

Transport and Telecommunication, 2014, volume 15, no. 4, 280–291
Transport and Telecommunication Institute, Lomonosova 1, Riga, LV-1019, Latvia
DOI 10.2478/ttj-2014-0024

A TRAVELLER INFORMATION SYSTEM: MINIMISATION OF THE NUMBER OF GRAPHS' NODES INVOLVED WHEN PROCESSING ROUTE REQUESTS

Zakaria Bendaoud¹, Karim Bouamrane²

University of Oran

Department of computer science

LIO Laboratory, BP 1524, EL M' Naouaer, 31000, Algeria

Phone: ¹+213796831225, ²+213771238121

E-mail: ¹cherad7@hotmail.com, ²kbouamranedz@yahoo.fr

The number of people using public transport is continuously increasing. Transport companies want to fulfil travellers' expectations wherever possible. However, the great number of public transport companies operating in the same area can sometimes confuse travellers as to which route they should take and how to obtain the information relative to their journey. In this paper we suggest integrating several traveller information systems from different companies into the same multimodal information system, offering companies the choice not to share their data. This encourages them to join the system. Additionally, we have minimised the number of nodes involved when processing travellers' requests in order to simplify the calculation process. To put our plan into action, we have opted for a multi-agent system coupled with the Voronoi decomposition for managing the network.

Keywords: Information system, traveller information, multi-agent systems, Voronoi decomposition

1. Introduction

The public transport sector is undergoing rapid expansion. Companies are increasingly innovative in how they meet travellers' needs. Each company has its own stops, modes of transport, and pathfinding algorithms, and interacts with the traveller through its information system, often via the internet.

A company's traveller information system (TIS) allows travellers to calculate their route and obtain the information they require (Adler and Blue, 1998). They can use two types of network: unimodal or multimodal. The first type takes a single mode of transport into account, whilst the second type takes at least two into consideration (Vasco et al., 2013). An effective TIS even considers walking as a possible mode of transport. With the competition and diversity that the transport sector is undergoing, travellers must have a sound knowledge of routes, websites and interchange stations served by each company in order to obtain the multimodal information desired. In the case of a long distance journey, it is very probable that travellers may have to use numerous operators' information systems in order to find the information they require.

To avoid travellers' manual searches, and to guarantee their comfort, we propose in this paper integrating multiple traveller information systems into the same multimodal information system. It is important to note that usually, a company must share all of its information, data and algorithms to belong to a multimodal information system. However, this data can be crucial for competition and mastery of the market. The potential loss of this valuable data can dissuade companies from joining a system. To remove this obstacle and encourage transport companies to join a multimodal information system, we suggest an architecture in which every company retains its autonomy from the point of view of data management. When the proposed system receives a request from a traveller, it breaks it down into multiple sub-requests and interrogates the companies selected for the desired route in order to provide sub-responses. The fusion of these sub-responses forms the final information offered to the traveller.

During bibliographic research in this field (Schulz, 2005; Zidi, 2006; Kamoun, 2007; Zgaya, 2007; Pajor, 2009; Zhang et al., 2012) we did not identify any project recommending a multimodal information system that integrates the information systems of different transport companies and allows the size of the search space¹ to be minimised when processing of travellers' requests. This work addresses the size of graphs when applying pathfinding algorithms. Instead of applying pathfinding algorithms to different

¹ The search space is the information search domain. In the case of transportation and route request processing, the search space represents the graphs of the companies' networks.

transport companies' graphs in their entirety, we suggest a virtual break down of the different networks according to their congestion. The decomposition is based on the Voronoi diagram (Slimani, 2011), in order to apply the pathfinding algorithms to the sub-networks. This technique has allowed us to minimise the number of nodes in graphs involved when processing route requests, which simplifies the calculations process.

The rest of the paper is organized as follows: section 2 provides an overview of the research area. Section 3 discusses the methods we have used to develop our system. In section 4 we present the results obtained following use of our approach. These results are discussed in section 5. We finish with a conclusion, including future possibilities.

2. Related Work

The first generation of traveller information systems worked to improve traffic flow by keeping motorists updated on traffic and disruptions via radio and signposts (Adler and Blue, 1998). At the end of the 1980s the city of Stockholm's timetables were computerized and gave rise to the first traveller information system (Kramers, 2014), accessed via a telephone number and using touch keys (*#). Advances in information systems and the internet, and the appearance of navigation systems such as the Global Positioning System (GPS) have contributed to developments in traveller information systems. A significant amount of research has addressed issues concerning passenger information and pathfinding algorithms. We shall list a few of them below.

In France, (Petit-Roze et al., 2004) proposed a multimodal mobility aid service, based on personalising information according to a traveller's profile. (Zidi, 2006) presented an interactive system to aid travelling under normal and heavy congestion conditions. This system aims principally to minimise travellers' waiting times. (Zgaya, 2007) suggested a multi-agent system that used genetic algorithms to search for and compose services linked to multimodal transport between several competing operators. (Kamoun, 2007) proposed a cooperative mobility information system by automating the pathfinding processes. In Germany the "DELFI" project, launched in 1996, gave rise to the first multimodal information system on the national level. The network is divided into sixteen regions and each federal state manages the networks of its own public transport companies. This requires applications to be integrated via standardised interfaces using the COBRA protocol (Danflous, 2000). (Schulz, 2005) carried out intensive research into the models of network representations and the passage from a unimodal to a multimodal network. (Pajor, 2009) undertook research on multimodal routing between intercontinental networks. The aim was to make a model of the network, apply pathfinding algorithms and finally to use optimisation techniques to accelerate the search process.

Other applications have been developed to aid multimodal mobility. In the United States, (Zhang et al., 2011; Li et al., 2010; Jariyasunant et al., 2010) presented applications to facilitate travel, taking into account real-time data. In the United Kingdom a number of multimodal information systems calculate travel from door to door, such as "Transport Direct", "Journey Plan", and "TFL" (Danflous, 2003). Switzerland possesses its own multimodal information system which is utilised at a national level, and is based on a totally centralised architecture (Danflous, 2001). In the Netherlands, the "9292ov" system allows for travel from one location to another. Initially launched using telephony, it is now accessible over the internet (Danflous, 2006). (Kumar, 2005) developed a traveller information system for journeys in the city of Hyderabad in India. Other firms such as Trapeze, Jeppesen, and Google have developed their own mobility aid products.

The cited works present a few aspects of multimodal information systems. All of these models are based on totally or partially centralised architectures. In centralised architectures, all the nodes of all public transport companies form a single network in which a pathfinding algorithm is applied to compose the itinerary required by the traveller. In partially centralised architectures (architecture centralised by zone), the networks of transport companies which operate within the same geographic perimeters form a super-network. Communication between these super-networks allows for a response to travellers' requests. These two methods, although effective, require all the nodes within each company's graph to be systematically taken into account, including those not relevant to the request. Processing a route calculation in a centralised system implies the manipulation of all companies' graphs, even if the journey only concerns a single company. In order to remedy this problem, we have proposed breaking down different companies' networks (graphs) according to congestion. Our system has an overview of all the companies' networks and breaks them down into several zones corresponding to the Voronoi diagrams (Slimani et al., 2011), then informs the various companies about which zone each node should

belong to. Upon receiving a route request, the proposed system only requests information from the relevant sub-networks of the companies concerned.

The systems cited above were concerned with processing route requests between several companies. Nevertheless, the traveller information can be of several types: information about routes, real-time information about disruptions, or other tourist information about monuments or other places that the traveller can discover whilst travelling (Zgaya, 2007). In this work we are interested in route requests and enquiries concerning tourist information.

3. Towards a Multimodal Information System

This work aims to offer a multimodal and multi-operator information system which minimise the number of nodes utilised in dealing with requests. Our system has an overview of all the companies' networks, it divides different companies' networks into multiple zones depending on the stations' congestion. Upon receiving a request, it localises the zones concerned and then searches within the transport companies' sub-networks which belong to these zones. In order to divide the general network into zones, we have adapted the Voronoi diagram to a transport network. Our system architecture makes use of a multi-agent system.

3.1. Adapting Voronoi diagram for transport

The Voronoi diagram (VD) is an interesting structure in computational geometry (Berg et al., 2008). It is the spatial solution for the nearest-neighbour problem. Given a set of N point sites S ($S = \{S_1, S_2, \dots, S_N\}$), for each site S_i , $S_{i,x}$ and $S_{i,y}$ denote the x -coordinate and y -coordinate of S_i , respectively. The VD of S ($VD(S)$) is the subdivision of the plane into N disjoint regions/cells according to the nearest-neighbour rule: each region/cell corresponds to one site and is composed of the set of the plane's points that have this site as the nearest site among all sites of S (Figure 1) (Slimani, 2011). A requisite step in the Voronoi decomposition therefore is to initialise pre-selected points.

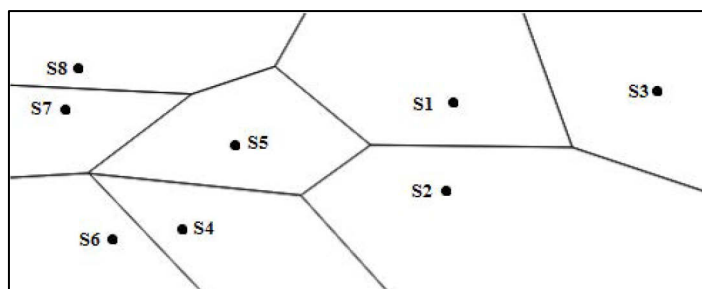


Figure 1. Voronoi diagram of eight randomly selected locations

The Voronoi diagram is applied to our problem as follows: let S be the set of all the stations of all the transport companies $S = \{S_1, S_2, \dots, S_n\}$. Let $S'' = \{S_1, S_2, \dots, S_m\}$ a subset of S , ($S'' \subset S$ and $m < n$) which represents the stations that we select according to their congestion (heavily congested stations, medium congestion stations, or uncongested station). One Voronoi cell of station S_1 ($S_1 \in S''$) is defined by the stations belonging to the set ($S - S''$) and closer to S_1 than to the other S'' stations. The number of cells (zones) depends on the number of elements within S'' . In order to calculate the distances between two stations, we have calculated the great circle distances (Gade, 2010). For this calculation we have assumed the Earth to be a perfect sphere with a diameter of 6367 km and we have used the formula presented in (Gade, 2010):

$$D(\alpha, \beta; \alpha', \beta') = 2R \arcsin \sqrt{\sin^2\left(\frac{\alpha' - \alpha}{2}\right) + \cos \alpha \cdot \cos \alpha' \cdot \sin^2\left(\frac{\beta' - \beta}{2}\right)}$$

R is the diameter of the sphere (Diameter of the Earth ~ 63670000 metres). The latitude and longitude, in radians, of the first station are α and β respectively, and the latitude and longitude, in radians, of the second station are α' and β' respectively. A transport company's stations can then belong to numerous zones according to their congestion. The union of sub-networks of a transport company forms the totality of its network. Only a transport company's sub-networks concerned by a request are considered, and not its total network.

3.2. Our System Architecture

Multi-agent systems (MAS) divide a process between several agents. Instead of having a single programme that manages the entire system, the problem is divided and each agent has the task of resolving a sub-problem. The final solution is obtained through the interaction between different agents, referred to as distributed artificial intelligence (Ferber, 1995).

Numerous works in the transport domain have made use of multi-agent systems to achieve:

- The modelling of urban transport networks.
- The regulation of disturbances in transport networks.
- The personalisation of multimodal information.

In order to respond to passengers' enquiries and produce multimodal information, a multimodal traveller information system must access the information systems of the various companies which compose it in order to work out the final answer. It becomes the mediator between the various information systems, it must find the right sources in a distributed environment, manage the heterogeneity of the information, and present travellers with clear solutions.

Our multimodal information system is organized around five types of agents (figure 2): interface agents (IA), identifier agents (IdA), directory agents (DA), collector agents (CA), and fusion agents (FA).

Interface agents act as the interface between our system, on the one hand, and the travellers or companies who wish to enrol, on the other. The identifier agents identify a request as a route request or a request for tourist information and send an enquiry to the directory agent for the search space corresponding to the request. The directory agents have an overview of the stations and networks of all transport companies. They enrol new transport companies wishing to belong to the system, carry out a virtual break down of the network according to the Voronoi diagram, and provide the search space by selecting the companies and exchange stations concerned. The collector agents are mobile agents which move through the search space to collect the necessary information from each company concerned by the request. Finally, the fusion agents compose the responses according to users' profiles.

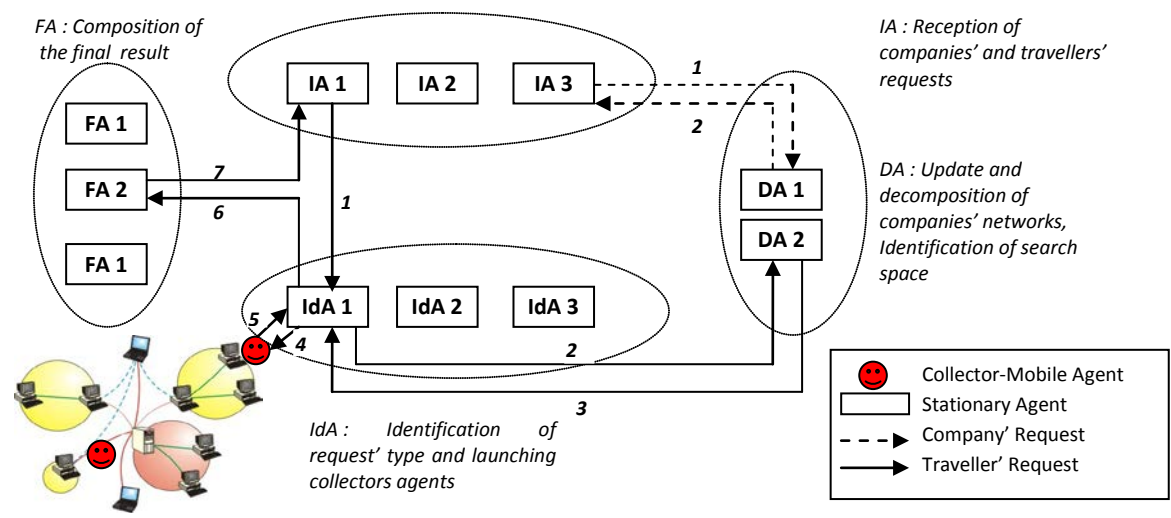


Figure 2. Architecture of our system

Interface Agent

The interface agent handles two different situations:

- Either it receives a request from a transport company that wants to integrate the system. Then it passes it onto the directory agent.
- Or it receives a request from a user searching for multimodal information. If the user submitting the request can be identified, his or her profile is extracted so that answers are processed and presented according to his or her needs. Otherwise, the user is assumed to have a standard profile and the request is submitted to the identifier agent.

Figure 3 is a diagram showing the interface agent's activities.

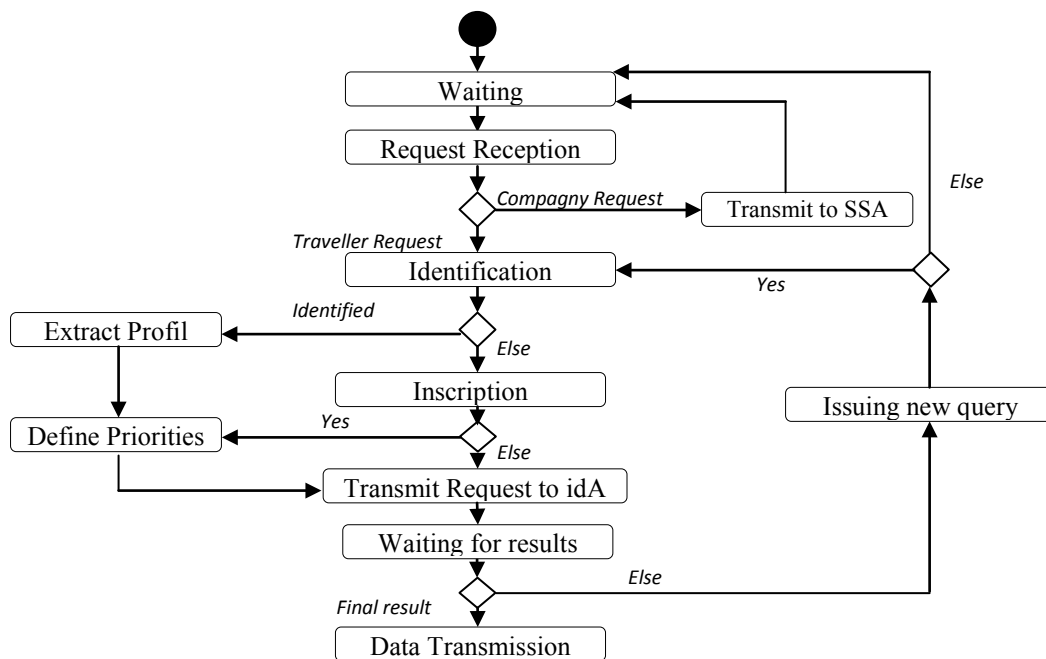


Figure 3. Activity diagram of the interface agent

Identifier Agent

The identifier agent receives requests from the interface agent. These requests can be of two sorts: either a route request, or a request for tourist information. If it is a route request, the agent determines the departure station and the arrival station and asks the directory agent for the search space. If the request is concerning tourist information, the name of a monument or square is extracted, and a request to determine the search space is sent to the directory agent. If the request is successful, the collector agent is launched to collect the required information from the different companies involved. Once the results have been obtained, it transmits them to the fusion agent. Figure 4 is a diagram showing the identifier agent's activities.

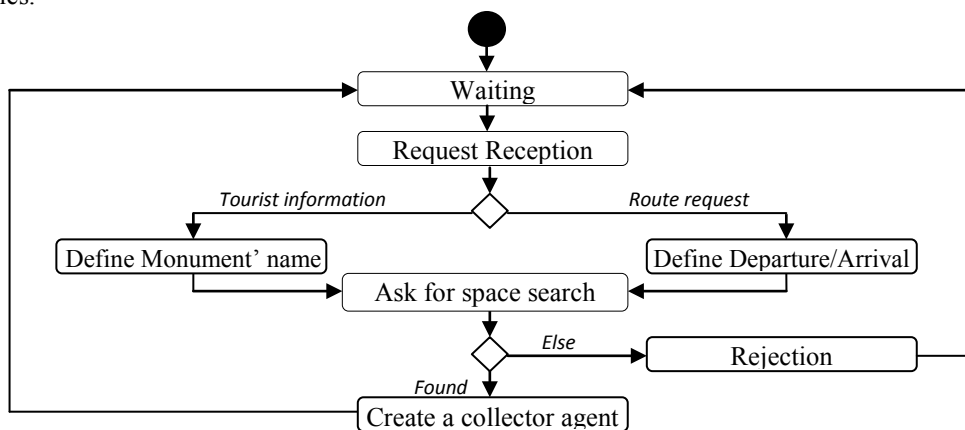


Figure 4. Activity diagram of the identifier agent

Directory Agent

The directory agent has two functions:

- It receives enrolment requests from new companies through the interface agents. Information including lines, exchange stations, and network addresses, is provided. To update the network zones, the network is broken down again using the Voronoi algorithm which is presented in section 3.3.

- It receives requests from the identifier agents concerning the search information space. Such requests are of two types:
 - The request has, as an argument, the name of a monument, a square, or a building. The directory agent localises the zone this site is designated to and the stations and stops closest to this site, using the great circle distances, which are the shortest distances between two points on a sphere, according to the formula presented in (Gade, 2010). The search space is defined by all of the companies responsible for these stations.
 - The request has, as arguments, a departure station and an arrival station. The directory agent selects the arrival and departure zones and, where necessary, the intermediary zones, and all companies involved in the request. The search space algorithm when dealing with route requests is presented in rubric 3.4.

Figure 5 is a diagram showing the directory agent's activities.

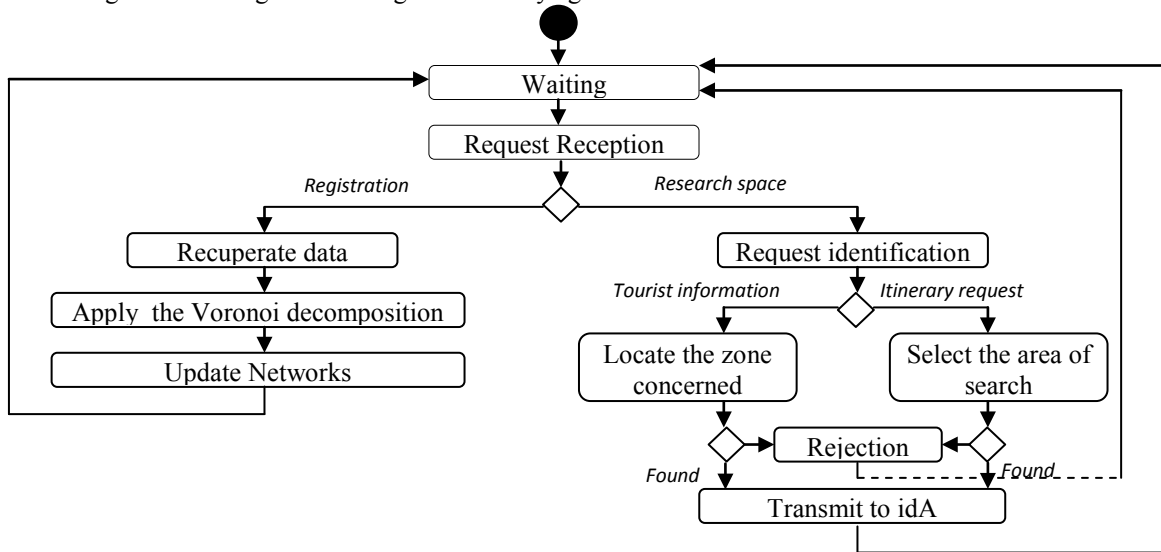


Figure 5. Activity diagram of the directory agent

Collector Agent

The collector agent is a mobile agent with the capacity to move throughout the system. It collects necessary information from each node. In our case, a node represents the information system of a company belonging to our multimodal information system. The network address and mode of communication of each company is at the collector agent's disposal. It collects the sub-answer which has been demanded of each node (company). Once all of the nodes have been visited, the collector agent returns to the host node and communicates the collected data to the fusion agent. Figure 6 is a diagram showing the collector agent's activities.

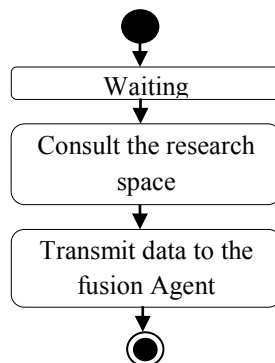


Figure 6. Activity diagram of the collector agent

Fusion Agent

The fusion agent is responsible for bringing together the collection of answers gathered by the collector agent. If the request is concerning tourist information, the reply is constant for all profiles. If the request is concerning an itinerary, the reply can vary depending on the traveller's profile. In order to better reply to travellers' expectations, we have considered two selection criteria which the user specifies at registration: age and preferences. Figure 7 is a diagram showing the fusion agent's activities.

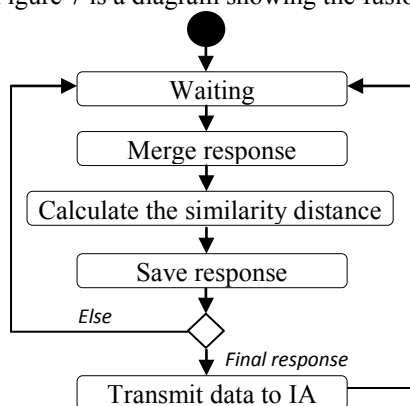


Figure 7. Activity diagram of the fusion agent

3.3. Applying the Voronoi Algorithm

As aforementioned, the Voronoi diagram consists of creating zones relative to pre-selected points. A zone is defined by the collection of points that are closest to one of the points chosen in advance. In our case, the points represent stations and stops and are pre-determined according to their congestion. The Voronoi algorithm, adapted to our study (the multimodal information system) is presented below:

Algorithm 1: adaptation of the Voronoi decomposition

1. **For all** preselected stations **Do**
 2. **For all** non-visited stations **Do**
 3. calculate the distance with preselected stations for zoning
 4. Select the minimum distance
 5. Allocate the station to the adequate zone
 6. **End For**
 7. **End For**
 8. Update zones
 9. **End**
-

3.4. Algorithm relative to the search space

This algorithm is used by the directory agent to determine the search space when dealing with a route request. The search space consists of providing pairs (companies/zones) involved in a route request. The algorithm determines the zones and companies responsible for departure and arrival stations, then determines the search space according to one of four cases:

- The departure and arrival zones are identical and the arrival and departure stations belong to the same company: the search space is defined by this company's sub-network within this zone.
- The departure and arrival zones are different but the arrival and departure stations belong to the same company: the search space is defined by the union of this company's two sub-networks in these two zones.
- The departure and arrival zones are identical but the departure company and the arrival company are different: an adjacency graph is established between the companies to determine the intermediary companies which could take part to satisfy the request. The adjacency graph is represented by an adjacency matrix, and the exchange stations represent the intersection

between the different companies in the adjacency table. The Yen algorithm (Yen, 1971) to calculate the shortest routes is launched to determine the intermediary companies concerned. The request is broken down into numerous sub-requests according to the exchange stations.

- The departure zone and arrival zone are different, and the departure and arrival company are also different: firstly, the Dijkstra algorithm (Dijkstra, 1959) is applied to an adjacency graph between the zones to determine the relevant intermediary zones. Then the Yen algorithm (Yen, 1974) is applied to the adjacency graph in each of the relevant zones to determine the relevant intermediary companies as explained in the point above.

4. Results

In order to validate the proposed approach, we have used real data drawn from the timetables of several public transport companies in the city of Oran. We worked on a computer equipped with an i3-380 processor, 4 gigabytes of RAM and a 750 GB hard drive. The programming language used is Java (Dale, 2011) and the platform used for the multi-agent system is JADE (Bellifemine, 2007). The application can be accessed via a web portal. The data are taken from the timetables of three companies operating three different modes of transport, i.e. bus, tram, and train. The bus network consists of 182 nodes and 52416 daily arrivals/departures. The tram network consists of 32 nodes and 5760 daily arrivals/departures. The railway network consists of 12 stations and 1850 daily arrivals/departures.

We calculated the number of nodes involved in graphs when processing route requests in five types of architecture:

- Centralised architecture
- Architecture centralised by zone
- Our architecture, initialising the Voronoi algorithm with heavily congested stations...(1)
- Our architecture, initialising the Voronoi algorithm with medium congestion stations...(2)
- Our architecture, initialising the Voronoi algorithm with uncongested stations...(3)

Table 1 shows the number of nodes involved in a journey of 800 metres, table 2 for a journey of 5100 metres, and table 3 for a journey of 12000 metres:

Table 1. Number of nodes involved in processing a route request of 800 metres

Architecture	Number of nodes	Number of edges
Centralised	226	503
Centralised by zone	101	197
Voronoi (1)	36	63
Voronoi (2)	64	152
Voronoi (3)	94	193

Table 2. Number of nodes involved in processing a route request of 5100 metres

Architecture	Number of nodes	Number of edges
Centralised	226	503
Centralised by zone	173	274
Voronoi (1)	78	158
Voronoi (2)	123	215
Voronoi (3)	171	404

Table 3. Number of nodes involved in processing a route request of 12000 metres

Architecture	Number of nodes	Number of edges
Centralised	226	503
Centralised by zone	226	503
Voronoi (1)	112	232
Voronoi (2)	164	308
Voronoi (3)	226	468

From tables 1, 2, and 3 it is clear that the number of nodes involved when breaking down the network according to the Voronoi diagram is less than in other architectures. Processing requests within an architecture based on Voronoi operates over a sub-network and not the network in its entirety.

The choice of initialisation points (N) in our approach is very important. We have calculated the processing time whilst varying the number of points and their nature (heavily congested, medium congestion, or uncongested). We have tested three different route requests (800, 5100, and 12000 metres).

Figure 8 shows the calculation time in milliseconds (ms) in each case. The curves on the left give the response time when initialising the Voronoi algorithm with three points, the curves in the middle show the response time when initialising the Voronoi algorithm with four points, and finally the curves on the right show the response time when initialising the Voronoi algorithm with five points. The curves in blue show the response time when initialising with heavily congested stations; those in red when initialising with medium congestion stations; and those in green when initialising with uncongested stations. The results clearly show that initialising the algorithm with heavily congested stations offers the best response times, regardless of the number of points.

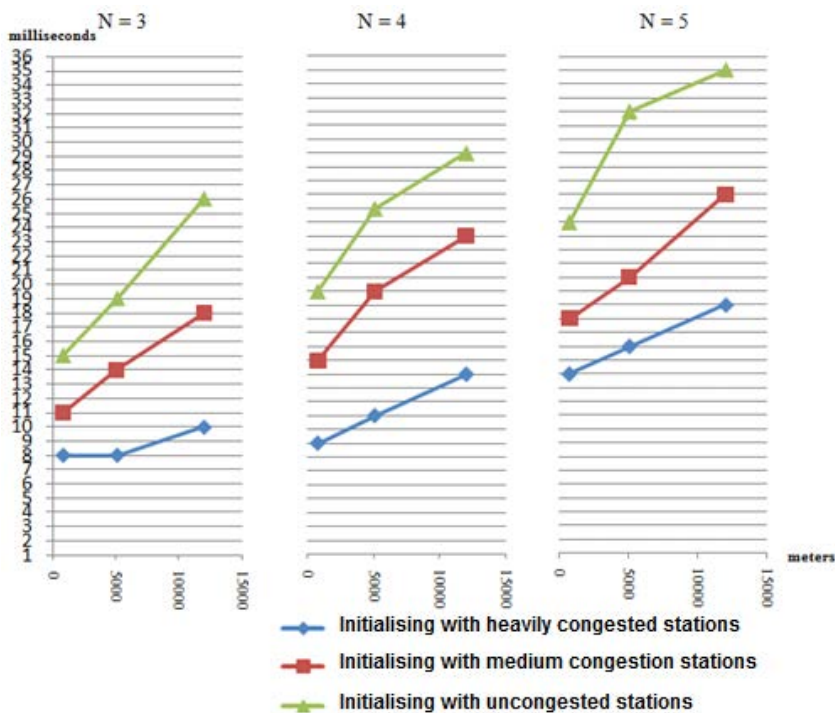


Figure 8. Comparison of the response times depending on the Voronoi initialisation

The following figure gives a preview of our programme being used online. The screenshot on the left shows access via a computer whilst the one on the right gives a preview of running the programme on a mobile phone.

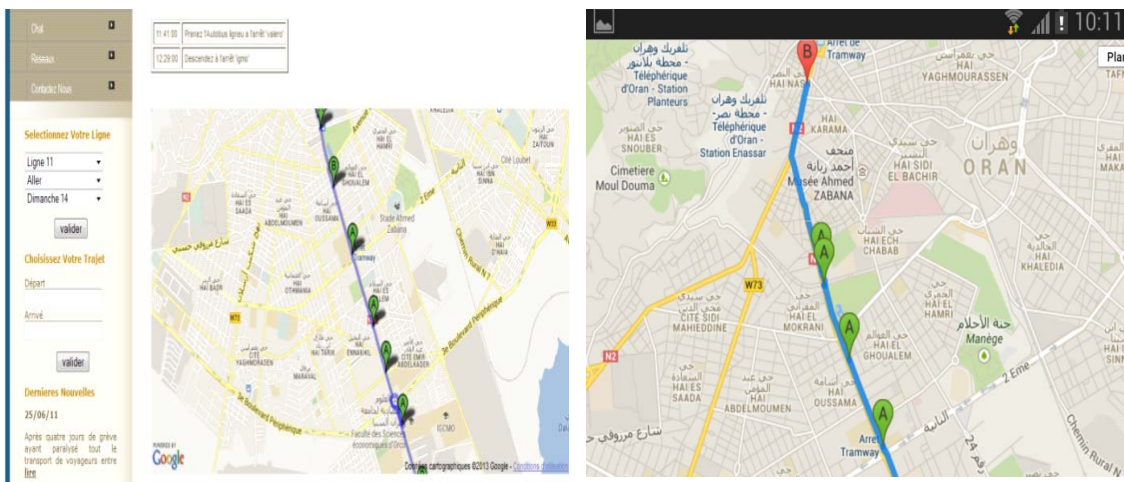


Figure 9. A preview of our system being used online

5. Discussion

The results in tables 1, 2, and 3 show that the number of nodes involved in processing itinerary requests within a centralised architecture is a constant 226 nodes. This number represents the total number of stops and stations of all transport companies belonging to the system. The centralised architecture takes into account all the networks of all the companies, regardless of the journey distance. The centralised architecture requires the transport companies to share their information. The sharing of information by a transport company poses the problem of controlling and modifying the data, besides issues concerning its market value. Transport companies stand to lose control of their data when it is shared and made available to a multimodal information system, plus this can cause update problems. The centralization of data is far better adapted to transport companies with small networks, where the number of nodes is limited, because it facilitates setting up and a better control of data. It is important to remember that in a transport network graph each station or stop is represented by several nodes: a physical node, and an event node for each departure or arrival (Pajor, 2009).

In partially centralised architectures (architecture centralised by zone), the number of nodes involved in processing requests varies according to journey distance (tables 1, 2, 3): 101 nodes for a journey of 800 metres, 173 nodes for a journey of 5100 metres, and 226 nodes for a journey of 12000 metres. The aim of an architecture centralised by zone is to group together the transport companies' networks which operate within the same geographic perimeters and to create a single network. In this case, when processing a route request, it is the networks of all the companies which belong to the zone that are taken into account: the longer the journey, the higher the probability of passing through several zones. In an architecture centralised by zone, the number of nodes involved depends on the journey distance and on the number of zones relevant to processing the request. The distribution of processes according to geographic zones can work around breakdowns more effectively than a centralised architecture, but who owns the information put at travellers' disposal is always an issue, especially in cases of inaccurate information.

The results obtained when processing requests within our proposed architecture show that the number of nodes involved is significantly lower than the numbers of nodes involved in a centralised architecture and an architecture centralised by zone. This considerable reduction in the number of nodes in comparison with other existing architectures can be explained by the fact that requests are processed using the transport companies' sub-networks. In effect, the aim of our work is to create the best zones in order to manage travellers' requests most efficiently. Nevertheless, the number of nodes involved in processing a request can change for the same route, depending on the initialisation of the Voronoi algorithm. Initialising the Voronoi diagram with heavily congested stations allowed the number of nodes involved to be advantageously minimised: 36 nodes for an 800-metre journey, 78 nodes for a 5100-metre journey, and 112 nodes for a 12000-metre journey. This number increases when initialising with medium congestion stations, and peaks when initialised with uncongested stations. Initialising with heavily congested stations clearly gives the best results, because they allow for zones with a heavy concentration of traffic. The choice of initialisation points remains difficult, and confronts the cold start problem, familiar to clustering algorithms. Numerous works have suggested solutions for initialising clustering algorithms (El Agha and Ashour, 2012; Reddy and Jana, 2012).

Figure 8 provides a visual presentation of the response times between different initialisations of the Voronoi algorithm diagram when processing a request according to journey distance and the number of zones chosen. Initialising with heavily congested stations provided the shortest time responses. The calculation time increases slightly with an increase in the number of zones, due to the number of messages exchanged between agents. It is necessary, therefore, to find the right equilibrium between the number of zones created and the number of messages exchanged between the agents. These should be paired with a good selection of initialisation points for the Voronoi algorithm, which could potentially be located using data analysis techniques. Finally, if choosing uncongested stations is the most costly in terms of response time, it would be interesting to experiment on a national level and test route requests in secluded villages or urban and suburban neighbourhoods.

Moreover, our architecture based on Voronoi diagrams offers transport companies the autonomy of not being forced to disclose their data. Transport companies retain control of information and our system becomes a mediator between the information systems of the various transport companies. The independence offered to companies should encourage them to integrate our system.

6. Conclusions

In this project we have proposed a multimodal traveller information system which handles requests concerning itineraries and information about exploring tourist attractions. This dual purpose system satisfies the needs of its clients by offering them a single portal for their journeys despite the existence of numerous companies. But it also satisfies transport companies needs' by eliminating the necessity for them to share data of commercial value. We have suggested breaking down all the companies' networks along the lines of the Voronoi diagram in order to minimise the number of nodes involved in processing itinerary requests. Upon receiving a route request, the system breaks it down into sub-requests and selects the relevant zones and companies. The mobile agents collect the replies to these sub-requests, which are brought together to form the final solution for the traveller. Choosing the initial points of the decomposition algorithm is often crucial: in retrospect, our system would benefit from analysis of data in order to automate the choice of points, as well as protocols on negotiation and cooperation between the agents for managing the network when services are disrupted. Additionally, the project could be advanced for application on a national level.

References

1. Adler, J.L., Blue, V.J. (1998) Toward the design of intelligent traveler information systems, *Transportation Research Part C*, 6, 157-172
2. Bellifemine, F.L., Caire, G., Greenwood, D. (2007) *Developing Multi-Agent Systems with JADE*, Wiley. 300 p.
3. Berg, M., Cheong, O., van Kreveld, M., Overmars, M. (2008) *Computational Geometry: Algorithms and Applications*, 3rd edition. Springer-Verlag. 386 p.
4. Dale, N., Joyce, D.T., Weems, C. (2011) *Object-Oriented Data Structures Using Java*, 3rd edition. Jones & Bartlett Learning. 802 p.
5. Danflous, D. (2000) *Déploiement national des systèmes d'information multimodale, DELFI : l'exemple Allemand*. Report CERTU/CETE Méditerranée France. ISSN : 1263-2570 ISRN : CERTU/RE-00-17-FR. France. 36 p.
6. Danflous, D. (2001) *Déploiement national des systèmes d'information multimodale, GOFAS : l'exemple Suisse*. Report CERTU/CETE Méditerranée France. ISSN : 1263-2570 ISRN : CERTU/RE-02-08-FR. France. 42 p.
7. Danflous, D. (2003) *Déploiement national des systèmes d'information multimodale, Transport Direct : l'exemple Anglais*. Report CERTU/CETE Méditerranée France. ISSN : 1263-2570 ISRN : CERTU/RE-03-14-FR. France. 56 p.
8. Danflous, D. (2006) *Déploiement national des systèmes d'information multimodale, 9292 : l'exemple des Pays-Bas*. Report CERTU/CETE Méditerranée France. ISSN : 1263-2570 ISRN : CERTU/RE-06-06-FR. France. 40 p.
9. Dijkstra, E.W. (1959) A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1, 269-271
10. El Agha, M., Ashour, W.M. (2012) Efficient and Fast Initialization Algorithm for K-means Clustering. *International Journal of Intelligent Systems and Applications*, 4, 21-31.
11. Ferber, J. (1995) *Les systèmes multi-agents, vers une intelligence collective*. InterEditions, 522 p.
12. Gade, K.. (2010) A Non-singular horizontal Position representation, *The journal of navigation*, 63, 395-417.
13. Jariyasunant, J. (2010) Mobile transit trip planning with real-time data. *Transportation Research Board Annual Meeting, Washington, D. C.*
14. Kamoun, M.A. (2007) *Conception d'un système d'information pour l'aide au déplacement multimodal : Une approche multi-agents pour la recherche et la composition des itinéraires en ligne*. PhD thesis, University of Lille, France, 180 p.
15. Kramers, A. (2014) Designing next generation multimodal traveler information systems to support sustainability-oriented decisions, *Environmental Modelling & Software*. In Press, Corrected Proof, <http://www.sciencedirect.com/science/article/pii/S1364815214000279#FCANote>, Available online 30 January 2014,
16. Kumar, P., Singh, V., Reddy, D. (2005) Advanced traveller information system for hyderabad city. *IEEE Transactions on Intelligent Transportation Systems*, 6, 26-37.

17. Li, J.Q., Zhou, K., Zhang, L., Zhang, W.B. (2010) A multimodal trip planning system incorporating the park-and-ride mode and real-time traffic/transit information. *Transportation Research Board Annual Meeting, Washington, D. C.*
18. Pajor, P. (2009) *Multi-modal route planning*. Master thesis, Karlsruhe Institute of Technology, Germany, 165 p.
19. Petit-Roze, C., Anli, A., Grislin-LeStrugeon, E., Abed, M., Uster, G., Kolski, C. (2004) Système d'information transport personnalisée à base d'agents logiciels, *Revue Génie Logiciel*, 70, 29-38.
20. Reddy, D., Jana, P.K. (2012) Initialization for K-means clustering using Voronoi diagram. *Procedia Technology*, 4, 395-400.
21. Schulz, F. (2005) *Timetable Information and shortest path*. PhD thesis, University of Fridericiana zu Karlsruhe, Germany.
22. Slimani, H., Najjar, F., Slimani, Y. (2011) Voronoi-Neighboring Regions Tree for Efficient Processing of Location Dependent Queries, *International Journal of Advanced Science and Technology*, 33, 101-119.
23. Vasco, R., Fabian, M., Giuseppe, P., Roberto, P. (2013) Rail and multi-modal transport, *Research in Transportation Economics*, 41, 17–30.
24. Yen, J.Y. (1971) Finding the K shortest loopless paths in a network. *Management Science*, 17, 712-716.
25. Zgaya, H. (2007) *Conception et optimisation distribuée d'un système d'information d'aide à la mobilité urbaine : Une approche multi-agent pour la recherche et la composition des services liés au transport*. PhD thesis. University of Lille, France, 240 p.
26. Zhang, J., Arentze, T., Timmermans, H. (2012) A multimodal transport network model for advanced traveler information system, *Journal of Ubiquitous Systems & Pervasive Networks*, 4, 21-27.
27. Zhang, L., Li, J.Q., Zhou, K. (2011) Design and implementation of a traveler information tool with integrated real-time transit information and multi-modal trip planning. *Transportation Research Board Annual Meeting, Washington, D. C.*
28. Zidi, K. (2006) *Système Interactif d'Aide au Déplacement Multimodal (SIADM)*. PhD thesis, University of Lille, France, 134 p.