NNGT Int. J. on Information Security, Vol.5, February, 2016

Exact N-modal Itineraries Generation for Ad Hoc Interconnection of Transportation Networks

Idriss Hassine and Philippe Canalda
Institut FEMTO-ST (UMR CNRS 6174)
Université de Bourgogne Franche-Comté
Multimedia Development Center NUMERICA
1, Cours Louis Leprince-Ringuet, 25200 Montbéliard, France
Email: idriss.hassine.2015@ieee.org, philippe.canalda@femto-st.fr

***Abstract*—CIMO is a combinatorial system which computes optimal multimodal itineraries consisting in itineraries which are sorted, multimodal and trans-territorial. In this work we propose the formalism of ad hoc multimodal itinerary problematic, the multi-constraints based dynamic programming algorithm approach, and a realistic evaluation of the solution proposed which address new aggregation of operational transportation territories. The solution is based on a dynamic programming algorithm ”cut”, ”price” and ”share”. This solution is multi-objectives and multi-constraints. Progressive versions of this algorithm are proposed following a methodological approach that enables evaluation of efficiency and complexity's gain. Test benchmarks are run to validate the contributions, until the overall system shows its capacity to propose multimodal itineraries for real metropolis area. Other parameters are defined as the speed up and the relative gain. This work also provides a set of evaluations of CIMO's versions: from a real illustration based on two extractions transport networks with small overlap (bus) and connected transversely by two other modes (train and cars), to a simulation of territorial areas comprising the territories served by 3-modes transport services (bus, car and train) of AOTS prefectures and sub-prefectures (2 x 800 stations, 2 x 20 lines, journey time-tabling's ranging from 6 am to 20 pm, lines of trains and buses serving longitudinally these 2 networks). This study shows that an optimal itinerary can be calculated with an exact, dynamic programming algorithm ”cut” and ”price” and ”share” which generates 3-modes itinerary over two connected transportation networks, deserving 300,000 inhabitants.**

I. A STEP TOWARDS A TRANS-TERRITORIAL AND MULTIMODAL ITINERARY CALCULATOR...

The on-the-fly generation of optimal multimodal itineraries is a very complex issue that arouses great interest from the scientific community and also from the Authoritative Transport Organization's commmunity (ATOs for short). Nowadays, innovative operational systems tackle, dynamically, the greatest numbers of multimodal arrangements (bus, train, triggered transport, carpooling, carsharing, and soft modalities such as walking or bike-sharing). They also face the new aggregation of territories, wherin ATOs must collaborate. They must offer a service that takes into account dynamic requests and very versatile information (advance/delay aleas, forecasting congestion, run-time application of itineraries,...).

A. Living territories, new mobilities, and innovative mobility-help services

We travel to work, to study, for leisure and for shopping. We travel alone or accompanied. Our movements are dependent on points of interest such as home and commercial centers. They are dependent on resources (finance and transportation), and also public and collective transport options serving human living area (waypoints, parking).
To address the need for mobility in urban areas or elsewhere, we use several modes of transportation: from private vehicle, public and collective transportation, to bike or walk on foot. Among the constraints of our movements, the principal is to be on time for an appointment. In order to satisfy this constraint we must estimate the best times to exit our home and the best itinerary to be followed to arrive at destination on time.

In this study, we address the opportunistic use of public transportion. It comes to solve the problem of multimodal itinerary calculations from offers arising from various ATOs. The trans-territorial movements using public transport are becoming commonplace within a modern society in which family members work in various places, and where urban and sub-urban areas federate their life way-points, and transportation services.

B. An illustrative example extracted from the real urban area of Belfort-Montbéliard (AUBM)

AUBM's area is composed of three major cities: Belfort, Montbéliard and Héricourt. The whole area hosts 300,000 inhabitants. It belongs to the Franche-Comté region.

The transport authorities of the urban area are diverse and they have difficulties in deploying multimodal itinerary calculator. By only considering here three modes of transportation, not less than 6 ATOs are involved in: the Transport Company of Montbéliard (CTPM) which delegates the exploitation to Keolis, the Mixte Syndicate of Public Transportation (SMTC) which delegates the exploitation to the Optymo network, the Mixte Syndicate of AUBM's Wide Area (SMAU), which coordinates the global mobility, the SNCF (part of Keolis) which operates train network under the controle and authority of Franche-Comté region and Alsace's one, and finally Illicom platform that provides information on the cross-department

DOI : 05.IJIS.2016.1.1

mobility (regional and inter-regional train, CTPM and Opymo buses, transportation on-demand and bike circuits).
From these statements, we understand that a transport user has to manipulate several information systems. Thus, the data management is an issue that belongs to many ATOs and transportation operators, with their own and specific equipments and information systems.

A master student of Université de Bourgogne Franche-Comté[1], who lives in Belfort and follows courses located over 2 sites, the Technological University of Belfort-Montbéliard (UTBM - Belfort site) and the University of Franche-Comté (UFC - Montbéliard site) is forced to solicitate these ATOs to achieve his home-to-work mobilities. The student lives close to the "Haut de Belfort" bus' station. He has to join, 3 mornings a week, the "Donzelot station" which deserves the Montbéliard university site. Delivered courses start at 8h30 a.m.. Obviously, a student has to be present a slightly before this time. Practically, the first step is to take the bus (Optymo network) at "Haut-de-Belfort" station to the Belfort train station. Then, in a second step, he must take a train (SNCF) from Belfort Train Station to Montbéliard station through Héricourt. Arriving at Montbéliard train station, he must take another bus (network CTPM) to arrive at Donzelot, the nearest station to the university. His multimodal itinerary is at least equal to three. Numerous lines deserve the train stations, the student residence and the university, varying by frequency or by huge and specified time-tabling.

C. On the ground, operators must exchange their transport network data and mobility services

Currently, there are 3 operating systems for calculating itineraries. The Information System (IS) of the CTPM calculates itineraries only on its bus network, ditto for the Optymo, but for Illicom platform, which is a system of services provided jointly by CTPM, Keolis and Montbéliard Agglomration, it treats the bus network of Montbéliard, the SNCF train lines between Strasbourg, Delles and Lyon, through Belfort, Héricourt and Montbéliard Train Stations. Car circuits are operated by the departments "Territoire de Belfort" and "Doubs", to complete the main tri-modal network that contribute joining the two cities. Last but not least, Belfort and Montbéliard, distant of 16 km, are accessible through the A36 motorway, or by bike-ways.

The conclusion is that the use of these systems provides results that are neither consistent nor dynamic. Moreover, there is no IS offering simultaneously all modes of transport in the emerging metropolitan urban area of Belfort-Montbéliard.

D. The underlying scientific problem is never completely formalized, and multimodal itinerary calculators are facing increasing complexity of data (heterogeneous modes, transport lines, timetables, scattered data)

The research and the latest innovations in ITS (Intelligent Transport System) aimed at developping multimodal itineraries mainly in two aspects, firstly in a static context, and secondly in a dynamic context [2], [5], [13]. In the static context, dynamic risks are rarely apprehended [13] (advances, delays, cancellation, the high-level, failure, triggered new race), and combinations of modalities considered (buses, trains, car, subway, bicycle, transportation-on-demand, carpooling, walking buses, parking, ...) vary from one transport system to another ([5] = Park-n-Ride, [12] = Bus + Train + Plane), and the respective characteristics of the modalities (number of lines, number of trips, number of stations, timetables by station and lines), and the natures of lines (regular or triggered, with partial or global "*cabotage*", with or without correspondences).
In the dynamic context, other factors complexify the problem : the nature of the risks considered and their sizes, the frequency of information refreshment, the speed of calculation and / or the provision of satisfactory operating itineraries to supply constraints and demand constraints, the quality of information (confidence levels, life expectancy, robustness).
Transverse to the static and dynamic contexts, many other characteristics complicate the formalization of the problem, and influence combinatorial techniques and solutions implemented, as: the properties of field graphs representing road infrastructure, parking spaces and modal transfers, parking spaces or intermodal exchange, modeling modes of transport and their connections.
Today the interconnection of operating systems, and their aggregated areas, the scale of mobility considered at international level [8], [9], at the level of interconnecting the cities [3], [4], at the level of connecting towns and "outline" networks, locally with its great diversity of relevant modalities (soft and triggered transport modes), all of this constitutes much formal and experimental challenges, where difficulties of access sparsed data and specialized demand from users are not the least of pitfalls. The interconnection of local transport networks are not addressed pragmatically [2] and no known theoretical solution are deployed and operated which address, realistically, the numerous and practiced modalities, and which consider the actual need of users' mobility (new metropolitan poles, communities of communes, evolving departments or regions).

Among major algorithmic solutions, we mainly consider exact approaches based on the calculation of k-best itineraries for a pair (origin, destination), taking into account [14], or not, time window and timetables of means of transport [5], [6]. Other works applied the Branch and Bound and the ant colonies techniques [15], either based on enumerative approach [16], or parallel's one. These works treat the bus, take into account space optimization related to the transport request [10]. In [14], if the time windows of the application are rarely included, with static timetables Pick-Up or Delivery at stations, the maximum number of modalities reports in an itinerary are less or equal than 2.

In [18] Christian Artigues follows another approach to render more efficient the enumerative approach. He uses an automata-based approach by using four modes of transporta-

---

[1] This work was partially funded by the University of Bourgogne-Franche-Comté, the University of Haute-Alsace and the Technological University of Belfort-Montbéliard. These three universities are involved in the master 2 Mobile and Distributed Computing.

[2] Outside holdings of greatest metropolitan poles as l'le de France, Grand Toulouse, Grand Lyon, Bombay [4], Sydney [14], [7], ...

DOI : 05.IJIS.2016.1.1

tion: Bus, car, foot and subway. A set of constraints on itineraries are defined: for example the private car can only be used once or subway can be only used once during the trip. Although these constraints affect strongly the simplification of problem complexity, they are not relevant for the itineraries involved round trip. These constraints remain applicable or only limited in the context of big cities, they are not relevant to new cities. In [19] the automata-based approach is used by incorporating the round trip's notion in exploring possible itineraries which presents the strength of this work in the field of intelligent transport systems. However, the itinerary calculation is mainly based on shortest itinerary's criteria in terms of distance, itinerary cost was not expanded to include number of correspondences, the effective transport time, the travel time...

In the remainder of this review which is an extended version of [1], we will first characterize the input data of the problem. We will formalize the multi-objective problem and corresponding multi-constraints. Then we will present the dynamic programming algorithm Cut&Price&Share by adopting an incremental approach. A series of detailed tests was performed on a real illustartion extracted from the urban area Belfort Montbeliard explain the principle of each version .A series of evaluations will allow to evaluate the *Cut* conditions efficiency, the performance of the shares of common sub-calculations, accelerating the termination of the itinerary generator. A focus will be put on the influence of using ordered stations. We will describe the test generator. We will conclude with an analysis of the proposed approach and we will discuss the future works.

II. HOW TO FORMALIZE THE PROBLEM OF OPTIMAL ITINERARY GENERATION WHEN CONSIDERING REALISTIC N-MODAL AND TRANS-TERRITORIES ITINERARY?

We introduce the CIMO system (Calculateur d'Itinéraires Multimodaux Ordonnés) which is calibrated to address a combination of three-to-four networks (OPTYMO, CTPM, Keolis, and Illicom) for modes[3] among bus, train and car.

A. Extracting a 3-modal Network of AUBM

We modeled our first extraction of the experimentation territory by a prototype of 9 stations distributed as follows:

- 3 bus stations in Belfort: *"Multiplexe Belfort", "Foch Belfort", et "Liberté Madrid Belfort".*
- *3 bus stations in Montbéliard: "Temple", "Jean Moulin", et "Place Ferrer".*
- *3 train stations between Belfort and Montbéliard : "Belfort Gare", "Hricourt", and "Montbéliard Gare".*

In this model it is assumed that the time window is defined as a sub-range from 6 a.m. to 8.30 a.m.

3 lines are extracted from the OPTYMO BUS' network of Belfort:

- OPTYMO bus' line number 1:
  "Foch Belfort" → "Gare Belfort"
  "Gare Belfort" → "Foch Belfort"
- OPTYMO bus' line number 2:
  "Multiplexe" → "Liberté Madrid"
  "Liberté Madrid" → "Multiplexe Belfort"
- OPTYMO bus' line number 3 :
  "Multiplexe Belfort" → "Gare Belfort"
  "Gare Belfort" → "Multiplexe Belfort"

4 lines are extracted from the CTPM BUS' network of Montbéliard:

- CTPM bus' line number 1:
  "Gare Montbéliard" → "Temple"
  "Temple" → "Gare Montbéliard"
- CTPM bus' line number 2:
  "Place Ferrer" → "Temple"
  "Temple" → "Gare Montbéliard"
- CTPM bus' line number 3:
  "Gare Montbéliard" → "Place Ferrer"
  "Place Ferrer" → "Gare Montbéliard"
- CTPM bus' line number 4:
  "Place Ferrer" → "Jean Moulin"
  "Jean Moulin" → "Place Ferrer"

For the SNCF TRAIN' network one line was extracted between Belfort and Montbéliard:

- *"Gare Belfort" → "Gare Montbéliard": Ter95860, Ter94008, Ter94832 et Ter94014.*
- *"Gare Montbéliard" → "Gare Belfort": Ter94003, Ter94007, Ter95863, Ter94833 et Ter94103.*

B. The internal data and data exchange interface

The set of all the lines of transport of the considered network:
$LigneTransport = (LT_1, LT_2, , \ldots, LT_j, \ldots, LT_{lt})$
with $|LigneTransport| = lt \in \mathbb{N}$
CIMO returns an ordered list of itineraries:
$Itins = \{Itn_1, Itn_2, \ldots, Itn_{It}\}$
with $|Itins| = It \in \mathbb{N}$
The set of all the stations in the considered transport sub-network: $S_{Ts} = (S_1, S_2, \ldots, S_s)$ with $|S_{Ts}| = s \in \mathbb{N}$.
A transport line LT_j is formed of an ordered list of stations, such as: $LT_j = \{S_{j_1}, \ldots, S_{j_{lt_j}}\}$ where lt_j is the size of the line LT_j corresponding to the number of stations of the line.
An itinerary is a sequence of pairs :
$((LT_i, Sens_{LT_i}, S_{PU,orig}, d_{PU}), (LT_i, Sens_{LT_i}, S_{D,dest}, d_D))$
where LT_i : indicates the transport line number i used, $Sens_{LT_i}$ indicates the direction[4] of the line LT_i used, $S_{x,y}$: indicates the station number x, y, with x specifying if it is a pick-up (PU) or a delivery (D), and y specifying the origin position or the destination, d_x: indicates the date of picking-up or delivering, at this station S_x.
Note : The pick-up time always precedes the time of delivery. In a itinerary there is always n (≥ 1) of couples PU&D. The time between a D and the next PU corresponds to waiting time, or even to the intermodal duration. The time between

---

[3] However, CIMO aims to manage all possible moves through different modes of transport in the territory of the urban area Belfort-Montbéliard.

[4] In this study, we consider the family of lines that can be defined with a departure station connected to a terminal's one $S_{i_1} \rightarrow S_{i_{lt_i}}$.

a PU and a D is considered for calculating the actual travel time, i.e. the actual time spent when a traveler is moving according to the modality (of each couple).

An itinerary Itn_i is presented as follows:

$$\begin{aligned} Itn_i = (&(LT_{i_{itin,1}}, Sens_{i_{itin,1}}, S_{i_1}, d_{i_1}),\\ &(LT_{i_{itin,2}}, Sens_{i_{itin,2}}, S_{i_2}, d_{i_2}),\\ &\qquad\vdots\\ &(LT_{i_{itin,l}}, Sens_{i_{itin,l}}, S_{i_l}, d_{i_l})) \end{aligned}$$

It is composed of an even number l of quadruplets. d_{i_1} is the date of PU or D at the station S_{i_l} We consider the timetable th, the list of time d_i of passage at the station S_{j_k} with a mode of transport associated to the line LT_j, and its direction is $Sens_{LT_{j_r}}$.
Also, we consider the matrix $Mod[i][j]$ of displacement between a station S_i and another station S_j. This matrix carries a certain amount of information, and part of them are derived from the input data, as the timetables of lines' passages at a station S_i and the direction belonging to the line $S_i \rightarrow S_j$. Each line is associated with a single indexed transport mode (Bus, Train, Car,...). Secondly, this matrix Mod carries information on shortest(s) path(s) to travel from S_i to S_j: existence of a multimodal itinerary and the time slot of the transport service operation. Therefore, the element of the matrix Mod gives access to features and following attributes:

- $modality(m) : boolean$: returns true if the modality m, with $m \in \{Bus, Train, Car, ...\}$, deserves the station S_i in the direction of station S_j;
- $th(mode_m) : date[][]$: returns the timetables of passage of all races for the mode $mode_m$;
- $Mod[i][j].th(m).next\_in\_th(ha) : date$: returns the next passage's date for the modality m, at station S_i, in the direction of S_j, that corresponds and succeeds to the date ha;
- $Mod[i][j].tet[m][c] : date$: returns the journey effective time of the itinerary c, of transport mode m, between stations S_i and S_j.

As illustrations, the first part of the Annex represent an extraction of real timetables issued from 8 bus or train lines of the networks OPTYMO, CTPM and SNCF, for a time periode ranged from 6h to 9h in the morning. The indexation of the 9 considered stations are : S_0 : Multiplexe Belfort, S_1 : Gare Belfort, S_2 : Gare Montbéliard, S_3 : Temple Montbéliard, S_4 : Hricourt, S_5 : Foch Belfort, S_6 : Place Ferrer Montbéliard, S_7 : Liberté Madrid Belfort, et S_8 : Jean Moulin Montbéliard.

C. Demand for mobility and qualitative objectives of multimodality

In the sequel, we present the itineraries generator algorithm of our CIMO calculator which addresses multiple objectives such as mobility request specifications, and qualitative objectives such as : not to arrive later than a specified date at the destination station, minimization of the number of modalities, minimazation of the overall trip time, and maximization of the correspondence waiting time. Last but not least, CIMO has to provide itineraries in a reasonable time.

The objectives of this algorithm are multiple:

1) consider the departure station and the arrival station;
2) consider a time window (TW for short) of the itinerary request, which is composed of the earliest departure time from the source position, and the latest arrival time to the destination position;
3) minimize the number of modal transfers: which is equivalent to minimize the number LT_l of quadruplets in the itinerary $Itin_i$.
4) minimize the travel time tt (including waiting time during correspondence(s)):

$$tt = d_{i_{LT_l}} - d_{i_1} \tag{1}$$

   $d_{i_{LT_l}}$: is the delivery date (D) to the destination station $S_{i_{LT_l}}$,
   d_{i_1} : is the PU date at the departure station S_{i_1}.
5) minimize the effective time of transportation tet (excluding waiting time during correspondence(s)) :

$$tet = \sum_{k=1}^{n} (d_{i_{lt_{2k}}} - d_{i_{lt_{2k-1}}}) \tag{2}$$

   with $n \in \mathbb{N},\ et\, l/2 = n$
6) maximize the sum of waiting times at correspondence's stations tac

$$tac = \sum_{k=1}^{n-1} (d_{i_{lt_{2k+1}}} - d_{i_l t_{2k}}) \tag{3}$$

7) and satisfy all of the following constraints:

   - those concerning one pair of PU and D positions in an itinerary:
     the same station cannot appear in both the PU and D 4-uple : $\forall j \in \left[1, \frac{l}{2}\right]: j, l \in \mathbb{N}$

$$S_{i_{2j-1}} \neq S_{i_{2j}} \tag{4}$$

     the D-date is upper than the PU-date :

$$d_{i_{2j}} > d_{i_{2j-1}} \tag{5}$$

     the line remains identical :

$$LT_{i_{2j}} = LT_{i_{2j-1}} \tag{6}$$

     idem for the direction :

$$Sens_{i_{2j}} = Sens_{i_{2j-1}} \tag{7}$$

   - those concerning two consecutive pairs, $\forall j \in \left[1, \frac{l}{2} - 1\right]: j, l \in \mathbb{N}$,
     the same PU station appears in the D station :

$$S_{i_{2j}} = S_{i_{2j+1}} \tag{8}$$

     PU, which follows a D, can be operated simultaneously :

$$d_{i_{2j+1}} \geqslant d_{i_{2j}} \tag{9}$$

DOI : 05.IJIS.2016.1.1

in an itinerary, a line is used once :

$$LT_{i_{2j+1}} \neq LT_{i_{2j}} \tag{10}$$

- those applyed to the entire itinerary : $\forall k, m, j \in \mathbb{N}$ avec $j \in [1, \frac{l}{2} - 1], k \neq 2j - 1, m \neq 2j$
  we do not take the same line twice

  $$LT_{i_k} \neq LT_{i_m} \tag{11}$$

  the same station cannot appear two times at PU's position or D's one, $\forall k, m, j \in \mathbb{N}$ with $j \in [1, \frac{l}{2} - 1], k \neq 2j, m \neq 2j + 1$

  $$S_k \neq S_m \tag{12}$$

- constraints which concern the number of modalities, the tet and the best itinerary:
  the number of modal transfers, and thus the number of 4-uple l_{Itin_i} in a itinerary i, must be greater or equal to the number of 4-uple of Best-Itinerary $l_{Best-Itin_i}$:

  $$l_{Itin_i} \geq l_{Best\_Itin_i} \tag{13}$$

  in case the numbers of modalities are identical, tet in $Itin_i$ must be greater than or equal to tet of $Best\_Itin_i$.

  $$tet_{Itin_i} \geq tet_{Bes\_Itin_i} \tag{14}$$

- finally, the constraint of existence of a chain of modalities, and the constraint of existence of coherent timetables passages between two consecutive stations of an itinerary of length l:
  $\forall j \in [1, \frac{l}{2} - 1]$,$\exists$ a modality m and a schedule of passage hp of a race c such as:

$$\begin{cases} \text{constraints } 2j-1, 2j & \begin{cases} Mod[2j-1][2j].modalit\acute{e}(m) = true \\ d_1 \leq hp \\ hp + tempsdetrajet(S_{2j-1}, S_{2j}, m, c) \leq d_{2j} \end{cases} \\ \text{constraints } 2j, 2j+1 & d_{2j+1} > d_{2j} \end{cases} \tag{15}$$

III. THE DYNAMIC PROGRAMMING ALGORITHM CUT AND PRICE AND SHARE OF CIMO

In this section we present the more achieved version of the algorithm (among four versions). The proposed algorithm is based on an exhaustive list of all the possible paths from a departure station $from$ to an arrival station $target$, according to a given time window, statis timetables, operationnal constraints and objectives.
The input variables are:

- **from**: (the index of) the departure station;
- depth: number of modal transfers;
- **target**: (the index of) the arrival station.
- mat : the displacement matrix Mod. Its size is $s{*}s$ where s is the number of stations in the network;
- horaireDepart : is the departure time;
- horaireArrivée : the desired arrival time.

The algorithm 1 represents the version v3.0 (APD-CPS_v3.0) of the dynamic programming algorithm Cut&Price&Share for generating all multimodal itineraries and satisfying the constraints. We denote by:

- **sorted stations** : all stations S_{Ts} are sorted by decrasing number of correspondence opportunities (**oc**), and then by the decresing shortest distance to the $target$ station;
- Table-blocage : a table of boolean. Its size is s, initialized with $False$ values. It serve not to pass through the same station 2 times in an itinerary;
- $price2'\begin{pmatrix} l \\ tet \\ tt \end{pmatrix}$: calculates the itinerary cost (to be detailed Part 4);
- Best-Itin: the best itinerary calculated during run-time calculation, or at the issue;
- **Search-next-mode-transportation** is a function that returns the next possible mode satisfying all the constraints, between two stations S_i and S_{i+1} of an itinerary Itin.

Algorithm 1 Dynamic Programming algorithm Cut&Price&Share (APD-CPS_v3.0)

```
$price_{min}\begin{pmatrix} l \leftarrow \infty \\ tet \leftarrow \infty \\ tt \leftarrow \infty \end{pmatrix}$
Itin ← ∅
Best-Itin ← ∅
Search-Itineraries(Position from, int depth, MatPoint2Point mat, Position to, Time horaireActuel, Time horaireDepart, Time horaireArriveePlusTard, Itinerary Itin){
Itin← Itin+from
if from.equalTo(to) then
    for i=1 to depth do
        Search-next-mode-transportation(mat,Itin,i,i+1,Time ha=horairActuel)
        Best-Itin← Itin
    end for
    $price_{min} \leftarrow price2'(Itin)$
end if
table-blocage[from] ← True //block the from position (CUT-V0: Do not go through the same station twice in a itinerary)
for $S_i$ in ordered stations do
    if ((mat[from][$S_i$].modality(Bus)=true or
    mat[from][$S_i$].modality(Train)=true) and
    (table-blocage[i]=False)) then
        (CUT-V2+Price2')
        if $price2'(Itin) \leq price_{min}$ then
            Search-Itineraries($S_i$,depth+1,mat,to,
            horaireActuel,horaireDepart,
            horaireArriveePlusTard,Itin)
        end if
    end if
end for
table-blocage[from] ← False //unblock from position
}
```

This is a recursive algorithm, based on an in-depth course. It realizes a path considering all network stations. If there is a station S_i for which there is an accessible modality (bus line or train line) and a schedule of PU from the current position $from$, then we realize the recursive call by changing the position and depth as follows:

- the from position becomes S_i, the next more satisfying station, and we then seek for completing the itinerary from the station S_i. We block S_i, not to pass twice;

DOI : 05.IJIS.2016.1.1

- depth=depth+1, the depth is increased by the modal step.

The algorithm searches for possible intermediate stations from the new start position S_i. This procedure is repeated until we reach the destination position **to**.

Once the connection is verified, we prepare an itinerary from the starting station **from** to the arrival station **to**. It displays the itinerary by posting, at each station S_i the type of transport mode that passes through this station, the time d of PU according to the transport mode selected, and the waiting time for this mode of transport to pass through S_i . After displaying the obtained itinerary, we unblock gradually the last intermediate station of the best and correct itinerary. The remaining intermediate itinerary (which can be considered as a common or shared parts) can be used to calculate other itineraries which can improve, incrementally the best itinerary.

When displaying an itinerary, to go from a station S_i to another station S_{i+1}, the current time is taken into account which is the traveler's arrival time at station S_i. The algorithm seeks for the mode of transport, either the bus or the train which catches passangers at station S_{i+1}. If the mode of transport is the bus (resp. train or foot, or car), the algorithm seeks the next bus and displays its number and schedule of the passageway at the station S_i. Then it calculates the waiting time at the station between two consecutives modes used to operate this current itinerary.

A. *Theoretical Complexity of APD-PS v3.0*

The complexity depends on various parameters:

- $|S_{Ts}| = s$: number of stations of the model;
- S_i : i is the index of the station, $i \in [0, s-1]$;
- S_{from}: $from$ is the index of the departure station;
- S_{to}: to is the index of the arrival station;
- q: the number of races of the line LT.

The general formula of the complexity can be modeled by the following formula :

$$\begin{aligned} T(s, S_{from}, depth, S_{to}) = & \sum_{i=0}^{s-1} T(s, S_i, depth+1, S_{to}) \Omega_{S_{from}}^{S_i} \\ & + \sum_{i=0}^{s-2} A_{s-2}^{i} * \overline{nbS/L} * q \end{aligned} \tag{16}$$

with :

- A_{s-2}^i is the arrangement of i stations amongst $s-2$ stations, excluding stations **from** and **to**;
- $\Omega_{IdS_{from}}^i$ =1 if there is a mode of transport returns the index station $from$ to the index station i, otherwise it is equal to 0;
- $\overline{nbS/L}$ is the average number of stations per line.

IV. A TERRITORY GENERATOR TO EVALUATE DIFFERENT VERSIONS OF ALGORITHM APD-CPS

We have realized a random data generator. We test each version of algorithms by varying number of network stations, the number of lines, the arrival station, the departure station and the time window.

We denote by:

- nbS: number of network stations;
- nbL: number of network lines;
- $\overline{nbS/L}$: the average number of stations per line;
- $\overline{nbC/L}$: the average number of correspondences per line;
- $nbCTotal$: the total number of correspondences;
- $nbComb-it$: the total number of all combinations of itineraries ;
- **from** and **to**: the departure station, resp. arrival's one ;
- Temps_exec: the necessary time to explore all itineraries from the departure station to the arrival station and to display them;
- nbIt: number of calculated itineraries.

Tests are performed on the Java platform with a machine DELL i7[5]. The table IV page i (column Execution time of APD-CPS v1.0) identifies initial results. These first results are basically made on the first set of tests extracted from a realistic trimodal network of Urban Area Belfort-Montbéliard. For a request established by the traveler with a departure station is Liberté Madrid (Belfort) and an arrival station is Temple (Montbéliard), APD-CPS v1.0 returned two itineraries: The first itinerary is with a modality number $l = 3$ and an effective time of transportation $tet = 40min$, the second itinerary is with a modality number $l = 4$ and an effective time of transportation $tet = 40min$. This version calculates all possible itineraries based on the corresponding time window without applying constraints on the modality number or l or the effective time of transportation tet.

The table V page ii (column Execution time of APD-CPS v1.0) identifies results obtained by applying this version on the random data generator. This table resumes the exponential behavior of time calculation and complexity of our propsoed algorithm, which combines itineraries generation modulo the existence of modalities, and the instanciation with effective PU and D times, and the in incremental display of best intineraries computed.

To address pramatically this complex problem study, we propose several gradual solutions. The objective is to demonstrate the feasibility of apprehending the calculation multimodal itinerary (nb modes ≥ 3, and a number of modal transfers for unlimited itineraries) on aggregatable operating transport territories (Sizes of 1000 stations at least).

Version 1.2 of Algo APD-CPS realizes and considers the following items.

- We denote by nM the maximum number of modalities in an itinerary. If this number is exceeded, then we abandon the constitution of the current itinerary. For example for such itinerary if $nM > 3$, it will be useless to continue to follow this itinerary because we always search for itinerary with a smaller number of modalities. That reduces the risk of a correspondence, and this also reduces the financial pressure on the traveler that should possess a fixed or specific ticket.

---

[5] In this study, the performance of the calculator with realistic tests and high size is not optimized. Neither the choice of the language C, and optimization possibilities of the compiled code and the best selection and use of libraries, neither a powerful dedicated machine, neither a parallelized work, neither a working static preparation tasks, is assessed at this stage of our work.

- The Cut Version1.2 (CUT-V1) is modeled by the following formalization: $\forall S_{i_k}, S_{i_m} \in Itin_i$

$$l \leq 3 \wedge (\forall k, m, j \in \mathbb{N}, j \in [1, l/2], k \neq 2j, m \neq 2j+1 : S_{i_k} \neq S_{i_m}) \tag{17}$$

  The table IV page i (column Execution time of APD-CPS v1.2) identifies initial results of tests extracted from the realistic trimodal network. For the same request established in the example of version 1.0, APD-CPS v1.2 returns only one itinerary. Comparing to the result provided by APD-CPS v1.0, only the first itinerary (modality number $l = 3$ and an effective time of transportation $tet = 40min$) has been selected because it coincides with the constraint of $l \leq 3$. The second itinerary (modality number $l = 4$ and an effective time of transportation $tet = 40min$) has been eliminated because $l > 3$.
  The table V page ii (column Execution time of APD-CPS v1.2) identifies results obtained by applying this version on the random data generator.
  The evaluation of version 1.2 shows the influence of the calculated solutions' display. It also shows the correlation between a test set, of small size, which is extracted from a real network, and the operating network of a similar size generated by our generator tests. Separating the display phase and the instanciation (modalities and real PU and D times), from the incremental in-depth course calculation of optimized itineraries is of greatest interest. Hence, APD_CPS version 1.2 breaks the complexity by reducing the computation time. For example, for a model of 13 stations, CIMO (Test7) consumes from 2 minutes to 0.0133 seconds to solve the multimodal request. If we compare the number of displayed itineraries $nbIt$, for the Test7, we decrease the displays from 74895 to 22. Note that, in this version 1.2, cost assessment (number of modalities) also avoids the reconstruction of all the combinations of the modalities of each portion (PU, D), and schedules PU or D.

We denote by $Best\_Itinerary$, the best itinerary before generating the display. This itinerary is also composed by the number of modalities (or modal transfers), as well as the tet. The display's phase is improved in this version by displaying only the calculated itineraries which the cost (number of modalities, and the tet), improves the solution $Best\_Itinerary$.
We display the itinerary composed of a minimum number of stations. We can minimize the number of correspondences of lines. An additional track to break the complexity, is to extend the Cut on this number of modalities, and so on $Price$ that considers the number of modalities and also the tet. This element is the main feature of version 2.0 of the algorithm APD_CPS.
These first series of tests can be criticized, at this state, because of bounded by targeted very large size networks. We have to reduce the complexity influence of passage's frequency of transportation modes, because even for the extracted sun-network considered, more than 10 bus passes through any station and for a time window of 2 hours.

So, to render possible managing a real transport network with a bigger number of stations, we propose Version 2 of the algorithm. We introduce a new objective of optimal itinerary which minimizes the number of modalities and the cost of effective time of transport (tet). This $Price$ is considered before the recursive call, to extend the previous Cut, with the hopes to speed up the calculation of the best multimodal solution. We estimate the length path during itinerary calculation. This cost promotes, initially the minimization of the modalities' number, then the tet minimization. if this cost exceeds the minimal cost of the current Best_Itinerary, we cut the recursive call.
The Cut (CUT-V2) is formalized as, $\forall S_{i_k}, S_{i_m} \in Itin_i$:

$$\begin{gathered} Itin_i \rtimes_1 Best - Itin_i \wedge \\ (\forall k, m, j \in \mathbb{N}, j \in [1, l/2], k \neq 2j, m \neq 2j+1 : S_{i_k} \neq S_{i_m}) \end{gathered}$$

Note that :$\rtimes_1$ is a law that sorts all itineraries $Itins$ according to $price2\begin{pmatrix} l \\ tet \end{pmatrix}$ defined as follows :

$$\begin{gathered} Itin_j \rtimes_1 Itin_i \Leftrightarrow \\ ((Itin_i.l < Itin_j.l)) \vee \\ ((Itin_i.l = Itin_j.l) \wedge (Itin_i.tet < Itin_j.tet)) \end{gathered}$$

$\rtimes_2$ extends the previous law by ordering all itineraries $Itins$ according to $price2'\begin{pmatrix} l \\ tet \\ tt \end{pmatrix}$:

$$\begin{gathered} Itin_j \rtimes_2 Itin_i \Leftrightarrow \\ ((Itin_i.l < Itin_j.l)) \vee \\ ((Itin_i.l = Itin_j.l) \wedge (Itin_i.tet < Itin_j.tet)) \vee \\ ((Itin_i.l = Itin_j.l) \wedge (Itin_i.tet = Itin_j.tet) \vee \\ \wedge (Itin_i.tt < Itin_j.tt)) \end{gathered}$$

The table IV page i (column Execution time of APD-CPS v2.0) identifies initial results of tests extracted from the realistic trimodal network. For the same request established in the example of version 1.0 and version 1.2, APD-CPS v2.0 returns only the first itinerary (modality number $l = 3$ and an effective time of transportation $tet = 40min$). Considering this first itinerary as $Best\_Itinerary$, in the second itinerary (modality number $l = 4$ and an effective time of transportation $tet = 40min$), APD-CPS v2.0 stops the itinerary exploration when l exceed 3. It is useless to continue the calculation because this solution will not improve the result to provide for the traveler.

The table V page ii (column Execution time of APD-CPS v2.0) shows performance's gain after the application of the new Cut. We can notice the immediate beneficial effect by demonstrating the ability of our approach to address networks of 500 stations, and that in a reasonable time. For Test12, in the simulation part, the time required to achieve and find Best itinerary calculation is in the order of 4 secondes. This duration is equivalent to the time required to manage a network of 100 stations by using Algo version 1.2.

To further reduce the complexity, another solution is introduced (APD-CPS_v3.0). This version sorts accessible stations, from a reference station, by minimizing the degree of correspondence and the proximity to the target station. When browsing stations to find the next intermediate station to consider, stations with an order of correspondence better than others are treated firstly. This increases the opportunity to find the optimal path in a shorter time by exploring fastly the best itineraries and then cutting more efficiently useless possible itineraries (composed of more correspondences, with a worst tet).

$\rtimes_3$ is a law to all stations S_{Ts} according to the order of correspondence oc and the distance between the station S_{i_k} and destination to(S_{i_l}) : $\forall S_{i_k}, S_{i_m} \in Itin_i$

$$\begin{gathered} S_{i_k} \rtimes_3 S_{i_m} \Leftrightarrow \\ ((S_{i_k}.oc > S_{i_m}.oc))\vee \\ ((S_{i_k}.oc = S_{i_m}.oc) \wedge (d(S_{i_k}, S_{i_l}) < d(S_{m_k}, S_{i_l}))) \end{gathered}$$

The figure 1 presents, synthetically the evolution of the execution time of the various versions of the algorithm APD_CPS depending on the network size (number of stations). This figure resumes well the exponential behavior of Version 1.0 (red curve), and others. Curve 2 in green, presenting version 1.2, also evolves linearly on a logarithmic grid Log_{10}. Noticeably below the previous curve, the version 2.0 proposal allows considering networks with a size of 200 stations, and dozens of lines, for a number of modalities lower or equaling to 3.

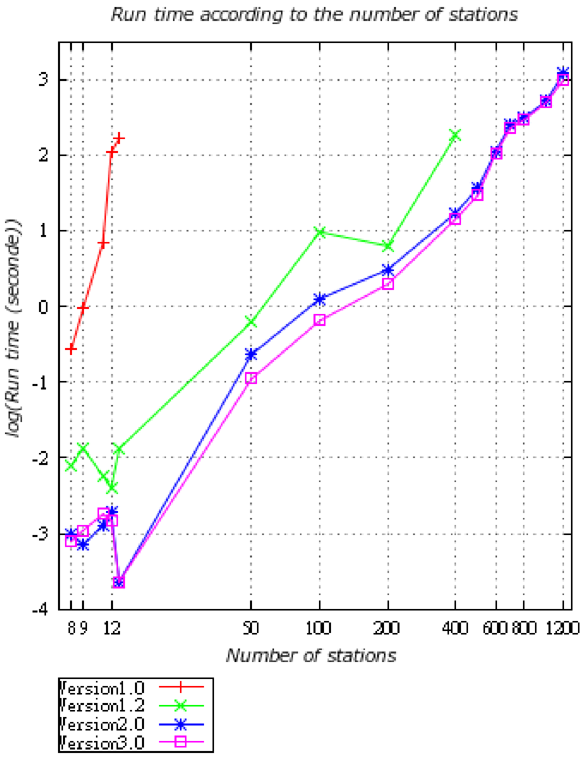


Figure 1: Cimo time-solving evolution based on the number of stations

The blue curve (version 2.0) shows the ability of our Dynamic Programming algorithm approach $Cut\&Price\&Share$ to apprehend real number of modalities, and for trans-territories netwoks ’ sizes that become realistic. Thus, the inclusion of transit schedules in station is no longer prohibitive to offer multimodal itineraries to travelers.

Curve 3 in pink shows the performance of the version 3.0 of the algorithm for treating networks with size 800. This version 3.0, whose results are presented in the table V (column Execution time of APD-CPS v3.0) implements an ordering of neighboring stations that are under treatment. In this ordered list, the comparison function first favors minimizing the number of correspondences, and then the minimization of the distance from this station to the final destination. The size 800 is the cumulative size of the networks Optymo, CTPM and SCNF that are deserving the Urban area Belfort-Montbéliard. At the date of March 2015, this size of these 3 networks number of stations, of lines, of races by line (size of timetables), and for 3 modalities Bus-Train-Car, our tests are well beyond the field requirements, and for a ratio that is favorable to our study 500/800 (cf. Bottom of the table V page ii).

Two other parameters are defined for comparing evolution between the different versions that are proposed:

- Le speed-up SU
- The relative gain RG

For two versions V1 et V2 we must have the following data:
t_1 is the time spent in order to find the best solution in V1.
t_2 is the time spent in order to find the best solution in V2.
t_0 is the time of test trigger for both versions V1 and V2.
Le speed-up SU provides information on the acceleration rate between versions V1 and V2

$$SU = \frac{t_1 - t_0}{t_2 - t_0}$$

$$\begin{cases} SU > 1: & \text{V2 evolves more rapidly than V1} \\ SU = 1 & \text{V1 and V2 evolve with the same manner} \\ SU < 1 & \text{V1 evolves more rapidly than V2} \end{cases}$$

The relative gain RG provides information on the gain brought by a V2 version compared to another version V1.

$$RG = \frac{t_1 - t_2}{t_1 - t_0}$$

$$\begin{cases} RG > 0: & \text{V2 provides more gain than V1} \\ RG = 0 & \text{V1 et V2 evolve with the same manner} \\ RG < 0 & \text{V1 provides more gain than V1 V2} \end{cases}$$

The table I page 9 identifies the variation of SU and RG between Version1.0 and Version1.2. This table shows that v1.2 is evolving more faster than v1.0 with providing more gain in the scale of execution time.
Test6 (network of 12 stations) indicates that v1.2 evolves 27750 times more faster than v1.0. The relative gain RG is adjacent to one (0.99) that is the most ideal value of RG.

The table II page 9 identifies the variation of SU and RG between Version1.2 and Version2.0. This table shows that v2.0 is evolving more faster than v1.2 with providing more gain in the scale of execution time.
From the test 9 (network of 100 stations), the relative gain is fixed at 0.99 which demonstrates that version2.0 is able better than version1.2 to manage big transport networks.

The table III page 9 identifies the variation of SU and RG between Version2.0 and Version3.0. This table shows that v3.0

DOI : 05.IJIS.2016.1.1

Number of stations	SU	RG
8	34.37	0.97
9	70.03	0.98
11	1531.55	0.99
12	27750	0.99
13	13007	0.99

Table I: Evolution of SU and RG between Version1.0 and Version1.2

Number of stations	SU	RG
8	8	0.87
9	19.28	0.94
11	3.46	0.11
12	2.1	0.52
13	57.82	0.98
50	2.8	0.65
100	4245	0.99
200	3435	0.99
400	92500	0.99

Table II: Evolution of SU and RG between Version1.2 and Version2.0

is evolving more faster than v2.0.
In this comparison SU has not exceeded 2 but while it's greater than 1 so the new version is faster than the old one. just note that for test 2.3 and 9 for networks 9 and 11 stations, the speed up is less than 1 (gain on less than 0), this is explained by that making an order for stations, according to the correspondence and the proximity of the destination increase the problem complexity for small-sized networks, but brings a reductive effect on the big networks.

Number of stations	SU	RG
8	1.25	0.2
9	0.63	-0.57
11	0.72	-0.38
12	1.26	0.21
13	1	0
50	1.97	0.49
100	1.95	0.48
200	1.55	0.36
400	1.19	0.16
500	1.23	0.2
600	1.07	0.07
700	1.11	0.1
800	1.04	0.04
1000	1.06	0.05
1200	1.007	0.007

Table III: Evolution of SU and RG between Version2.0 and Version3.0

V. CONCLUSION

We have proposed a comprehensive review of CIMO, a solution to calculate multimodal itinerary. This study was approved by tests from the different versions that are proposed. Two comparison criteria are defined for evaluating its (the speed up and the relative gain).

Our study proposed :

1) formalizing terrain data (Lines, Directions, correspondence stations, timetables) and queries of requests itineraries (departure at the earliest, arrival at the latest).
2) formalizing multiple constraints of the problem:
    - the calculated and displayed itineraries with respect to the window time of the application.
    - each itinerary is consistent with lines, their directions, and linear orderings of the stations, as well as with respect to the origin destination matrix .

    and formalizing the multi-objectives of :
    - Satisfying the constraints.
    - minimizing the modality number.
    - minimization of the effective time of transportation.
    - minimizing of travel time including the correspondence waiting time.
    - minimizing the standard deviation of waiting times in correspondences.

The Cimo algorithm performs a prior calculation of the optimal itineraries, modulo the combinatorial instantiations of real modalities (when multiple modalities serve a segment between 2 stations). It is in the second time of displaying generation that all other objectives are valued using the previously exposed hierarchy. By this approach, practical complexity of the algorithm is broken. The termination of Cimo, and the production of the best solution, is accelerated by ordering accessible stations from a reference station, depending on the degree of correspondence as well as the proximity to the target station (version 3.0).

The computational complexity of problem solving and progressive tests show the influence of the display economies, that of the cut and the acceleration of the ordering applied to the neighboring stations. This study shows that an optimal itinerary can be calculated from a dynamic programming algorithm cut and price and share from generation of 3-modal itineraries of 2 neighboring transport networks connected to the size of 2 prefectures serving 300 000 inhabitants.

However, our medium-term ambition is to treat all the modalities of the territory. It is appropriate to treat the dynamism, and also access to soft modes (cycling, walking), to triggered public transport and also studying and integrating the dynamic carpooling.

REFERENCES

[1] Idriss HASSINE, and Philippe CANALDA. Cimo: an efficient 2-phases calculator of multimodal itineraries for real trans-territories based on a dynamic programming In *The World Congress on Information Technology and Computer Applications*, 2015.

[2] N. Borole, D. Rout, N. Goel, P. Vedagiri and T. V. Mathew. Multimodal Public Transit Trip Planner with Real-time Transit Data In *journal Procedia - Social and Behavioral Sciences, volume 104, Proceedings of the 2nd Conference of Transportation Research Group of India (2nd CTRG)*, pages 8, 2013.

DOI : 05.IJIS.2016.1.1

[3] C. Chen, R. Kitamura and J. Chen. Multimodal daily itinerary planner interactive programming approach In *Transportation Research Board, volume 1676*, pages 37 – 43, 1999.

[4] A. Ranade, M. Srikrischna, K. Tilak and M. Datar. Mumbai Navigator In *Indian Journal of Transport Management*, 2005.

[5] J.Q. Li, K. Zhou, L. Zhang, L. and W.-B. Zhang. A Multimodal Trip Planning System Incorporating the Park-and-Ride Mode and Real-Time Traffic and Transit Information, In *Proceedings ITS World Congress, Volume 25*, pages 65 – 76, 2010.

[6] J. Jariyasunant, D. Work, B. Kerkez, R. Sengupta, S. Glaser and A. Bayen. Mobile Transit Trip Planning with Real-Time Data, In *Transportation Research Board 89th Annual Meeting, Volume 2*, pages 139 – 146, 2010.

[7] T. Walsh and L. Antsfeld. Finding multi-criteria optimal paths in multimodal public transportation networks using transit algorithm. In *19th world Congress on Intelligent Transport Systems*, 2012.

[8] H. Bast, E. Carlsson, A. Eigenwillig, R. Geisbe rger, C., Harrelson, V., Raychev and F. Viger. Fast Routing in Very Large Public Transportation Networks using Transfer Patterns. In *European Symposium on Algorithms, Lecture Notes in Computer Science, Volume 6346*, pages 290 – 301, 2010.

[9] Transit Google, http://www.google.com/intl/en/landing/transit.

[10] J.-Q. Li, K. Zhou, L. Zhang and W.-B. Zhang. A multimodal trip planning system with real-time traffic and transit information. In *Journal of Intelligent Transportation Systems, 16(2)*, pages 60 – 69, 2012.

[11] B. Lovepeople and A. Lovescience. My Searcher is Rich In *Proceedings Jokes'94*, pages 107–120, Miraflores, Spain, 1994

[12] J.-M. Su and C.-H. Chang. The multimodal trip planning system of intercity transportation in Taiwan. In *Expert Systems with Applications, Volume 37*, pages 6850 – 6861, 2010.

[13] V. Spitadakis and M. Fostieri. WISETRIP- International multimodal journey planning and delivery of personalized trip information In *Procedia Social and Behavioral Sciences, Volume 48*, pages 1294 – 1303, 2012.

[14] W. Xu, S. He, R. Son and S. Chaudhry. Finding the K shortest paths in a schedule-based transit network In *Computers and Operations Research, Volume 39, Elsevier*, pages 1812 – 1826, 2012.

[15] M. Friedrich, I. Hofsab and S. Wekeck. Timetable-based transit assignment using branch and bound. In *Proceedings of 80th annual of transportation research board*, Washington D.C., USA, 2001.

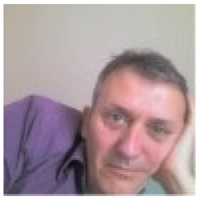
[16] NJ. vander Zijpp, and S. Fiorenzo Catalano. Path enumeration by finding the constrained K-shortest paths In *Transportation Research-B, number 35*, pages 545 – 63, 2005.

[17] Mohamed Amine Kammoun. Conception d'un systme d'information pour l'aide au dplacement multimodale : Une approche multi-agents pour la recherche et la composition des itinraires en ligne *cole centrale de Lille*, 2007

[18] Christian Artigues, Marie-Jos Huguet, Fallou Gueye, Frderic Schettini, Laurent Dezou. State-based accelerations and bidirectional search for bi-objective multimodal shortest paths. In *Transportation Research Part C: Emerging Technologies, Elsevier*, pages 233 – 259, 2013.

[19] Dominik Kirchler. Efficient routing on multi-modal transportation networks. Data Structures and Algorithms. In *Ecole Polytechnique X*, 2013.

Idriss HASSINE received his engineering degree in computer science from the National Engineering School of Tunis (ENIT) in 2014. Currently he performs his research internship at FEMTO-ST institute in Montbliard, France in order to obtain the master degree in distributed and mobile computing from Technological University of Belfort-Montbliard.



Philippe CANALDA received his Ph.D (1997) from the INRIA Research Centre and the University of Orleans. He worked 2 years in the Associated Compiler Expert start-up on real-time optimization compilers, then 2 years at LORIA research institute on synchronized and complex processes. Since 2001, as assistant professor at UFC, he is focusing on dynamic matching algorithm and Very Reactive Complex Processes to provide safe and secure mobility services for ITS: notably On Demand Transportation applications, and Real-Time Carpooling applications. He participates as active member to the French (and European) Normalization of Real-Time ITS services since 2008. He's also inventing new combined positioning system wireless-based, position person in urban or peri-urban situation to provide continuity of mobile services.

APPENDIX

TIMESLOT EXTRACTION [6H-8H30] FROM TIMETABLE OF OPTYMO BUS' NETWORK, CTPM BUS' NETWORK AND SNCF TRAIN' NETWORK

OPTYMO BUS' network of Belfort

ligne1 : Foch Belfort→Gare Belfort

Foch Belfort	6h15	6h35	6h55	7h10	7h20	7h27	7h34	7h41	7h48	7h55
Gare Belfort	6h20	6h40	7h00	7h15	7h25	7h32	7h39	7h46	7h53	8h00

ligne1 : Gare Belfort→Foch Belfort

Gare Belfort	6h22	6h38	6h48	6h55	7h02	7h09	7h16	7h23	7h30	7h37	7h44	7h51	7h58
Foch Belfort	6h26	6h42	6h52	6h59	7h06	7h13	7h20	7h27	7h34	7h41	7h48	7h55	8h02

ligne2 : Multiplexe→Liberté Madrid

Multiplexe	6h17	6h31	6h45	7h07	7h14	7h28	7h42	7h56
Gare Belfort	6h20	6h34	6h48	7h10	7h17	7h31	7h45	7h59
Liberté Madrid	6h23	6h37	6h51	7h10	7h13	7h34	7h48	8h02

ligne2 : Liberté Madrid →Multiplexe

Liberté Madrid	6h13	6h52	7h06	7h20	7h41	7h48	8h02
Gare Belfort	6h16	6h55	7h09	7h23	7h44	7h51	8h05
Multiplexe	6h19	6h58	7h12	7h26	7h47	7h54	8h08

ligne3 : Multiplexe Belfort→Gare Belfort

Multiplexe Belfort	6h19	6h39	6h49	7h09	7h19	7h29	7h39	7h49	7h59
Gare Belfort	6h22	6h42	6h52	7h12	7h22	7h32	7h42	7h52	8h02

ligne3 : Gare Belfort→ Multiplexe Belfort

Gare Belfort	6h18	6h29	6h38	6h48	7h59	7h08	7h18	7h28	7h38
MultiplexeBelfort	6h21	6h32	6h41	6h51	7h02	7h11	7h21	7h31	7h41

CTPM BUS' network of Montbéliard

ligne1 : Gare Montbéliard→Temple

Gare Montbéliard	6h11	6h25	6h39	6h53	7h16	7h26	7h46
Place Ferrer	6h15	6h29	6h43	6h57	7h20	7h30	7h50
Jean Moulin	6h22	6h36	6h50	7h04	7h28	7h38	7h59
Temple	6h33	6h46	7h00	7h15	7h39	7h49	8h11

ligne1 :Temple→Gare Montbéliard

Temple	6h14	6h39	6h52	7h10	7h25	7h40	7h55
Jean Moulin	6h25	6h50	7h04	7h22	7h37	7h52	8h06
Place Ferrer	6h33	6h58	7h12	7h30	7h45	8h00	8h14
Gare Montbéliard	6h36	7h01	7h15	7h33	7h48	8h03	8h17

ligne2 : Place Ferrer→Temple

DOI : 05.IJIS.2016.1.1

Place Ferrer		6h35			7h17		
Gare Montbéliard	6h09	6h38	6h52	7h06	7h20	7h41	7h51
Temple	6h31	7h00	7h16	7h30	7h44	8h05	8h14

ligne2 :Temple→Gare Montbéliard

Temple	6h07	6h22	6h37	6h54	7h05	7h23	7h39	7h52
Gare Montbéliard	6h27	7h42	7h57	7h16	7h29	7h47	8h03	8h15

ligne3 : Gare Montbéliard→Place Ferrer

Gare Montbéliard	6h16	6h55	7h24	7h41
Place Ferrer	6h19	6h58	7h28	7h45

ligne3 :Place Ferrer→Gare Montbéliard

Place Ferrer	6h18	6h40	7h01
Gare Montbéliard	6h21	6h43	7h04

ligne4 : Place Ferrer→Jean Moulin

Place Ferrer	6h42	7h35	8h12
Jean Moulin	6h49	7h44	8h21

ligne4 :Jean Moulin→Place Ferrer

Jean Moulin	6h15	7h11	7h37
Place Ferrer	6h22	7h19	7h45

SNCF TRAIN' network Belfort-Montbéliard

Gare Belfort→Gare Montbéliard

Train Number	Ter 95860	Ter 94008	Ter 94832	Ter 94014
Gare Belfort	6h24	7h04	8h04	8h24
Hricourt	6h31	7h11	8h11	8h31
Gare Montbliard	6h39	7h17	8h17	8h39

Gare Montbéliard→Gare Belfort

Train Number	Ter 94003	Ter 94007	Ter 95863	Ter 95833	Ter 94103
Gare Montbéliard	6h40	7h41	8h13	8h27	8h47
Hricourt	6h47	7h50	8h20	8h34	8h55
Gare Belfort	6h55	7h56	8h29	8h41	9h01

TEST MEASUREMENT

The table IV identifies initial results. These first results are basically made on the first set of tests extracted from a realistic trimodal network of Urban Area Belfort-Montbéliard for a request established by the traveler with a departure station is Liberté Madrid (Belfort) and an arrival station is Temple (Montbéliard).
The table V identifies results obtained by applying different version of APD-CPS on the random data generator.

from	to	APD_CPS_v1.0				
		Temps_exec	Complexity	nbIt	l	tet
station0	station3	0sec. 056ms	19	2	3 4	40min 40min
station5	station8	0sec. 026ms	19	2	4 4	54min 29min
station7	station3	0sec. 050ms	19	2	3 4	40min 40min
station7	station6	0sec. 052ms	19	2	5 3	58min 22min
station5	station6	0sec. 025ms	19	2	5 3	62min 23min
station0	station8	0sec. 053ms	19	2	4 4	52min 27min
station1	station8	0sec. 023ms	17	2	3 3	49min 24min

from	to	APD_CPS_v1.2				
		Temps_exec	Complexit	nbIt	l	tet
station0	station3	0sec. 021ms	13	1	3	40min
station5	station8	0sec. 010ms	9	0		
station7	station3	0sec. 021ms	13	1	3	40min
station7	station6	0sec. 034ms	13	1	3	22min
station5	station6	0sec. 020ms	13	1	3	23min
station0	station8	0sec. 010ms	9	0		
station1	station8	0sec. 023ms	17	2	3 3	49min 24min

from	to	APD_CPS_v2.0				
		Temps_exec	Complexit	nbIt	l	tet
station0	station3	0sec. 056ms	10	1	3	40min
station5	station8	0sec. 064ms	19	2	4 4	54min 29min
station7	station3	0sec. 058ms	10	1	3	40min
station7	station6	0sec. 042ms	13	2	5 3	58min 22min
station5	station6	0sec. 089ms	19	2	5 3	62min 23min
station0	station8	0sec. 089ms	19	2	4 4	52min 27min
station1	station8	0sec. 028ms	17	2	3 3	49min 24min

Table IV: Tests Mesurements for different versions of ALGO APD_CPS applied on realistic trimodal network of Urban Area Belfort-Montbéliard

	nbS	nbL	$\overline{nbS/L}$	$\overline{nbC/L}$	**from**	**to**	Temps_exec de APD_CPS				nbIt de APD_CPS			
							v1.0	v1.2	v2.0	v3.0	v1.0	v1.2	v2.0	v3.0
Test1	8	8	7	7	3	1	0,275sec	0,008sec	0,001sec	0,0008sec	227	5	2	1
Test2	9	8	6	6,8	5	2	1,3sec	0,016sec	0,0008sec	0,0007sec	210	6	3	1
Test3			7	7	8	0	0,591sec	0,011sec	0,0006sec	0,0015sec	759	9	1	1
Test4	11	8	8	8	4	10	9,045sec	0,007	0,0006sec	0,0006sec	3406	3	1	1
Test5	11	8	8	8	6	4	4,739sec	0,002sec	0,002sec	0,003sec	1880	4	4	4
Test6	12	8	6	6	10	2	1min. 51sec	0,004sec	0,0019sec	0,0015sec	52165	6	1	1
Test7	13	8	4	4,8	5	4	2min. 53sec	0,0133sec	0,00023sec	0,00023sec	74895	22	1	1
Test8	50	8	22	22	41	23	*	0,6268sec	0,217sec	0,110sec	*	15	2	1
Test9	100	8	43	43	61	73	*	9,765sec	0,0023sec	0,0015sec	*	194	1	1
			43	43	10	81	*	*	1,933sec	0,817sec	*	*	1	1
			43	43	19	51	*	*	1,332sec	0,79sec	*	*	1	1
			43	43	5	71	*	*	1,82sec	1sec	*	*	1	1
Test10	200	8	81	80,4	64	27	*	2,210sec	0,0017sec	0,00079sec	*	24	1	1
			81	80,4	152	111	*	10,5sec	0,002sec	0,002sec	*	9	1	1
			75	80,4	50	171	*	*	7,988sec	5,785sec	*	*	1	1
			72	72	190	101	*	*	5,606sec	3,290sec	*	*	1	1
			70	67,6	158	47	*	*	1,916sec	0,868sec	*	*	1	1
Test11	400	8	158	158,4	160	53	*	3min. 5sec	0,002sec	0,0014sec	*	96	1	1
			137	134	278	143	*	*	13,207sec	10,49sec	*	*	1	1
			156	153,2	71	103	*	*	8,061sec	5,028sec	*	*	1	1
			137	130,4	18	313	*	*	48,639sec	42,805sec	*	*	1	1
Test12	500	8	203	201,2	410	27	*	*	0,0126sec	0,0038sec	*	*	1	1
			202	200,8	317	115	*	*	4,366	2,737sec	*	*	1	1
			204	198,6	131	228	*	*	24,960	22,335sec	*	*	1	1
			203	201,2	111	492	*	*	1min. 58sec	1min. 34sec	*	*	1	1
Test13	600	8	212	211,2	473	343	*	*	2min. 19sec	2 min	*	*	1	1
			212	211,2	211	501	*	*	2min. 30sec	2min. 19sec	*	*	1	1
			223	221,2	521	51	*	*	0,500sec	0,300sec	*	*	1	1
			192	188,8	31	481	*	*	2min. 44sec	2min. 42sec	*	*	3	3
Test14	700	8	241	230	271	437	*	*	3min. 19sec	2min. 52sec	*	*	1	1
			263	254,6	131	681	*	*	4min. 45sec	4min. 17sec	*	*	1	1
			212	192,8	73	527	*	*	4min. 23sec	4min. 9sec	*	*	1	1
			261	237,2	325	692	*	*	5min. 3sec	4min. 20sec	*	*	1	1
Test15	800	8	356	352,2	340	67	*	*	2,756sec	2,203sec	*	*	1	1
			356	352,2	345	700	*	*	6min. 36sec	6min. 6sec	*	*	1	1
			284	262,2	455	792	*	*	6min. 39sec	6min. 23sec	*	*	1	1
			289	273,2	505	691	*	*	6min. 12sec	6min. 12sec	*	*	1	1
			310	303,2	485	721	*	*	6min. 26sec	6min	*	*	1	1
Test16	1000	8	390	390,2	845	101	*	*	11,565sec	6,683sec	*	*	1	1
			390	390,2	665	304	*	*	2min. 27sec	2min. 6sec	*	*	1	1
			327	292,6	185	823	*	*	11min. 34sec	11min. 14sec	*	*	1	1
			368	352,2	211	932	*	*	15min. 38sec	14min. 23sec	*	*	1	1
			370	348,4	351	843	*	*	14min	13min. 35sec	*	*	1	1
Test17	1200	8	485	479	1020	1001	*	*	23min. 48sec	23min. 31sec	*	*	2	1
			485	479	131	808	*	*	16min. 51sec	16min. 50sec	*	*	1	1
Trans-territory network	CTPM : 392	11	32,4	32,4										
	OPTYMO: 112	5	24,8	8,5										
	SNCF : 3	5	3	2										

Table V: Tests Mesurements for different versions of ALGO APD_CPS applied on the random data generator

DOI : 05.IJIS.2016.1.1