# A Web Integration Framework for Cheap Flight Fares

Manuel Sánchez[1], Juncal Gutiérrez-Artacho[1] and Jorge Bernardino[1,3]
[1]*Superior Institute of Engineering of Coimbra, Polytechnic of Coimbra, Coimbra, Portugal*
[2]*Department of Translation and Interpreting, University of Granada, Granada, Spain*
[3]*Centre of Informatics and Systems of the University of Coimbra, University of Coimbra, Coimbra, Portugal*

Keywords: Data Integration, Web Services, Relational Database Systems, Graph Systems, Big Data, Data Mining.

Abstract: Travel agencies offer their services via the Internet, which creates new methods of communication and connection between customers and third companies. Due to the difficulty that the management of large volume of flight routes represents, it is necessary to capture the information provided by airlines through a variety of services, providing end customers with competitive fares. In this paper, we analyse the information source of flight fares offered by airlines, studying the difficulties, limitations and costs involved in accessing these data. A framework that explores the possibilities of finding "hidden" flight fares that result in much cheaper options in comparison to the average price of each flight route will be presented.

## 1 INTRODUCTION

Travel agencies have been around for decades, but throughout their history, and technological progress, they have been evolving and adapting to a new environment: the Internet. In order to make the transition to the Cloud, travel agencies have been transformed into a new kind of service provider, known as OTAs (Online Travel Agencies). This new business model, which offers its services over the Internet, has forced companies to create new methods of communication and connection between customers and third companies, through the use of Web Services.

A Web Service is a technology that uses a set of protocols and standards that are employed to exchange data between different applications on different platforms, using the Internet to transmit these data. Web Services are supplied on a uniform programming interface called an API (Application Programming Interface) (Binstock, 2015).

Online Travel Agencies provide Programming Interfaces to extend their services to third parties, and this is the key for accessing relevant information on airfares in order to enable automated management processes and the optimization of flight searches. In the field of OTAs, we focus our work on the flight management segment, and airline ticket acquisition services.

Due to the size represented by the treatment of this large volume of data (flight routes on a global scale), it is necessary to address the relevant topic of Big Data (IBM Big Data, 2015) in terms of the capture of information provided by the airlines through a variety of services and products (Guterman, 2015). In this paper, we propose a solution for processing and analyzing all obtained information, to achieve favourable patterns for future searches and thus providing customers with the best fares.

The motivation behind this work is to present and analyze business and technical tool options on the market, and develop a new feature increasingly demanded among customers. With our framework we find the best deals on airfares and with them make combinations that result in new flight routes, offering customers the possibility of travelling to places they would not have considered. As far as we know, this new feature is missing from all of these companies' services. Despite this, it is partially achievable, but in order to do so it is necessary to perform an exhaustive manual search to find a trip with a truly affordable price.

In practice, the main contributions of this work are:

• To analyse the tools provided by Online Travel Agencies (OTAs) and major flight search engines

• To offer their services through third companies, which will allow the development of new features.



Sánchez, M., Gutiérrez-Artacho, J. and Bernardino, J.
A Web Integration Framework for Cheap Flight Fares.
DOI: 10.5220/0006287702600267
In *Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017)*, pages 260-267
ISBN: 978-989-758-246-2

• To obtain the best possible fare for a particular route

• To discuss why current systems do not implement this feature automatically.

The remainder of this paper is organized as follows. Section 2 gives an introduction to the current systems available on the market, what are the best options, when thinking about using these services and what they do not provide to their customers. Section 3 analyses two of the most widespread classes of Database Management Systems (DBMS) on the market, their functions and features, and the choice of a tool according to the nature of the problem. Section 4 explains the problems encountered in establishing a single information source for flight fares. Section 5 describes our proposed web integration framework explaining the architecture of the system developed and giving an example of the results of flight fares using our solution. Finally, in Section 6 we present the conclusions and indicate future work.

## 2 BACKGROUND

Nowadays there are services that OTAs increasingly use to grasp the attention of potential customers, which include attractive features that enable greater accessibility and dynamism. Many of these companies are well known for offering the best deals on airfares, others for having flexible tools to specify a date for travelling, and others also offer the possibility of choosing multiple destinations.

The main difficulty that these OTAs have when offering possible flight combinations are the restrictions applied to tariffs by the airlines companies. These difficulties differ from temporary restrictions on advance ticket purchase, up to combination restrictions between different flights, which can become quite complex, making too difficult processing the information for determining all possible combinations. An example of a temporary restriction could be the application of a particular fare that can only be carried out with the purchase of a ticket 14 days before departure (Valles & VanLoy, 1991).

We can see an example of combinations of airfare restrictions when attempting to combine this fare with another provided by a different company or airline alliance. This results in an incompatible approach to applying these prices under a unified buying process, because the offers are only valid when combined with other flights from the same company.

Due to the complexity of the restrictions on the airfares, the implementation of an upper layer of analysis of these results is necessary, which allows us to achieve optimal results searching fares independently, without infringing the restrictions applied to each one.

Some examples of implemented systems that provide similar services with fare graphic visualizations are companies like Vayant, Skypicker and Kayak. These companies have systems that, starting from a predetermined origin, find flight routes with the existing lowest price across all dates, so that if we choose a point of origin they can find the lowest fare flight for each of the possible destinations.

Other OTAs such as Skyscanner offer total flexibility for choosing travel dates, being able to search a range of weeks, months, or even the full year, thus obtaining the best possible price for a given route. Figure 1 shows an example of the Skyscanner application.

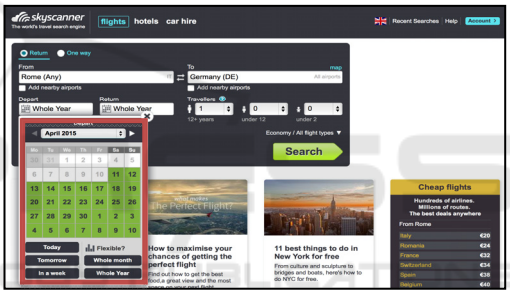


Figure 1: Skyscanner application.

Skyscanner is a complete platform where we can search with flexible dates and general locations such as a city which may have several airports, or an entire country, although it is not possible to suggest the best deals to the client on routes not established by the user, and they cannot make combinations that result in new flight routes. A real example of this can be observed in Figure 2, where we found a flight from Cancun (CUN, Cancun, Mexico) to Brussels (BRU, Brussels, Belgium) for €80, which is an excellent deal, much cheaper than the next lowest price, €1,000. If we had not made a search between Mexico and Belgium we would never have found this offer, because Skyscanner does not display these suggestions automatically.

## 3 DATABASE MANAGEMENT TOOLS

In this section we analyse and compare the most

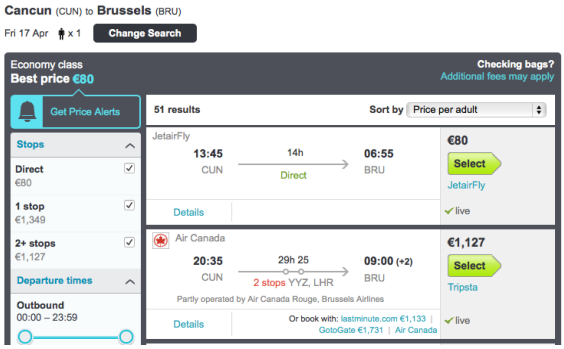


Figure 2: Great fare from Cancun to Brussels.

relevant information storage systems in the market: Relational Databases and Graph Databases. These systems are tested considering two metrics: data storage size and performance using typical queries. We also studied the services and facilities offered by each of these databases systems. At the end of the section we discuss the nature of the problem, i.e., possible combinations of routes based on nodes and relationships to get from an origin point to a given destination point.

## 3.1 Relational Databases

Relational Databases have been the standard system for storing and accessing information used by most systems since the early '80s, and even today. Most current business models, due to their characteristics, internal processes and restrictions, are naturally adapted for this architecture based on ACID (Atomicity, Consistency, Isolation and Durability) properties, helping to understanding and integrate traditional models.

An important goal of any database system is to model the real world accurately in a consistent manner with the user's perception of the data. Unfortunately most traditional DBMS do not provide adequate integrity features to ensure the accuracy of data in their databases. Correspondingly, most of traditional DBMS do not provide the necessary optimally modelling tools. They present difficulties because these databases are more oriented to define the characteristics that identify the relationships of the problems that determine how they relate to each other.

The advantages provided by a relational DBMS (RDBMS) are based on the capacity for integration and standardization of the language and architecture, serving a homogeneous model on mathematical rules. We also noticed the disk storage a requirement of information, when compared to other DBMS is an important feature to consider. In Table 1 we show the experimental results of the volume of data occupied in disk with structures of random integers, and strings with 8KB, and 32Kb. For these experiments we choose MySQL as a tool for Relational Databases, and Neo4j for Graph Database.

Table 1: Size comparison of different data types.

| Database | #Nodes | Data Type | MySQL Size | Neo4j Size |
|---|---|---|---|---|
| 1000int | 1000 | Int | 0.232M | 0.428M |
| 5000int | 5000 | Int | 0.828M | 1.7M |
| 10000int | 10000 | Int | 1.6M | 3.2M |
| 100000int | 100000 | Int | 15M | 31M |
| 1000char8k | 1000 | 8K Char | 18M | 33M |
| 5000char8k | 5000 | 8K Char | 87M | 146M |
| 10000char8k | 10000 | 8K Char | 173M | 292M |
| 100000char8k | 100000 | 8K Char | 1700M | 2900M |
| 1000char32k | 1000 | 32K Char | 70M | 85M |
| 5000char32k | 5000 | 32K Char | 504M | 406M |
| 10000char32k | 10000 | 32K Char | 778M | 810M |
| 100000char32k | 100000 | 32K Char | 6200M | 7900M |

As we can see from Table 1, the results of storing data into Neo4j database takes between 1.25 to 2 times the size of the MySQL database. One disadvantage of using relational databases to tackle this issue is at the same time its best feature, strong constraints of the model tables/relationships and structured language (SQL) that needs to be applied. Therefore discards using an architecture of this type, opting for the use of other NoSQL models, namely Graph Systems (Kamel, 1994). Systems based in NoSQL allow dynamic structures, and are faster to process the information, a key to dealing with massive data volume as will see in the next section.

## 3.2 Graph Databases

Graph Databases are based on NoSQL model that have gained recognition in recent years due to new features provided when modelling a database. The term “NoSQL”, as a term for modern web data

stores, first began to gain popularity in early 2009. It is a topic that has gained credit from the IT community but has yet to garner large-scale academic study.

The advantages provided by these systems are evident; the freedom to organize the structures so that they are not restrictive, and the speed to process information stored (higher than RDBMS when larger is the amount of information to be processed in a query). Table 2 shows the queries to simulate some of the types of queries used in our provenance systems. For example, traversals are necessary to determine data objects (nodes) derived from or affected by some starting object or node:

- Q0: Find all orphan nodes. That is, find all nodes in the graph that are singletons, with no incoming edges and no outgoing edges.
- Q4: Traverse the graph to a depth of 4 and count the number of nodes reachable.
- Q128: Traverse the graph to a depth of 128 and count the number of nodes reachable.

Table 2: Structural query results (in milliseconds).

| Database | MySQL Q0 | Neo4j Q0 | MySQL Q4 | Neo4j Q4 | MySQL Q128 | Neo4j Q128 |
|---|---|---|---|---|---|---|
| 1000int | 1.5 | 9.6 | 38.9 | 2.8 | 80.4 | 15.5 |
| 5000int | 7.4 | 10.6 | 14.3 | 1.4 | 97.3 | 30.5 |
| 10000int | 14.8 | 23.5 | 10.5 | 0.5 | 75.5 | 12.5 |
| 100000int | 187.1 | 161.8 | 6.8 | 2.4 | 69.8 | 18.0 |
| 1000char8K | 1.1 | 1.1 | 1.1 | 0.1 | 21.4 | 1.3 |
| 5000char8K | 7.6 | 7.5 | 1.0 | 0.1 | 34.8 | 1.9 |
| 10000char8K | 14.9 | 14.6 | 1.1 | 0.6 | 37.4 | 4.3 |
| 100000char8K | 187.1 | 146.8 | 1.1 | 6.5 | 40.9 | 13.5 |
| 1000char32K | 1.3 | 1.0 | 1.0 | 0.1 | 12.5 | 0.5 |
| 5000char32K | 7.6 | 7.5 | 2.1 | 0.5 | 29.0 | 1.6 |
| 10000char32K | 15.1 | 15.5 | 1.1 | 0.8 | 38.1 | 2.5 |
| 100000char32K | 183.4 | 170.0 | 6.8 | 4.4 | 39.8 | 8.1 |

For the traversal queries, Q0, Q4, and Q128, Neo4j was clearly faster, sometimes by a factor of 10, as detailed in Table 2. This was expected since relational databases are not designed to do traversals.

We conclude that the Graph Database systems are the most optimal for searching existing connection between nodes, providing response times much lower than those obtained using Relational Databases, which is the critical problem of flight management systems.

## 3.3 Graph Database Tool

Based on previous experiments, we have chosen the Graph DBMS architecture. In this section, we discuss which is the most suitable tool to implement a system of this type, which contains several requirements when implementing this solution. These requirements are the speed of data processing, heterogeneity of access to information, migration of the platform, or a possible scalability of the system. Among all the possibilities, and based on a study of efficiency of Graph DBMS platforms, we have determined that the most efficient tool could be Neo4j (Woodie, 2015). It is a Graph DBMS that we can get in free community versions, or Enterprise, which involves an economic outlay, taking advantage of technical assistance, which can be of great help in certain occasions.

After choosing the best Graph DBMS tool used to develop the application, it is necessary to analyze in which platform will deploy it. When choosing where to host the database, there are several possibilities: hosting on AWS / EC2, Windows Azure, or Cloud Hosting providers like GrapheneDB, GraphStory, Structr, etc. Thinking about the flexibility of the system, the best option may be to acquire a server in a hosting. This option is based in being able to deploy databases, implement Data Mining system, and develop diverse features under the same hosting.

## 3.4 Problem Definition

When thinking about how to approach the solution to the problem of connections between different flights, we have the idea of a large interconnected network in which all points will not be interconnected, but we have a variety of possibilities to arrive at the desired point from a given origin. By this way we can model the problem, and the best option is using graphs, because the essential information resides in the interaction between connections, naturally expressed by graphs DBMS.

An example of the nature of the problem is shown in Figure 5. In order to determine the possibilities of reaching from Paris-Orly Airport (ORY, Paris, France) to the International Brussels Airport (BRU, Brussels, Belgium), we have not defined a direct

route, but if we can determine different possibilities to reach final destination. The proposed objective is to reach any destination at the lowest possible cost.

In this case, the intermediate routes that could be selected to reach the desired destination could be through the connection at Barcelona airport (BCN, Barcelona, Spain), Barajas airport (MAD, Madrid, Spain) and Porto airport (OPO, Porto, Portugal). The result of final route would be determined by the minimum possible path and the lowest price found among all these possibilities.

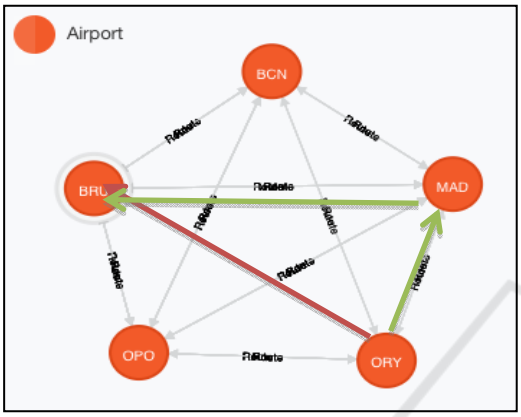


Figure 3: Routes between airports example.

## 4 INFORMATION ACCESS POINT PROBLEMS

The main way to access airlines flight fares are through GDS (Global Distribution System). GDS is a network operated by a company that enables automated transactions between third parties and booking agents in order to provide travel-related services to end consumers. A GDS can link services, rates and bookings, consolidating products and services across all three-travel sectors (book flight, book hotel, car hire).

We can find two variants of the GDS systems (Jonas, 2013); one of them is GNE (Global Distribution System New Entrant). It does not include incentives in transactions like regular GDSs, therefore the profit margin for travel agents is less, and will offer a more affordable price for the purchase of flights.

The LCC (Low Cost Carrier) is another variant that includes the largest listing group flights of low cost companies and also offering their offers on their own websites. Therefore, it is also a good option for our use.

A GDS system is composed of several modules management programs called CRS (Computer Reservation System), which is a computerized system used to store and retrieve information and conduct transactions related to air travel, hotels, car rental, or activities. Originally designed and operated by airlines, CRSs were later extended for the use of travel agencies (AXSES, 2013).

The main advantage of using these systems is the centralization of the flights of different airlines, which provides the same source information point, and with the same interface for access to data (Hospitality Net, 2015).

The major disadvantage of the centralization of these data is the free use of them, which becomes rather difficult to handle for a single user or a small business, due to the access and use of these systems includes a payment. The payment is normally under a SaaS license, whose cost is determined by the amount of transactions that are made to the system (Consumer Reports WEB Watch, 2015).

The GDS systems are not the source of the information of the flights from airlines. On the contrary, there are two main agencies that saved and updated every hour, virtually in real-time information provided by the airlines. One of them is ATPCO (Airline Tariff Publishing Company), and the other is SITA (Société Internationale de Télécommunications Aéronautiques).

ATPCO is a corporation that publishes the latest airfares for more than 500 airlines, multiple times per day, and provides fare data in an electronic format, which make the information suitable for computer processing. The only competitor to ATPCO is SITA, who distributes some fares only in Asia, Africa and Europe.

Fares are distributed hourly each day and airlines carefully monitor new public fares filed by their competition for publication through their systems. Once time these corporations have distributed the fares, airlines detect the action of other airlines increasing or decreasing their fares for specific connections, and then use this information to set their own pricing strategy. For instance, if they see a competitor introducing special promotional pricing between two cities, they may want to quickly react by filing their own special fares.

Access to the data provided by these major corporations is very expensive and imposes several restrictions, such as having an accredited title of travel agent, which is obtained by performing a test, and paid substantial fees. Due to these difficulties, the best option would be to access the information by consuming Web services that provide companies more focused to the end customer, the OTAs (Online Travel Agencies).

# 5 WEB INTEGRATION FRAMEWORK

In this section we explain our web integration framework, describing its architecture and also giving a working example of our proposal.

## 5.1 Proposed Architecture

The modelling of the solution has been made from the information source that provides Skyscanner through their offered API. The information processing consist in four fundamental steps, and assume a starting point in the system where the database has stored the list of existing airports and all the routes that connect it directly (see Figure 6).

To dispose the system at this early stage, it is necessary to extract the relevant information from airports and routes from another sources, in this case it will be FlightStats and OpenFlights. We can access the list of active civil airports through these two organizations, and active routes between each of them through OpenFlights. To obtain the information through OpenFlights is only necessary to download the appropriate files.

If we make requests to FlightStats, the response will be in JSON format, and we must transform and store the data in our database. When storing airports, we get as a result, nodes that will serve as a connection between possible routes in our system.

Once we have in the system the information about airports and routes, we can make requests in a logical and structured way through the Skyscanner API to obtain flight offers. Skyscanner API is a useful and essential tool to develop this solution; it obtains the "best offer" for a given route by a simple request (offers are subject to airfare restrictions). To determine the extent of an offer, it is necessary to know the average price that has a route, and in this case, Skyscanner also offers an optimal solution.

Requests are performed following the scheme set out in our database for all stored routes of available airports. With this requests, we complete a first layer of flight fares, and the average prices for each route.

This process will be processed with an algorithm of minimum nodes cover (airports), to avoid overloading the system with duplicate requests.

At this stage, we already could publish flight offers, because fares are the lowest found searching by default. It is also advisable to develop a parallel system of Offer Management, which will determine the quality of the offer comparing with the average price of the route.

Once default fares are stored, we are at second phase (see Figure 6), in which it will be possible to make combinations between these routes. The first combination process is made with one level of depth, i.e., we expanded a node in one the level of depth search. The combinations obtained will be the result of filtering all possibilities with a variety of date and flight time restrictions.

At the third phase, we perform a process of comparing the existing offers in the system and new combinations of routes processed. This process of comparison will be made between the price of the new route obtained, and the average price of the route of the origin and destination of the new offer (the new route obtained). If not exist connection route, would be established a new one directly. On the contrary, if the route already exists and has a lower price than average, will be stored and offered as a possible offer.

After completing this processing of information, we are at phase four of the architecture. After storing a new route, the process is repeated continuously. The search for new routes stops when the price of the new route is higher than the average price. This search is based in a deep level determined by the amount of times of processed routes.

The combination process of possible new routes is activated when storing new data extracted through Skyscanner, optimizing with that, the response time to publish a new offer. The process of publishing results is present at all stages, listening new insertions into the database.

After presenting the architecture of our system, we obtain the knowledge to create the framework where the publication of flight offers is not limited by the restrictions that apply airlines to their airfares. This framework also noted for its ability to offer users the best flight offers, publishing them in an order that determines the quality of the offer. Figure 6 is a diagram of the processes that belong to the system architecture.

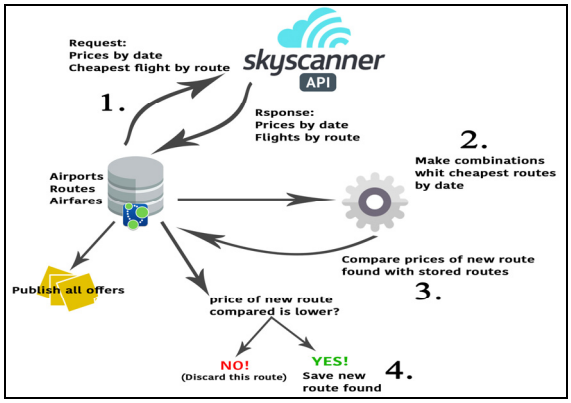


Figure 4: Architecture diagram.

In the next section we will give a working example how the system gives the output to users.

## 5.2 Problem Definition

Our intention is to implement a system to find the best offers in existing flights without having to perform a thorough and manual search based on routes and dates.

The flight search engine used to perform this analysis of prices is "Skyscanner", because it does not include internal rates when searching for flights, and automatically redirects to the main agents shopping online (Expedia, Opodo, eDreams, etc.) and major airline companies.

Doing various searches to check the veracity of this assumption, we can see that a flight from Barajas airport (Madrid, Spain) and destination airport New York John F. Kennedy (NY, USA) could cost nearly 550€. However, if we do an exhaustive search performing different route connections in Europe before crossing the Atlantic Ocean, we can obtain a price around 400€. In this case was chosen as the optimal connection Oslo Gardermoen airport (Oslo, Norway).

Therefore, if we search only the origin and desired destination, flight search engine will show us different air routes with a price about around 150€ more expensive.

Another case in which we can see that the search for lower prices are made according to the fares restrictions of holding corporations, is the alarming case to make trips from Ireland, in any of its cities, and as a destination Puerto Vallarta (Jalisco, México).

If we search with origin airport Dublin (Dublin, Ireland) and destination Puerto Vallarta (Jalisco, México), we get a price almost 900€, performing all routes with an airline holding, Skyteam, including flights from airlines like Delta, Aeroméxico and Air France.

On the contrary, we found another cheapest option for the same date, but under different corporate holdings, and also making connections within Europe before crossing the Atlantic Ocean. In this case the airport connection between the origin city and destination would the London Gatwick airport, where the fare is less than 100€ by Ryanair. From London Gatwick airport to Puerto Vallarta with the company Thomson Airways would have a fare less than 400€. A saving of almost 400 euros! Therefore, for the same origin and destination, performing different routes, we can be able to get a price 50% lower than shown by default search engine. Such examples can be found in several destinations, both on flights from Europe to America, like from Europe to Asia.

In this part of the process of analysis of results, is where take part the functionality of implementing optimal search minimum cover algorithms of our proposal. The intention is to filter and limit searches to a certain depth, obtaining as a result offers lower



Table 3: Skyscanner search at 06/04/2015.

| Origin | Destination | Departure | Arrival | Price | Company |
|---|---|---|---|---|---|
| MAD | JFK | 29/01/2016 | 30/01/2016 | 552€ | Delta + Aeroméxico + Air France |
| JFK | MAD | 05/02/2016 | 06/02/2016 | | |
| **Origin** | **Destination** | **Departure** | **Arrival** | **Price** | **Company** |
| MAD | OSL | 29/01/2016 | 29/01/2016 | 173€ | B. Airways + Iberia |
| OSL | MAD | 06/02/2016 | 06/02/2016 | | |
| OSL | JFK | 29/01/2016 | 29/01/2016 | 251€ | Norwegian |
| JFK | OSL | 05/02/2016 | 06/02/2016 | | |

Table 4: Skyscanner search at 06/04/2015.

| Origin | Destination | Departure | Arrival | Price | Company |
|---|---|---|---|---|---|
| DUB | PVR | 11/04/2015 | 12/04/2015 | 855€ | United |
| PVR | DUB | 25/04/2015 | 27/04/2015 | | |
| **Origin** | **Destination** | **Departure** | **Arrival** | **Price** | **Company** |
| DUB | LGW | 11/04/2015 | 11/04/2015 | 96€ | Aer Lingus |
| LGW | DUB | 26/04/2015 | 26/04/2015 | | |
| LGW | PVR | 11/04/2015 | 11/04/2015 | 387€ | Thomson Airways |
| PVR | LGW | 25/04/2015 | 26/04/2015 | | |

than average price established for a given route, and after that, making possible combinations of offers obtained above.

The application of this algorithm would not be useful for a path when repeating the process, and did not obtain results in a final path with lower price than obtained when it made searches by default way.

# 6 CONCLUSIONS

In our work we conclude that relational databases do not offer the necessary flexibility to process flight management problem with optimal solutions, due to strong structural constraints and a weak dynamic indexing. We reach the conclusion that Graph Databases are the best solution for storing and processing data flight fares and we choose Neo4J.

We review and analyse diverse existing commercial solutions and tools to implement new features using data integration between different platforms. To overcome the restrictions on flight fares was our main motivation for implementing a data mining system, in such a way, that we dispose a system free of restrictions. With this system, we can show to final customer the best possible flights fares, combining connections between different airlines and even different airline holding companies.

Also, the business alliances between airlines hinder the process to obtain the most competitive offers, and therefore, when choosing the cheapest way to travel, these types of alliances would be damaging the final consumer. It is true that this kind of enterprise unions provides many facilities to customers, due to the centralization of the purchase (Pels, 2001). But, how much would be willing to pay for this convenience? This is an open question that we would like to have an answer in a near future.

We present a framework to find the best deals on airfares, and with them make combinations that have as results new flight routes, offering to the customer a trip with a truly affordable price. To the best of our knowledge this is the first proposal of this new feature that is missing from all OTA systems.

As future work we propose a new feature implementated by major GDS, or flight search companies, to allow a real optimized search. Resulting that the customers have the option to find the best flight fare possible for a specific date.

We also must take in consideration the amount of information processing that this feature needs (IBM Big Data, 2015), because it would have to perform an analysis of real-time fares from a massive amount of data and could easily overload a system. Therefore, our system should be able to process flight fares from more than 30M of annual flights, and the forecast worsens in 2030, can reach up to 60M annual commercial flights.

# REFERENCES


AXSES, 2013. *AXSES Travel marketing about global distribution, GDS history and GDS marketing. Axses.com.* Retrieved August 20, 2013, from http://axses.com/encyc/archive/arcres/arcrates/users2/globalmarketing_about_gds.cf

Binstock, M., 2015. Oracle and the End of Programming As We Know It. Retrieved April 11, 2015, from http://www.drdobbs.com/jvm/oracle-and-the-end-of-programming-as-we/232901227

Guterman, J., 2015. *Release 2.0: Issue 11. O'Really Media* http://www.oreilly.com/data/free/release-2-issue11.csp

IBM Big Data. 2015. *What is Big Data – United States*. (2015, January 27). Retrieved April 11, 2015, from http://www.ibm.com/big-data/us/en/

Jonas, D., 2013. *Vision 2020: Old Dogs, New Tricks And The Future Of GDSs.* Retrieved November 22, 2013.

Kamel, M. N., 1994. A prototype rule-based front end system for integrity enforcement in relational databases: An application to the naval aircraft flight records database. pp. 713–723.

Pels, E., 2001. A note on airlines alliances. *Journal of Air Transport Management* Volume 7, Issue 1, pp. 3–7

Valles, A.J., VanLoy, J.A., 1991. An Expert Auditing System for Airline Passenger Tickets. In *Proceedings of the The Third Conference on Innovative Applications of Artificial Intelligence* (IAAI-91).

Woodie, 2015. *Neo Rides the Graph Database Surge*. Retrieved April 7, 2015.