Int. J. Complex Systems in Science
vol. **2(1)** (2012), pp. 27–32

Acceptance of reservations for a rent-a-car company

J.Alberto Conejero[1], Cristina Jordan[2,†] and
Esther Sanabria-Codesal[1]

[1] *Instituto Universitario de Matemática Pura y Aplicada, Universitat Politècnica de València, E-46022 València.*

[2] *Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València, E-46022 València.*

Abstract. Airlines schedules can been modeled by using time-space networks. One of the core problems of the management of rent-a-car companies is to automate the process of the acceptance of vehicles reservations from the clients. We propose a model to deal with this problem. Our solution is based on the admissibility of flows on these networks. The right choice of the edges of the network and which are their maximum and minimum constraints constitute the base of our work.

Keywords: Graph Theory, Ford-Fulkerson algorithm, maximization of flows, rent-a-car, scheduling of reservations
MSC 2000: 5C21, 68R10, 90B35

[†] **Corresponding author:** cjordan@mat.upv.es
Received: November 15th, 2012
Published: December 17th, 2012

1. Introduction

Scheduling problems concerning acceptance of reservations for air travel have been widely studied since travelers frequently buy flight tickets with 1 or 2 scales, see for instance [3, 4, 5]. These problems can be modeled using flows and networks, where nodes stand for airports at different hours and edges are used for representing flights or for staying at the airport waiting for a connection. Capacities of the flights and the number of reservations that are already confirmed impose the restrictions in order to accept or reject a future petition of a reservation. With such a model, the acceptance of a reservation can be decided by an application of Ford-Fulkerson algorithm.

This application of Graph Theory lets companies to accept or not on-line reservations via web in few seconds.

The structure of the aforementioned problem in air travel can be partially taken into account in order to model car reservations for a rent-a-car company that operates at a multi-city level. In this case, nodes play again the role of cities at different hours and days, and edges represent either reservations or to stay at the parking. Nevertheless, both problems have some differences: Firstly, the flow in the first one are the passengers and in the second one, the cars; and secondly, in the case of flight scheduling the flow is maximized, and in the case of rent-a-car scheduling the goal is to find an admissible flow tied to some constraints. These restrictions are minimum capacities at certain edges, that are either used to force the flow to respect reservations between pairs of nodes, or to represent the parkings that are used at every moment at each city.

In this note we give an overview of how the reservations for a rent-a-car company can be managed. In the first section, we will recall some definitions about flows and networks. Later, we will introduce time-space networks, and finally, we will see how flows can be introduced in these networks in order to deal with our problem.

2. Flows and networks

We start with a brief summary on the fundamentals of flows and networks:

Definition 1 *A **network** N is a weakly connected directed graph $G = (V, E)$, where V is the set of vertex and E is the set of edges, with two special vertex:*

- *s, called **source**, with $d_0(s) > 0$ (positive outgoing degree), and*
- *t, called **sink**, with $d_i(t) > 0$ (positive incoming degree);*

*and a non-negative function $c : E \to \mathbb{N}$, called the **capacity** of the network N. We will denote this network as $N(G, s, t, c)$.*

Definition 2 *Let $N(G, s, t, c)$ be a network. A **flow** f on N is a function $f : E \to \mathbb{N}$ such that*

- *$0 \leq f(u, v) \leq c(u, v)$ for every $(u, v) \in E$ (**the flow from** u **to** v **cannot exceed the capacity**).*
- *$\sum_{v \in V, (u,v) \in E} f(u, v) = \sum_{w \in V, (w,u) \in E} f(w, u)$, $u \neq s, t$ (for every vertex that is not a source nor a sink the **Flow Conservation Law** holds).*

We can define the value of the flow f on a network $N = (G, s, t, c)$, namely $f(N)$, either as the sum of the total flow that departs from s, or as the total flow that arrives to t:

$$f(N) = \sum_{u \in V, (s,u) \in E} f(s, u) = \sum_{u \in V, (u,t) \in E} f(u, t)$$

The flow can be maximized using Ford-Fulkerson algorithm, see for instance [1].

There are several modifications that can be considered on a network so as to adapt it in order to model different types of problems. Among others, we can also consider networks with:

- several sources and several sinks,
- maximum capacities at the vertex, and
- minimum capacities on the edges.

3. Time-space networks

Usually, vertex in networks are considered to represent physical places. Nevertheless, we can also consider that a vertex can represent a location at a certain time. Therefore, if we want to consider n locations at m different times, then we will need nm vertex to represent this situation. Such networks are called *time-space networks*, and they will be the main tool used for representing the flow of passengers and vehicles through several places along the time.

These networks are commonly used in air transportation when we want to represent flights between several cities at different hours. Here, vertex stand for airports at different hours and edges are used for three different things: to represent flights, to stay at the airport, and to connect the source (resp. the sink) with the airports at the initial (resp. final) time. In the first case, the capacity of the edge represents the number of free seats in the plane. In the second case, capacities are infinity since no limitation is required. In the third case, maximum capacities are introduced depending on the problem to be solved.

Example 3 *In Figure 1 we have a representation of 3 airports (Barcelona (BCN), Madrid (MAD), and Valencia (VLC) with 4 flights (no minimum connection time between flights is considered):*

- *Flight 1 departs from Valencia at 9:00 and arrives to Barcelona at 10:00 with 75 free seats.*

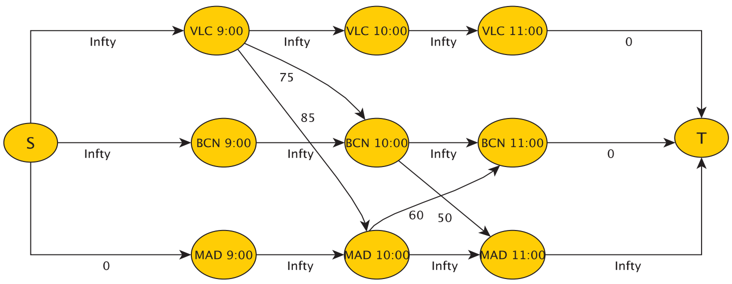

Figure 1: Example of a time-space network for representing flights connecting different airports

- *Flight 2 departs from Valencia at 9:00 and arrives to Madrid at 10:00 with 85 free seats.*
- *Flight 3 departs from Barcelona at 10:00 and arrives to Madrid at 11:00 with 50 free seats.*
- *Flight 4 departs from Madrid at 10:00 and arrives to Barcelona at 11:00 with 60 free seats.*

Suppose that we want to know how many people can travel from Barcelona or Valencia to Madrid. We suppose that the restrictions are only given by the number of free seats in the flights. Therefore, neither a restriction on how many people want to travel from Barcelona or Valencia is imposed, nor on how many people can arrive to Madrid. All these facts are introduced in the network by assigning an infinite maximum capacity to the edges (S,BCN 9:00), (S,VLC 9:00), and (MAD 11:00, T). Since we exclude the people that are at 9:00 in Madrid and the people that at 11:00 are still in Barcelona or Valencia, then we assign a maximum capacity of 0 *to the edges (S, MAD 9:00), (BCN 11:00,T), (VLC 11:00,T). So that, the solution of our problem is given by the maximum flow from S to T on this network.*

4. Acceptance of reservations

When we are dealing with air transport, if we have to decide whether to accept a reservation or not, we only have to look for augmenting paths in the network (see the proof of Ford-Fulkerson algorithm). These paths can be easily found, since no edges in reverse order can appear in any of them.

In the case of rent-a-car companies it must be taken into account that the flow represents the movement of the vehicles along the time. The acceptance of a reservation depends on having a car available at the departure destination (without leaving unattended any other previously scheduled reservation) and on having an empty parking space at the arrival destination (at the arrival time and later on). Therefore this problem can be solved by looking for an admissible flow on a network with minimum capacities at certain edges, see [2], which are related to the number of cars at every destination at the initial moment, and to the number of reservations that have been requested.

Let us see how to assign these capacities. Firtsly, suppose that we consider n dealers of a rent-a-car company at different cities. Let k_i, $1 \leq i \leq n$, be the number of cars at the dealer i at the initial time. To represent this fact, we assign the value k_i to the maximum and minimum capacities of the edge that departs from the source and arrives to the vertex for dealer i at the initial time. These capacities allow us to affirm that the k_i cars will remain at dealer i until we got a reservation departing from this dealer.

Furthermore, for every reservation of m cars between two (different) dealers at different times, we define the maximum and minimum capacities of the edge connecting both places at the corresponding times as m. If not, the flow through this edge could be smaller than m and the reservation will not be considered.

In this work, by using a simulation of petitions of reservations for a rent-a-car company and taking into account the above constraints, a computational model is proposed. This method also helps to study the real costs of one-way reservations and to determine if it is advisable for the company to increase the fleet of cars or the size of the parking at a certain dealer.

References

[1] J.L. Gross and J. Yellen, *Graph Theory and its Applications.* USA, Chapman & Hall/CRC (2006).

[2] G. Hernández-Peñalver, *Grafos. Teoría y Algoritmos.* Spain, Fundación General de la Universidad Politécnica de Madrid (2003).

[3] M. Lohatepanont and C. Barnhart, Airline schedule planning: Integrated models and algortihms for schedule design and fleet assignment, *Transportation Sci.* **38**, 19–32 (2004).

[4] D. Teodorovic, *Airline Operation Research.* New York, Gordon and Breach Science, (1988).

[5] S. Yan and C.H. Tseng, A passenger demand based model for airline flight scheduling, *Computers and Operations Research* **29**, 1559–1581 (2002).