Efficient airlines planes management using multiagents and the Contract Net protocol

Dulce J. Magaña Lozano
Tecnológico de Monterrey
Centro de Computación Inteligente y Robótica.
E. Garza Sada 2501
Monterrey, México
dulce.magloz@itesm.mx

Arturo López Pineda
Tecnológico de Monterrey
Centro de Computación Inteligente y Robótica.
E. Garza Sada 2501
Monterrey, México
arturolp@itesm.mx

Ramón F. Brena Pinero
Tecnológico de Monterrey
Centro de Computación Inteligente y Robótica.
E. Garza Sada 2501
Monterrey, México
rbrena@itesm.mx

***Abstract*—Low cost airlines are growing around the world, so each time they need to manage more airplanes and take care of more passengers, as well as to keep their profitability high and their costs low. In this report is described a multiagent-based solution to the problem of managing airplanes in such airlines. An implementation of the Contract Net protocol for establishing the communication among airports and airplanes is presented, including a prototype using the NetLogo platform.**

I. Introduction

In the past few years, a number of low cost airlines have entered to the Mexican aviation market offering a real alternative to long distance travels. The low cost airlines offer point-to-point relations with very limited turn time. The basis of success for these companies lies in cheaper fares, mainly based on the cost-effectiveness.

First, they keep away from high frequented airports using rather peripheral ones with lower landing and parking fees. Second, in- flight service is reduced to the necessary basics (no frills); in most cases passengers have to pay for any additional services besides the pure air transport. Third, distribution costs are comparably low, focusing on internet and call centres, with intermediaries [2].

These airlines have grown rapidly in the Mexican market. In a period of two years they have obtained a very good market rate due to an increment in the amount of passengers. The travelers have been attracted by the lower price of these companies like Interjet, Volaris, VivaAerobus, Alma and Avolar, all newly created low cost companies in Mexican aviation industry.

The rapid growth of passengers has motivated these airlines to increase the number of flights by managing the time of each airplane more effectively and also reducing the time of their operations on each airport that they operate. These efforts also are combined with a very aggressive growing plan which includes new airplanes and more destinations.

In order to improve their profits, the companies need to increase their sales by transporting more passengers with the least effort possible and with the best security standards. This is not always easy to fulfill because of the flexibility and rapid change of data that can cause serious delays in the service provided by the airline. These delays affect in a negative way and it is a good idea to have an efficient way to manage the booking and the use of the airplanes, which is the context for this project [4].

Currently a central system plans the routes of the airplanes, then at the time of adding or removing a destination or airplane is necessary to make adjustments on the system, because most of the times these systems are based on restrictions [1]. For this reason, it is proposed a multiagent system design, in which the airplanes can communicate with the airports, obtaining with this the route planning to the different destinations that the airline handles. The objective is to decrease the costs of the company.

In this project, the method that will use for the agents is the Contract Net protocol, in which the airplanes will be the contractors and the airports the managers. An important point to consider is that the airplanes may not answer requests from the airports even if they are free, and this is because they need some kind of maintenance and therefore have to return to the airport base to be attended.

The remainder of this document is organized as follows: Section II explains the problem being analyzed by this report, including a brief description of the proposed solution. In Section III is described the overall system organization of agents, interactions between them and their protocols. In the Section IV are described the activities that the agents perform. Section V describes the simulation implemented using the NetLogo software. An explanation of the use of the Contract Net protocol is shown in Section VI, detailing the tests done with the prototype and the results obtained from it, due to observation and experimentation. Finally, section VII summarizes the work done and exhibits some conclusions about this work and it establishes some future work on this topic.

II. Problem and Multiagent Environment

For this project, the size of an airline is determined by quantity and quality of the airplane fleet, the number of destinations and the average load of passengers per year. In order to establish a basic model for the prototype, it has been considered that the airline studied is the company Interjet, which has 21 airplanes that operates 17 destinations with several frequencies. This airline is one of the most important in the Mexican low cost aviation industry. The sustained expansion of the company is reflected by the constant growth of their fleet and destination airports (from their origins, in December of 2004).

This airline, like the other low cost companies, requires moving a flow of people into a stationary and variable environment, since it changes to respect to the season of the year and date in question. It is necessary to consider that these routes not always will be covered by the same airplane, because there are various restrictions such as:

- **Time**. Here is included the time for airplane maintenance, the flight time, the time on track of each airplane, and so on.
- **Cost**. This is calculated by the time of use of each airplane and the number of passengers that the airline transports.

One of the airports (in this case, Toluca's International Airport) works as operations headquarters and maintenance base. After a certain amount of cumulative flight time, each airplane has to return to the base and stay there for time to receive maintenance.

A. The Multiagent Approach

In this section we present an overview of our multiagent approach.

If the environment of the problem is complex, large, or unpredictable, it can be addressed reasonably by developing a number of specific modular agents that are specialized at solving a particular problem aspect. The modularity let each agent to use accurate algorithms or protocols for solving each particular problem. When a given problem overpasses its boundaries and become interdependent for many agents, then those agents must coordinate themselves to ensure a properly managed operation [6].

The multiagent approach is suitable for this project since it presents a complex, highly variable and specialized environment, which requires modularity in some levels in order to perform greater tasks. It is very useful to develop applications that are more suitable for social and cognitive models like norms, organizations, beliefs and goals. These applications correspond to resources and services that are part of the multiagent environment and describe its main characteristics [3].

The agents in this project have different roles and responsibilities which will correspond to different individual goals, but in general they will collaborate to achieve a greater task, particularly for this research they will try to increase the number of passengers that the airline can manage.

III. System Organization

In the proposed system, the organization is a critical issue for the agents. In this section we will describe the organization that will follow the agents as well as the interaction between them.

The agents that exist in this prototype are the following:

Airports: The airports are self-interested agents that interact among them and with aircrafts to establish the more efficient routes in the best time for passengers.

Airplanes: There are also several airplanes that act like self-interested agents which are motivated to preserve wealthy economy by serving the most routes they can and by assuring high safety levels.

The general structure and interaction among these agents will be ruled by the Contract-net protocol. This protocol was chosen because of the following characteristics:

- All the agents are self-interested in pursuing their own objectives
- The general task is distributed among all agents as well as the information and the components.
- Contract-net negotiation gives flexibility and efficiency to route planning.
- Airplanes and airports goals are different, but not completely opposite.

A. Role Model

For each type of agent there are a number of roles that they play in order to interact with the environment, or to other agents. Roles are important for the system to determine the activities that each agent will perform but also as a way for specifying its behavior. These roles and their characteristics, for each kind of agent, are described ahead:

Airports: These agents are the managers and make requests to the contractors. They can perform two types of roles, which are explained below:

1) **Active Airport**. It is implied that all airports have this role, because they will have a certain number of passengers and traffic to other airports. In this role, airports make petitions of transports to airplanes and they are always active. It is implied that any given airport has all the facilities needed for its normal operation, and the maintenance is done while the airport is in active status. The Figure 1 shows the Schema of this role. For the sake of space economy we are not presenting the other Role schemas in detail.
2) **Hub Airport**. This role is only performed by the Toluca Airport, because it is the only one that can repair ships. It does not mean that this airport will not have also the 'active airport' role, but this role actually gives more action range to this specific airport. The Toluca Airport is set as the 'Hub airport' by the management of the airline by its strategic location, costs of operation and facility of use.

Airplanes: These agents act in the system as contractors because they serve petitions from the airports and they can perform three different types of roles:

1) **Available Ship**. This role is performed by those airplanes that are available to serve a certain flying route

Role Schema ActiveAirport		
Description	The airports perform in this role activities related to flights and the management of booking passengers	
Protocols	broadcastFlights, getPassengersInfo	
Permissions	*Reads*	tableOfDestinations, tableOfFlights, tableOfBooking, numberOfPassengers
	Change	availableFlights
Responsabilities Liveliness	ActiveAirport = (broadcastFlights . getPassengersInfo)$^\omega$	
Safety	Non applicable	

Fig. 1. Active Airport Schema

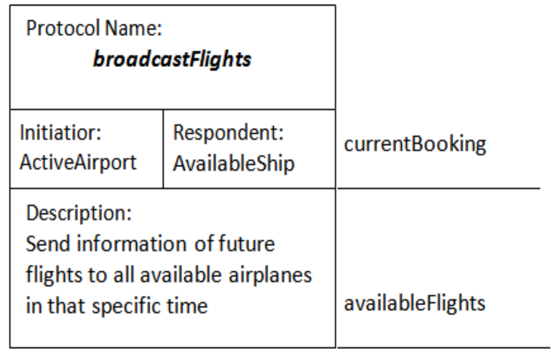

Fig. 2. Broadcast Protocol

in the time specified by the airports. Any airplane that accomplishes the requisites to be able to transport passengers with enough safety and on time will be considered as available ships; therefore they are subject to participate in the contract net procedure.

2) **InTransit Ship**. In this role the airplanes are attending a petition from an airplane, transporting passenger between destinations. This means that they are active and perhaps they can also be part in future transporting services or if they have enough flights they must go to the Hub to receive maintenance. This means that the airplane needs to get some repairs in its mechanics or just to have a general check-up in order to comply with safety and quality measurements.

B. Interaction Model

In the current model, a set of protocols for each of the roles assigned to the agents, have been established. These protocols are interdependencies and relationships between roles, and can be used as a set of institutionalized patterns of interactions which will usually give rise to more than one message at run time.

The protocols used by the roles are listed below:

- ActiveAirport: broadcastFlights
- HubAirport: broadcastFlights
- AvailableShip: makeOfferOfService, coverFlight
- InTransitShip: moveToDestination

Broadcast Flights: When an airport has already noticed that there are enough passengers booked to a certain flight, it will broadcast to all available ships the information related with this flight and wait for an answer. Then, using the contract net architecture, it will decide which airplane will cover that flight. This protocol is illustrated in Figure 2. For the sake of brevity we do not present the other protocols in the system.

Make Offer Of Service: When an airplane wants to participate in a flight covering the route and transporting the passengers, it should make an appropriate offer according to its possibilities and considering its safety levels. If it is free in the time proposed, then it could make an offer to the airport because if accepted it will increase the airplane profitability.

Cover Flight: When an airport has assigned a certain route to the airplane, then the airplane must acknowledge the task by accepting it and start the normal transporting procedure.

Move to Destination: If an airplane has a defined itinerary, including the number of passengers, the origin airport and destiny airport, then it must make the arrangements with the next airport to perform activities related to the landing.

These protocols relate with the agents roles in several interactions, where the output of one becomes the input of another. In order to clarify this fact the next Figure 3 establishes those relationships.

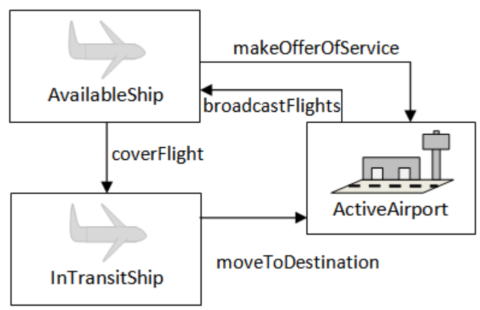

Fig. 3. Interaction Diagram

IV. Activities Description

The activities that the agents perform are described ahead:

- **getPassengersInfo**: The airports are frequently concern about getting updated passengers information. This information tells the airports how many passengers want to fly in a certain route, and it is used to determine the booking available and the next flight to be broadcasted.
- **repairShip**: The Hub Airport repairs the airplane by initializing all their variables and setting the maintenance counter to the original levels. All the airplanes must keep track of their own safety measurements.
- **endOfFlight**: When the airplane is transporting passengers between airports, and it has ended its flying operation it must acknowledge this fact by changing its current status.

- **checkSafety**: This protocol is used to determine if an airplane is suitable to perform another trip or not, due to safety issues. In normal operation the airplanes must establish proceedings to check the time that has been flying and others like aging of the fleet and so on. This means that the airplane needs to plan ahead to determine if it can make another trip or has to be in maintenance procedures.
- **goToHub**: In this protocol the next step in the airplane itinerary is for sure the Hub Airport, since it does not have another alternative for having the maintenance check up done. This protocol implies that the airplane needs to establish contact with the Hub Airport in order to get a revision.
- **activateShip**: If an airplane complies with all the safety measurements and is ready to begin normal flight operations, then it can become available for making offers and receiving the broadcast from airports.

A. Acquaintance Model

The communication links between agents is an extremely important issue when dealing with bottleneck problems. In the current system there are only two types of agents but in order to clarify the difference between the flying operations and the maintenance proceedings, the airport agent is considered as separate communication agents.

In the next diagram this communication is shown including the difference made to the airports:

Some of the possible communication problems, such as bottlenecks or deadlocks, in this system are:

- That no airplane can cover a certain flying request from an airport
- That there are two or more airplanes with the same possibility of flying to the same destination; and the airport must decide as soon as possible or the profitability of the airline will be affected.

Some possible actions for solving these problems could be the following:

- The Contract-Net architecture must provide mechanism in order to be aware of concurrent activities
- To give enough safety margins to all airplanes so that they can be aware with anticipation of the necessity for going to the Hub airport.

Within these actions, there must be also some overall cost function to determine the general profitability of the airline, and all agents must contribute within their own possibilities and capacities to achieve it.

V. Prototype

In order to develop a suitable prototype that reflects the aim of the multiagent approach, in this project was used the NetLogo environment as programming language and Integrated Development Environment.

The NetLogo environment is a multi-agent programming language and modeler for simulating natural and social phenomena. This program is very well suited for modeling complex systems evolving over time. Any given modeler can give instructions to hundreds or thousands of independent agents all operating concurrently. This particular agent-oriented program makes it possible to explore connections between micro-level behaviors of individuals and macro-level patterns that emerge from their interactions. [7]

In this project was constructed a simulation of the airline daily operation considering eight airports: Toluca (Hub), Tijuana, Cancun, Monterrey, Guadalajara, Los Cabos, Cd. Juarez and Veracruz. These airports have different routes between them and also different costs associated to the traveling distance, cost of fuel, maintenance, etc. and this costs represents a loss for each airplane. These facts are drawn in the Figure 4.

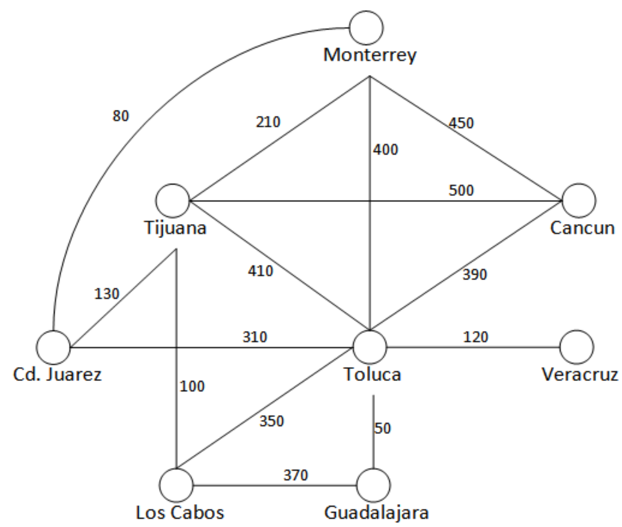

Fig. 4. Travelling costs and routes

Capabilities

Here is a list of some of the capabilities of the simulator:

- Manage the number of airplanes. This action will be carried by the user. It refers to the ability to change the number of airplanes the airline has, from two to ten airplanes.
- Manage the maximum number of passengers in the airports. This is another action that can only be carried by the user and its purpose is to generate the distribution of passengers in each airport.
- Activate Airports. This action is also carried by the user and it helps to define which airports can send requests to the airplanes, depending on demand of passengers.
- Airplane maintenance. This action will be carried out by the agents and it refers to the fact that, after certain amount of cumulative flight time, they must return to the base airport to receive maintenance. This will change the

number of available airplanes for a given moment and probably trigger a modification of the established flight routes.
- Visualization of improvement. This will show the history of monetary gains by ploting the number of passengers transported and it will serve as an evaluation metric for the performance of the system.

In the prototype we show a bar chart of the number of passengers that each airport have in any given time. And the **Model Setting** draws the map and sets the airports and airplanes in order to graphically observe the system running.

We also show the Global cost per passenger of the airline calculated as the average of the individual cost for all time units:

$$TotalCost = \sum_{i=1}^{n} \frac{Cost(Ai)}{Distance(Ai)} + \sum_{j=1}^{m} Cost(Lj) * k \quad (1)$$

Ai: all airplanes that are flying
Lj: all airplanes that are in land
k: cost for having an airplane landed
n: number of airplanes that are currently flying
m: number of airplanes that are available for flying

$$TotalPassengers = \sum_{i=1}^{n} Passengers(Ai) \quad (2)$$

Ai: all airplanes that are flying
n: number of airplanes that are currently flying

$$IndividualCost = \frac{TotalCost}{TotalPassengers} \quad (3)$$

Airports can also increase the number of passengers when an airplane lands and 20% of the incoming passengers decides to take another trip, the rest will be considered to have arrived to their final destination.

VI. THE CONTRACT NET PROTOCOL

The Contract Net Protocol is a high level protocol for communication among the nodes in a distributed problem solver. It facilitates distributed control of cooperative task execution with efficient internode communication. [5]. This protocol has several steps that agents follow in order to communicate and assign tasks. This protocol is illustrated with the current airline problem:

- *Recognition and Announcement*: The protocol begins when an airport (manager) realizes that it needs a service to be performed by airplanes (contractors). This service includes picking up passengers and sending them to a new destination. The airport needs the help of the airplanes in order to fulfill its goal, and that is the reason why it broadcast an information message to all airplanes detailing the service needed.
- *Bidding*: When an airplane receives the notification by the airports of possible flights, it must determine the possibility of participating in the service offered by the airports. For that reason, the airplane bids to the three main destinations that carries more passengers, depending on which airport it is. In the case of the Veracruz airport, it only has one possible choice, to return to the Central Airport, Toluca.
- *Awarding and Expediting*: The airports that send the task notification of service to the airplanes must choose between bids and decide who to award the contract to. The result of this process is communicated to airplanes that submitted that bid.

The awarding of the bid is restricted by the following cases:

- The airport chooses the bid with the cost. This is because it is interested in achieving its individual goal as soon as possible.
- If the airport has similar bids from different airplanes it is equally valid to choose one of them.

A. Testing the prototype

In this section we describe some results showing the performance of the prototype. These results are described given the various tests performed with the NetLogo environment.

The tests performed include the evaluation of stress conditions, changing the values of four characteristics: amount of airplanes, airports and passengers, and the running time for the particular experiment. This characteristics were combined with each other giving a total of 32 different tests, shown in the Table I for the Heuristic and in the Table II for the Contract Net Protocol:

Heuristic					
Ticks	Pssgrs.	2 airplanes		10 airplanes	
		3 airports	8 airports	3 airports	8 airports
10K	1K	11.76	12.61	7.15	39.84
	10K	9.21	3.80	7.01	5.13
100K	1K	1.98	13.73	5.93	68.24
	10K	0.44	14.89	2.10	19.69

TABLE I
AVERAGE RESULTS FOR THE HEURISTIC

Contract Net Protocol					
Ticks	Pssgrs.	2 airplanes		10 airplanes	
		3 airports	8 airports	3 airports	8 airports
10K	1K	6.72	15.12	6.04	22.19
	10K	2.82	3.92	3.29	6.39
100K	1K	6.91	69.50	5.38	35.34
	10K	3.18	14.61	3.27	12.14

TABLE II
AVERAGE RESULTS FOR THE CONTRACT NET PROTOCOL

The results can be categorized to consider the utility of applying the Contract Net Protocol or letting the normal heuristic to administer the airline operation. In most of the tests, the Contract Net protocol shows very stable results with smaller standard deviations than the heuristic.

The tests performed involve five variables:

- Method: Using Contract Net Protocol or the Heuristic.
- Number of airplanes: Considering 2 as the minimum airplanes and 10 as the maximum.
- Number of airports: For the minimum is 3 active airports which are Toluca (the Hub airport), Tijuana the airport with the higher cost from the Hub airport and Guadalajara which have the smaller cost to Toluca. And for the maximum value the eight airports are considered.
- Number of passengers by airport. Where the least number is 1 thousand and the largest is 10 thousand.
- Running time of the prototype. It is the number of ticks that the prototype was run, considering a tick as a period of time.

Using average variables of operation, we obtain the following results:

55K Ticks		
	5500 Passengers Ticks	
	CNET 8 airports 4 airplanes	HEUR 8 airports 4 airplanes
1	21.94	14.07
2	10.83	16.91
3	18.63	20.14
4	22.02	33.14
5	14.86	29.54
6	16.14	22.19
7	17.66	60.05
8	13.07	15.57
9	21.94	50.27
10	18.36	45.8
avg	17.545	30.768
stdev	3.870292122	16.19067619

TABLE III
AVERAGE NUMBERS

From the analysis of data we can draw the following observations:

The number of airplanes is one of the most sensitive topics for airlines, since the cost of maintenance and fuel increases the total cost of the company. Logically, it is easier to distribute passengers by assigning more airplanes, but since this is a very strict restriction, there must be a consideration of the model with fewer airplanes.

The cost per passenger when there are few airplanes is slightly smaller in long runs. This could be due to the fact that when there are more passengers and fewer airplanes then all the flights are fully booked; while increasing the number of airplanes to the same number of passengers will make that the airplanes transport fewer passengers individually.

The amount of airports that are active affects considerably the global cost, since more requests are made by the managers and the travelling cost for each airplane increases. The normal operation of the airline considers the eight airports but the prototype was tested with fewer in order to determine the impact of having temporal use of a given airport.

When the application deals with higher amount of passengers, the global cost diminishes. This could be caused by the higher rates of booking in individual flights which are more pruned to have better profits.

VII. Conclusion

In this report has been described the implementation of the Multiagent approach using the Contract Net Protocol for negotiation in the problem of efficient use of airplanes in a Mexican low cost airline. It also has been described the characteristics of a prototype implemented in the agent oriented environment NetLogo, which reflects a simulation of the main elements of the real world operation of the airline.

Since the problem is very complex and highly variable, the multiagent approach was an excellent technique for managing efficiently the airplanes in normal operation conditions. A formal specification was also shown as a way of describing the functionality and interaction of the prototype.

For the two agents involved in the prototype, it was recognized that they must cooperate in order to achieve individual goals and at the same time to increase the overall profitability by reducing the global cost. This negotiation was done using the Contract Net protocol and also using a simple Heuristic in which the airplanes selects the airport with more passengers.

The tests performed to the prototype included the consideration of four variables: number of airports, number of airplanes, maximum number of passengers per airport and running time of the prototype. These tests showed that the Contract Net behaves better in the long run while a reference heuristic implemented for comparison purposes was better for short terms, and in general this approach is more pruned to be very suitable for smaller problems.

Finally, the multiagent approach was very useful for dealing with an environment with multiple tasks and it actually behaves very similar to the real life operation. For communication between agents it could be used a very simple heuristic such as the higher passengers rate, but better and more stable results can be achieved with the Contract Net protocol, which is a more sophisticated mechanism for dealing with this problem.

References

[1] Michael F. Argello and Jonathan F. Bard A GRASP for aircraft routing in response to groundings and delays. *Journal of Combinatorial Optimization*, 1:211–228, 1997.

[2] Thomas Bieger and Christian Laesser. The market entry of low cost airlines (lca): Implications for mode choice between switzerland and germany. In *4th Swiss Transport Research Conference*, 2004.

[3] M. Dastani, J. Hulstijn, F. Dignum, and J. Meyer. Issues in multiagent system development.

[4] Josef J. Langerman Agent-based models for the creation and management of airline schedules. *Faculty of Science at the University of Johannesburg*, 2005.

[5] Reid G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, 1981.

[6] Katia Sycara. Multiagent systems. *AI Magazine*, 10(2):79–93, 1998.

[7] Seth Tisue and Uri Wilensky. Netlogo: A simple environment for modeling complexity. *International Conference on Complex Systems*, pages 16–21, 2004.

[8] Franco Zambonelli, Nicholas R. Jennings, and Michael Wooldridge. Developing multiagent systems: The gaia methodology. *ACM Trans. Softw. Eng. Methodol.*, 12(3):317–370, 2003.