5: Argument Acceptance vs Number of Rounds

Regarding the passenger satisfaction level, figure 6 shows the results. As it is possible to see, regarding the quality of the final solution (Q1) they classified as Very Good (80%) and Good (20%). Regarding the use of the application (Q2) most of them classified as Very Good (60%) and Good (30%) and a small percentage as Average (10%). Finally, regarding the flow of the argumentation process, 40% classified as Very Good and 60% as Good.

Although no one classified the application as Bad or Very Bad, there are improvements to be made as we state on the conclusion section.

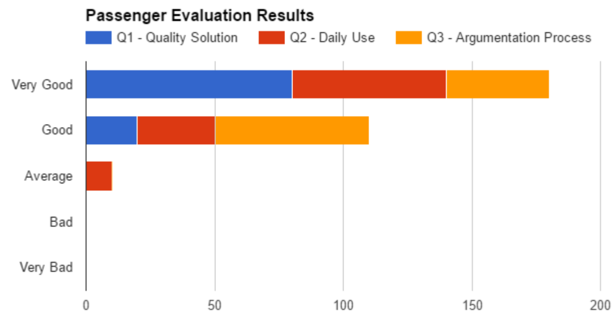

Fig. 6: Evaluation Results

4 Conclusion

The contribution of this paper is twofold. From a more scientific point of view, the proposed argumentation algorithm includes a way of find attacking arguments. Although simple, it includes interesting features such as the verification of argument's veracity, passenger importance and claims reversion.

From the air transport point of view, the application of automated argument based negotiation between a disrupted passenger and a computerized multi-agent system as well as the use of a mobile application by the disrupted passenger to participate in the passenger re-accommodation process, without having to contact the airline company passenger services, contribute to increase the passenger satisfaction and allows the airline to reduce their fixed operational costs.

Regarding future work, the argumentation process could be improved by using machine learning (namely, case based reasoning) and natural language processing, in order to open the range of possible arguments to be used. On the one hand to generate brand new arguments and on the other hand to parse any input the disrupted passenger might use.

References

1. Bratu, S., Barnhart, C.: Flight operations recovery: New Approaches Considering Passenger Recovery. Journal of Scheduling 9(3), 279-298 (2006)
2. Castro, A.J.M., Rocha, A.P., Oliveira, E.: A New Approach for Disruption Management in Airline Operations Control, Studies in Computational Intelligence, Springer, Vol. 562 (2014)
3. Kraus, S., Sycara, K., Evenchik, A.: Reaching Agreements Through Argumentation: A Logical Model and Implementation. Journal of Artificial Intelligence, (1998)
4. Maher, S.J.: A Novel Passenger Recovery Approach for the Integrated Airline Recovery Problem. Computers & Operations Research, 57, 123-137 (2015)
5. Zhang, Y., Hansen, M.: Real-Time Intermodal Substitution: Strategy for Airline Recovery from Schedule Perturbation and for Mitigation of Airport Congestion, Transportation Research Record: Journal of the Transportation Research Board, Vol 2052 (2008)