Distributed Graphs for Solving Co-modal Transport Problems
Jeribi Karama, Mejri Hinda, Zgaya Hayfa, Hammadi Slim

To cite this version:
Distributed Graphs for Solving Co-modal Transport Problems

Karama Jeribi, Hinda Mejri, Hayfa Zgaya and Slim Hammadi

Abstract—The paper presents a new approach based on a special distributed graphs in order to solve co-modal transport problems. The co-modal transport system consists on combining different transport modes effectively in terms of economic, environmental, service and financial efficiency, etc. However, the problem is that these systems must deal with different distributed information sources stored in different locations and provided by different public and private companies. In order to resolve the co-modal transport problems, we propose a distributed co-modal approach based on special distributed graphs that adapts to the distributed nature of real world transport information.

I. INTRODUCTION

Co-modal transport approach, almost like its predecessor “the multimodal transport approach”, refers to the “use of different modes on their own or in combination”. It consists on developing infrastructures and taking measures that will ensure optimum combination of individual transport modes enabling them to be combined effectively in terms of economic efficiency (i.e. providing the most cost effective combination), environmental efficiency (the least polluting combination), service efficiency (level of service provided), financial efficiency (best use of society’s resources), etc [1]. With this approach, we don’t seek anymore to oppose transport modes one to another but rather to find an optimum exploiting the domains of relevance of the various transport modes and their combinations.

Since its introduction by the European commission in 2006 [2], the new notion of co-modality started to interest researchers while many research projects were devoted to develop multimodal transport systems that recommend to travelers a combination of transport means door-to-door (i.e. providing the most cost effective combination), environmental efficiency (the least polluting combination), service efficiency (level of service provided), financial efficiency (best use of society’s resources), etc [1]. With this approach, we don’t seek anymore to oppose transport modes one to another but rather to find an optimum exploiting the domains of relevance of the various transport modes and their combinations.

The remainder of this paper is structured as follows. The distributed co-modal approach is presented in the section 3 with some definitions and presentation of the transfer graph model. Then, the adopted algorithm DSRA in order to solve the transfer graph is presented in the section 4. The section 5 presents an example that illustrates the adopted approach. Finally, conclusion and prospects are addressed in section 6.

II. CONTEXT DESCRIPTION

The majority of the national implemented transport’s information systems are mono-modal dealing only one means of transport. In fact, the majority of the transport’s operator websites offer just one service like public transport, carpooling… However, the main concern of our system is to satisfy the users by offering different complementary transport services, providing their itineraries and respecting their priorities criteria.

At a time t, many transport users can formulate simultaneously a set of requests. Each transport user uses a medium of communication (e.g. laptop, PDA) in order to express his demand and provide a departure and arrival points and the correspondent earlier and later schedules. So the system should find feasible decompositions in terms of independent sub-itineraries called Routes recognizing similarities. For a given Route, we can have several possibilities with different vehicles, which are available to ensure this Route through the same time window. All these identified Routes constitute our co-modal graph and we have to recognize the different possibilities of Routes Combinations to compose each itinerary demand. The problem is how to choose the most effective Route Combination to a given user, taking into account his constraints and preferences in terms of total cost, total travelling time and total greenhouse gas volume.

Our problem is defined by:

- $N$ itineraries demands formulated through a short interval of time $\Delta_k$. $I_k$ is the set of these requests.
- $I_{k,t}(d_k, a_k, W_k) \in I_k$ is an itinerary request formulated by a user $k$ at a time $t$ from a departure point $d_k$ to an arrival point $a_k$ through a time window $W_k = [td_k, ta_k]$; $td_k$ and $ta_k$ correspond respectively to the most earlier (minimum departure time from $d_k$) and the most later (maximum arrival time to $a_k$) possible schedules with $t \leq td_k < ta_k$;

Manuscript received April 10, 2011. This work was supported in part by the ADEME Agence de l’Environnement et de la Maîtrise de l’Énergie and regional council Nord Pas-de-Calais.

Karama Jeribi and Slim Hammadi are with the Laboratoire d’Automatique, Génie Informatique et Signal –LAGIS FRE CNRS 3303, Ecole Centrale de Lille, Villeneuve d’Ascq 59851 France (e-mail: karama.jeribi@ec-lille.fr; slim.hammadi@ec-lille.fr)

Hinda Mejri is with Faculté de Droit et des Sciences Economiques et Politiques de Sousse –Cité Errriadh 4000 Sousse Tunisie (e-mail: hindamejri@yahoo.fr).

Hayfa Zgaya is with the EA2694 Laboratory of Public health, Institut Lillois D’Ingénierie de la Santé, 42, Rue Ambroise Paré 59120, LOOS, France (e-mail: hayfa.zgaya@univ-lille2.fr)
• $RC_{k,t} = \{RC_{k,t,p}, p \in [1..P]\}$ is the set of all possible Route combinations identified to answer to the request $I_{k,t}(d_k,a_k,W_k) \in I_t$. $P$ is the total number of these Route Combinations.
• $R_g(d_g,a_g,W_g)$ is a Route identified to response to one or many itinerary requests $I_{k,t} \in I_t$. A junction or a succession of different routes $R_g(d_g,a_g,W_g)$ composes a possible solution for the request. $RC_{k,t,p}$ is a possible Route Combination identified to respond the request $I_{k,t}(d_k,a_k,W_k) \in I_t$.
• For one Route $R_g(d_g,a_g,W_g)$, we need a mean of transport available to move from the departure point $d_g$ to the arrival point $a_g$ through a time window $W_g = [t_d,g,t_a,g]$ with $t_d,g$ and $t_a,g$ correspond respectively to the most possible earlier departure time to leave $d_g$ and the most possible later arrival time to attend $a_g$.
• $R_t$ is the set of all identified Routes to response to $I_t$

Our system is a co-modal system that combines different means or transport services like the public transport service, carpooling and free use vehicle services. So, we are in front of a co-modal transport problem and our Distributed Co-modal Approach (DCA) is described in the next section.

### III. DISTRIBUTED CO-MODAL APPROACH (DCA)

In this section, the distributed co-modal approach is presented. The based steps of the approach are described in the Fig. 1.

![Co-modal Graph](image)

**Fig. 1 Steps of the distributed co-modal approach**

The co-modal graph is obtained by collecting all the feasible and possible routes that could response to the requests.

From the co-modal graph constructed, a transfer graph is deduced. This graph model was adopted for solving time-dependent multimodal transport problems [7]. This model corresponds to our needs since it adapts to the distributed nature of real word transport information sources and allows updates on each transport network independently. From the transfer graph, the first shortest paths can be calculated and obtain then the final graph.

### A. Definitions

Let $G = (N,E,M)$ denotes a co-modal graph or co-modal network, where $N = \{n_1, ..., n_l\}$ is a set of vertices, $E$ is the total number of vertices, $E = \{e_1, ..., e_L\}$ is a set of edges, $L$ is the total number of edges and $M = \{m_1, ..., m_k\}$ is a set of transport services (e.g. Public transport, Carsharing or Carpooling), $K$ is the total number of transport services. An edge $e_i \in E$ with $i \in \{1..L\}$ can be identified by $(n_{p,i},n_{q,i},m_{r,i})$, where $n_{p,i},n_{q,i} \in N$ and $m_{r,i} \in M$ with $p,q \in \{1...J\}$ and $r \in \{1..K\}$. The $e_i$ expresses that it is possible to go from vertex $n_{p,i}$ to $n_{q,i}$ by using transport service $m_{r,i}$. A value $D_{n_i} = D(n_{p,i},n_{q,i})$ is associated to each edge $e_i$, indicating the weight and the cost of including the edge in the solution.

**Definition 1 (Co-modal path):** Given a co-modal graph $G = (N,E,M)$, a path or a route combination $RC_{n_1,n_l} = (n_1 \rightarrow n_l)$ is a sequence of edges between a pair of vertices $n_1$ and $n_l$ with $((n_{1,1},n_{1,2},m_{1,1}), (n_{1,2},n_{1,3},m_{2,1}), ..., (n_{l-1,1},n_{l,1},m_{l,1}))$ where $\forall i \in \{1,...,l\}, n_j \in N, \{n_{p,i},n_{q,i}\} \in E, m_i \in M$ and $i \in \{1..K\}$.

So, a path $RC_{n_1,n_l} = (e_1,e_2,...,e_l)$ is said to be co-modal if $\exists e_p,e_q \in E$ with $e_p = (n_{p,p},p,m_{p}), e_q = (n_{q,q},q,m_{q}), m_{i,p} \neq m_{i,q}, i \neq j$ and $i,j \in \{1..L\}$. If there is only one service involved in the route combination, the route combination is said to be uni-modal or uni-service.

**Definition 2:** Given a route combination $RC_{n_2,n_l} = (n_1 \rightarrow n_j)$ or an edge $(n_{p,i},n_{q,i},m_{i})$, $i,j \in \{1..L\}$ and $i \in \{1..K\}$, a time window is defined as a time interval $[t_{n_1},t_{n_j}]$ where $t_{n_i}$ denotes the departure time from vertex $n_i$ and $t_{n_j}$ the arrival time at $n_j$.

**Definition 3 (Dynamic co-modal graph):** Since each edge represents a route assured by a transport service, the cost of edges is considered to be time-dependent. $\forall e_i \in E$ we can have $D_{t_{n_1}}(t_1) \neq D_{t_{n_2}}(t_2)$. Our graph $G = (N,E,M)$ becomes a dynamic co-modal graph.

### B. Transfer Graph

The approach of transfer graph was proposed in order to solve a time-dependent multimodal transport problems while a transfer graph is composed of a set of uni-modal graphs[7]. In our case, we adopt this approach with a transfer graph described by a set of uni-service networks and a set of arcs connecting them.

It is defined by $G_T = (C, TR)$ where $C = \{C_1, C_2, ..., C_k\}$ is the set of uni-service networks called components and $TR$ is the set of virtual edges which interconnect them. Each component $C_i = (N_i, E_i, M_i, PTC_i)$ is such that $\forall i,j \in \{1,...,K\}, M_i \neq M_j$. Besides, $N = \bigcup_{i\in\{1,...,K\}} N_i$, $E = \bigcup_{i\in\{1,...,K\}} E_i$, $M = \bigcup_{i\in\{1,...,K\}} M_i$ and $TR = \{ (n_i,n_j) \text{ such as } n_i \in C_i, n_j \in C_j, n_i = n_j \}$ where $(n_i,n_j)$ represents a transfer from service transport $m_i$ to another service transport $m_j (m_i,m_j) \in M$ at the co-modal transfer point $n_i$ (or $n_j$). $n_i \in PTC_i$, $n_j \in PTC_j$, $i,j \in \{1..K\}$ are called Co-modal Transfer Points and symbolized.
the same location. So, we have $PTC_i = \{ n_i \in C_i | \exists n_j \in C_i$ with $n_i = n_j, i, j \in \{1, ..., K\}$.

The Fig. 2 illustrates an example of a transfer graph where $C_1, C_2$ and $C_3$ are components connected by four transfers, and vertices a, c, b and d are Co-modal Transfer Points. Each component contains edge belonging to only one transport service. In this example, we can go from $d_k$ to $a_k$ using only the public transport $RC_{d_k,a_k} = (d_k, c)_{c_1}, (c, b)_{c_2}, (b, a_k)_{c_3}$. Another possibility is the Route combination $RC_{d_k,a_k} = (d_k, d)_{c_2}, (d, e)_{c_3}, (e, a_k)_{c_3}$.

![Fig. 2 Example of transfer graph](image)

In the transfer graph, we distinct two path’s types: inter-components and intra-components. An inter-component path is considered as any path which connects two vertices $x, y \in N$, where at least two edges belong to two distinct components. However, an intra-component path with $C_i$ is a path which connects two vertices $x, y \in N_i$ whose edges belong to only one component. It is possible to have several route combinations $RC_{x,y}$ which connect $x$ and $y$.

An intra-component can be one of the following categories:

- $RC_{d_k,a_k}^i$ is the shortest path which starts at source vertex $d_k$ and ends at target vertex $a_k$ within $C_i$.
- $RC_{d_k,PTC}^i$ is the set of shortest paths which start at source vertex $d_k$ and end at a Co-modal Transfer point $PTC_i$ within $C_i$.
- $RC_{PTC,PTC}^i$ is the set of shortest paths which start at any Co-modal Transfer Point $PTC_i$ and end at $PTC_j$ within $C_i$.
- $RC_{PTC,PTC}^i$ is the set of shortest paths which start at any Co-modal Transfer point $PTC_i$ and end at target vertex $a_k$ within $C_i, i \in C$.

Given to the transfer graph $G_T = (C, TR)$, we have to compute for all components $C_i \in C$ the different intra-component paths: $RC_{d_k,a_k}^i, RC_{d_k,PTC}^i, RC_{PTC,PTC}^i, RC_{PTC,PTC}^i$. The approach adopted to compute will be explained in the section 4.

C. Final Graph

After the computation of shortest intra-component paths $RC_{d_k,a_k}^i, RC_{d_k,PTC}^i, RC_{PTC,PTC}^i, RC_{PTC,PTC}^i$, (section 4), we can deduce the final graph.

**Definition 4 (Final graph):** Given a transfer graph $G_T = (C, TR)$, we define a final graph as $G_F = (N_f, E_f)$, where $N_f = \bigcup_{C_i \in C} PTC_i \cup \{d_k, a_k\}$ and $E_f = \bigcup_{C_i \in C} RC_{d_k,a_k}^i \bigcup RC_{d_k,PTC}^i \bigcup RC_{PTC,PTC}^i \bigcup RC_{PTC,PTC}^i \bigcup RC_{PTC,PTC}^i \forall C_i \in C$.

IV. SOLVING THE TRANSFER GRAPH

The system has to compute the shortest paths $RC_{d_k,a_k}^i, RC_{d_k,PTC}^i, RC_{PTC,PTC}^i, RC_{PTC,PTC}^i, RC_{PTC,PTC}^i$ for all the component $C_i \in C$. Each component $C_i$ represents a transport service and can be provided thanks to different operators. Each Transport Operator’s Information System (TOIS) is composed of a local database (DB) describing the different means of available transport that it manage and of an Itinerary Calculating Algorithm (ICA) which uses these local data to search optimal itineraries for users requests. To integrate the co-modal information from the different heterogeneous transport operator’s information systems (TOIS), we proceed to applications integration. This integration tries to take advantage of the current multi-modal or mono-modal information systems and make the TOIS cooperate to calculate multi-modal route.

In order to adapt our system to the distributed nature of real world transport information sources, we decompose each service or each component $C_i = (N_i, E_i, M_i, PTC_i)$ into different classes $C_{ij}$ with $1 \leq j \leq K_i$, $K_i$ is the total number of operators related to the component $C_i$.

**Definition 5:** A class in a distributed system refers to an autonomous subsystem. A class possesses its independent resources[8].

Each class $C_{ij}$ is represented by a graph $G_{ij}(N_{ij}, E_{ij})$ with $N_{ij}$ and $E_{ij}$ are respectively the set of vertices and edges related to the operator $j$ of the service $C_i$. A vertex $n_{ij}$ can be even a public transport’s station, carsharing station or a departure or arrival point for a carpooling service. Also, an edge $e_{ij}$ represents a route using a transport mode managed by the operator $C_{ij}$ with $1 \leq j \leq K_i$.

The graph of a component $C_i = (N_i, E_i, M_i, PTC_i)$ represents the aggregate structure of the distributed system. Typically, as transport networks they represent, the graphs $G_{ij}$ overlap at some places, which are called intersections. The vertices in intersections are the transferring stops at which travelers finish their tours with one operator and start with another.

The graph $C_i = (N_i, E_i, M_i, PTC_i)$ is a supergraph that allows more than one edge between a pair of vertices. For example, if $x$ and $y$ are intersections of class $C_{ij}$ and $C_{ik}$ (i.e., $x \in N_{ij}, x \in N_{ik}, y \in N_{ij}, y \in N_{ik}$), $(x, y)_{ij} \in E_{ij}$ and $(x, y)_{ik} \in E_{ik}$, then there are two edges $(x, y)_{ij}$ and $(x, y)_{ik}$ between vertices $x$ and $y$ in...
\[ C_i = (N_i, E_i, M_i, PTC_i) \] and each edge has a label \( D((x,y),i,j) \) and \( D((x,y),i,k) \), respectively.

Since the component represented by
\[ C_i = (N_i, E_i, M_i, PTC_i) \] is a distributed system in which local classes maintain their own data and there is not an aggregate central database, \( C_i = (N_i, E_i, M_i, PTC_i) \) is actually a virtual graph and not stored with the central computing server. The distributed shortest path problem is defined as a problem for the central computing server to find the shortest inter-class route between any two vertices in \( C_i = (N_i, E_i, M_i, PTC_i) \) based on some local information provided by individual classes.

Some researchers proposed for the distributed shortest path an approach based on an intersection graph[8][9][10].

**A. The Intersection Graph**

The following procedure formally defines the complete intersection graph \( G_{int} \).

**Definition 6:** Let \( G_{int} \) denotes the intersection graph. A vertex \( n \in G_{int} \) if and only if \( n \) is an intersection vertex in \( (N_i, E_i, M_i, PTC_i) \). There is an edge between vertices \( m \) and \( n \) in \( G_{int} \) if and only if \( m \) and \( n \) are clannish.

Let \( RC_{m,n}^{i,j} \) be the length of the shortest one of the routes linking \( m \) to \( n \) in class \( C_i,j \). Also, let \( I(m,n) = \{ j \mid m,n \in C_i,j \} \) denote an index set of classes containing both vertices \( m \) and \( n \).

Each edge \((n,m)\) has two labels, \( RC_{m,n}^{i} \) and \( rec(m,n) \), so that:

\[ RC_{m,n}^{i} = \min_{i \in I(m,n)} (RC_{m,n}^{i,i}) \text{ and } rec(m,n) = C_j \text{ if } RC_{m,n}^{i} = RC_{m,n}^{i,j} \text{ for some } j \in I(m,n). \]

Label \( RC_{m,n}^{i} \) on edge \((m,n)\) in \( G_{int} \) represents the shortest distance between vertices \( m \) and \( n \) by using the resource of only one class. \( RC_{m,n}^{i} \) is the shortest intra-class route between \( m \) and \( n \). Label \( rec(m,n) \) indicates the class associated with \( RC_{m,n}^{i} \).

**B. The Distributed Shortest Route Algorithm**

To compute the shortest route between \( d_k \) and \( a_k \) in the time window \( W_k \), we need to extend the complete intersection graph by including \( d_k \) and \( a_k \) in the graph. This graph is called the virtual extended intersection graph, denoted as \( G_{veint}(d_k,a_k,W_k) \).

**Definition 7:** Let the virtual extended intersection graph \( G_{veint}(d_k,a_k,W_k) \) contains all vertices of \( G_{int} \) plus the departure and arrival vertices \( d_k \) and \( a_k \) (if they are not in \( G_{int} \)).

\[ G_{veint}(d_k,a_k,W_k) \] contains all the edges of \( G_{int} \) plus the following: an edge between \( d_k \) and each of its clannish vertices and an edge between \( a_k \) and each of its clannish vertices.

The shortest path on the extended virtual complete intersection graph can be calculated by dijkstra’s algorithm. The distributed Shortest Route Algorithm using all the definitions described below, is described as following.

**Distributed Shortest Route Algorithm (DSRA)**

1. **Step 1:** Construct the complete intersection graph \( G_{int}(d_k,a_k,W_k) \).
2. **Step 2:** Construct the extended virtual complete intersection graph \( G_{veint}(d_k,a_k,W_k) \).

**Step 3:** Compute the shortest route \( RC_{d_k,a_k} = \min_{d_k,a_k} \) using a Shortest Route Algorithm (SRA).

**Step 4:** For each pair of vertices on the shortest route \( RC_{d_k,a_k} = \min_{d_k,a_k} \), call relative class to the edge \((n_k,n_{k+1})\) to fill in the details of the intra-class route associated.

We consider that:

\( \varphi_{d_k}(d_k) \) denotes the earliest arrival time to the vertex \( j \) leaving from the depart vertex \( d_k \) at the time \( t_d_k \).

\( pred_{d_k}(j) \) defines the predecessor vertex \( j \) at \( t_d_k \).

\( N^+(j) \) defines the set of successors of \( j \).

The algorithm that computes the shortest route \( RC_{d_k,a_k} \) is described as following:

**Shortest Route Algorithm (SRA)**

1. **Initialization**
   \[ \varphi_{d_k}(d_k) = t_d_k \]
   \[ \varphi_{d_k}(d_k) = \infty \text{, } pred_{d_k}(d_k) = \infty, \text{ and } rec(d_k) = \infty \]

2. **Node Selection**
   Let the node \( l \) with \( min_{j} \in N_{current}(\varphi_{d_k}(d_k)) \)

3. **Exploration of possible successors**
   \[ \forall j \in N^+(l) \text{ do} \]
   \[ \text{If } (\varphi_{d_k}(d_k) < \varphi_{j}(\varphi_{d_k}(d_k))) \text{ and } (\varphi_{d_k}(d_k) < t_d_k) \]
   \[ \text{Then } \]
   \[ \varphi_{d_k}(d_k) = \varphi_{j}(\varphi_{d_k}(d_k)) \]
   \[ pred_{d_k}(j) = l \]
   \[ rec(j) = \text{Class Index} \]

4. **If** \( N_{current} = \emptyset \text{ alor }s N_{current} = \{j\} \cup N_{current} \text{ go to step 2.} \]

After the shortest paths in term of time, the final graph is generated. The identification of the best Route combination with an evolutionary approach is not handled in this paper [11][12].

**V. SIMULATION EXAMPLE**

In order to explain and evaluate the solution proposed in this paper and validates the distributed co-modal approach for the vehicle sharing services system we applied the methodology proposed on an academic example for transport requests.

We are developing our system, with JADE platform (Java Agent Development platform) [13]. JADE is a middleware which permits a flexible implementation of multi-agents systems; it offers an efficient transport of ACL (Agent Communication Language) messages for agents communication which complies with FIPA specifications [14]. Also, we connect our system to a geo-localization...
software “Cartocom”. This software is able to communicate with mobile phones and locate them in real time. However, for this example, we use the Cartocom as a software of simulation where we can see the responses of our system to the user’s request.

We consider four itinerary requests at t=7h15:
- I₁₅(t, A, B, [7h15, 8h30]) : does not like public transport with Criteria weights (0.5, 0.3, 0.2);
- I₂₅(t, C, D, [7h40, 8h30]) : cannot drive a bicycle with Criteria weights (0.2, 0.8, 0);
- I₃₅(t, A, E, [7h15, 7h55]) : does not like carpooling with Criteria weights (0.5, 0.2, 0.3);
- I₄₅(t, A, F, [7h15, 7h40]) : nothing to announce with Criteria weights (0.2, 0, 0.8).

With Cartocom we can locate the departure and arrival points of the different itinerary requests in the map as shown in the Fig. 3.

From the identification of all possible routes going from A in direction to B, C in direction to D, A in direction to E or A in direction to F, we can construct the co-modal graph. Then, the transfer graph is constructed. The graph is composed of three components $C = \{C_1, C_2, C_3\}$ with $C_1$ represents the multimodal public transport service, $C_2$ represents the carpooling service and finally $C_3$ correspond to the free use vehicle.

In this example, we obtained four Co-modal transfer Points \{o, s, m, n\} as shown in the Fig. 4.

The next step is to compute in each component $C_i \in C$ all the $RC_{d_k, a_k}^{\ast i}$, $RC_{d_k, PTCL_i}^{\ast i}$, $RC_{PTCL_i, PTCL_j}^{\ast i}$, $RC_{PTCL_j, a_k}^{\ast i}$ for each itinerary request $I_{ki}$ thanks to the Distributed Shortest Route Algorithm (DSRA).

We detail the application of this algorithm in the first component $C_1$. The graph related to the Public transport service at the time t=7h15 is presented in the Fig. 5.

The application of the DSRA in order to compute $RC_{A,B}^{\ast 1}$ begins by constructing the extended virtual intersection graph $G_{vint}(A, B, W_{A,B})$ presented in Fig. 6. Then, we use the SRA in order to find the shortest path. So, the obtained result is $RC_{A,B}^{\ast 1} = (A, F, n, B)$.

For each pair of nodes in this path, we determine the corresponding class and the detail of the path (Fig. 7).
We continue in the same way the identification of all the rest of shortest Routes in the three components. So, the system provides to each transport demander a set of optimized itineraries taking account of his preferences. The Fig. 8 shows the results thanks to the Cartocom software.

For each transport demander, the system responds to his request and presents to him the best route combination (Fig. 9). The assignment of the users and the identification of best route combination is described in others works [11][12].

VI. CONCLUSION

In this paper, we proposed a Distributed Co-modal Approach (DCA) for solving co-modal transport problems. The contribution of this paper is the adoption of the transfer graph and its resolution thanks to the Distributed Shortest Route Algorithm applied in each component of the transfer graph in order to compute shortest paths intra-component in a distributed system. For this part, we can extend our research as a future works into a shortest path research in a distributed dynamic stochastic environment. In fact, the stochastic graphs may be interesting in case of perturbation.

REFERENCES

[14] www.fipa.org