First Digital Tunisian Louage's Transportation Solution

Mohamed Hassine*, Philippe Canalda*, Idriss Hassine*
Emails: mohamedhassinef35@gmail.com, philippe.canalda@femto-st.fr, drisshassine@yahoo.fr
*Institut FEMTO-ST (UMR CNRS 6174) – Centre de Développement Multimédia NUMERICA
1, Cours Louis Leprince-Ringuet, 25200 Montbéliard, France

***Abstract*—The Louage is a Tunisian collective transportation service for intra and inter-governorates' displacement. It completes the global public transportation offer. IT operates as triggered bus lines, and also as flexible's one. In this study, we propose the first definition of Louage's problem, a the first multi-criteria and multi-objectives modelling of this problem. We propose a simplified modeling of the optimization problem underlying the Louage's service, and an implementation of the exact resolution using a greedy algorithm. The solution's evaluation is based on real cases, extracted from the Louage's service set up on the Tunisian territory. The GILA solution proposed is dynamic with multi-constraints and mono-objective, being based on a Greedy Incremental Louages's Algorithm. The evaluation of specific instances show the feasibility to operate an Information Communication Technology (ICT)-like Louage service as an Intelligent Transport System(ITS) solution, in a decentralized manner as in an overall and instantaneous centralized manner.**
Key words: incremental Louage, exact approach, Intelligent transport system, greedy algorithm.

I. Introduction

Transportation is a key-problem in our daily-life when moving for work, study and shopping, moving alone or accompanied. Displacements can be categorized according to points of interest such as home, work, school and shopping malls. They can also being distinguished through personal vehicle used or using public transportation modes. Among modes of transportation, in Tunisia it exists the Louage which is as effective transportation alternative for Tunisians. So far, the digital market for ITS lack of a solution that manage the Louage, though widely used in Tunisia.

There is no TIC based solution to manage the Louage's service in Tunisia. That is probably due to the particular assignment process between drivers and passengers, and the legislative and operational rules being applied to integrate such a service as a complement into public transportation offers (plane, train, bus, urban taxi, peri-urban collective taxi). When arriving at the Louage's station, a driver registers his line (one among three) to an operator, candidate to be launched. For the sequel we designate by an offer request $Offer$ this operation, and we designate by a demand's request *Demand* the booking action of a passenger. The matching between $Offer$ and *Demand* is performed by the operator while respecting the order of the requests, numerous constraints __ among heterogeneous fleet capacities, comfort, prices, declarative lines and heterogeneous services__, and multi-objectives. The result is an ordered list of alternative organizations, similar to itinerary generations, with assignments. Each organization consists of a trips' list and there are as many trips as vehicles that will leave their louage's station to realize their lines-itinerary. The best organization that respect a bulk of constraints and reaches a set of objectives (maximization of the occupancy rate, minimization of trips times ...) is validated and chosen as an incremental constraint for the next matching process. That illustrates the incremental dimension of the louage's system (see figure 1 page 3). Up to now, the service described above is not automated. It is up to the operator of each Louage's station to implement this process, and when this operator is not present, this process is devoted to the louage's driver.
In the sequel we discuss the first two aspects (definition and implementation) of an automated Louage's solution based on multi-constraint and multi-objective modeling. Our objective is to make this mode more flexible to use and more profitable for all human actors participating in the system (drivers, operators, passengers). Our long-term ambition is to distribute the idea of The Louage all around the world, as a framed solution for the individual vehicle in collective use and complementary to a global public transport policy.

II. Tunisian Louage's description

The Louage is a public transportation mode widely used in Tunisia. There are several stakeholders in this system : the legislative state, the drivers with their habilitated cars, the users, the operator and the network of taxis. The former organizes the field of the transport service. It defines lines and specifies the itinerary of the different trips. A driver and vehicle couple can be habilitated until three lines (an itinerary composed of source and destination positions, plus vias for a line of intra-governorates' Louage. These lines are specified in the driver's exploitation card. There are two types of actors, the driver who is the service provider, and the passenger who is the applicant for the service. The major objective of the customer is to reach a particular destination, when the driver is to fill his vehicle. The operator, if present, is the equivalent to a mediator. Its role is to organize the trips by matching drivers and passengers, and then start its course when its car is fully occupied. In a TIC-based solution, the operator corresponds to the system's administrator. The latter the vehicles' network is comosed of vehicles with more than six seats but less than thirty[9].
There are two types of Louages, firstly the white louage with

red tape. This louage performs a connection between several governorates.

It starts from a big city, and serves another big city (ex Tunis-> Nabeul or Sousse->Sfax). It leaves once it is full, and only stops at final destination. Secondly, there is the white louage with blue tape. This louage circulates into a unique governorate. It operates shorter trips and serves large localities and medium intermediate localities. It can stop at fixed points (or vias) that are specified in the line of the driver's operating card.

III. State of the art : problems, operational services, methodologies, techniques

First we introduce the major transportation systems based on the use of individual vehicles (IV for short), and then we rely on the Tunisian Louage's system description. Second, we present succinctly the different combinatorial problems applicable to the field of on demand intelligent transportation. Third, we end with The Louage's problem positioning in relation to the previous items.

A. On Demand Transportation services

The Dial a Ride Transportation (DRT) is a transport mode triggered as a result of the passenger's request(s). There are several modes of DRT in the world:
[*Individual taxi service:*] the individual taxi is a particular car with up to five seats, including the one for the driver. It is equipped with a taximeter, which calculates the cost of a trip. This service is public.
[*shared taxi service:*] the shared taxi is a car with six to nine seats, including the driver's one. This service can be performed either on a a specific territory or on three lines maximum (depending on country legislation), and each line connects two points located within a single urban area.
[*Rural transportation service:*] This service is similar to shared taxis, but it is performed into rural areas.
[*Uber [10]:*] it is a private service that offers contacting users with drivers performing private transportation services on demand [1].
[*Transport Car with Driver (TCD) [11]:*] This is a private transport car with driver, which offers between 4 and 9 seats, including the driver's. TCD can not support a client unless the driver can prove a client's reservation. At the end of the trip, the driver must return to the operator's park, unless it justifies another reservation. Table 1 presents a comparison of public transportation services using the IV. The criteria presented are: vehicle function, coverage area, availability and cost per Km transported. The Louage's service and the taxi offer propose the best time availability. Louage is the best in terms of spatial availability as it covers inter-urban and inter-governorates territories. Louage's rates are fixed by the state, like for taxis, and public transport ([12]). In Tunisia prices practiced are very fair.
This table shows that triggered transports vary in terms of price, and in Tunisia Louage's price per km reaches 11% of the commonly practiced european uber price. It also shows a lack of available offer based on IV, serving the interurban and inter-region in general. Louage in Tunisia complete pragmatically the offer of trains, planes and buses, and offers a more reliable public service than TCD and better control than Uber.

[1]The parent company is based in the Californian city of San Francisco, and in 2015 its applications are sold in over 310 cities worldwide.

transportation service based on individual vehicle	*function of vehicle*	*area(s)*	*availibility*	*cost per Km per passenger transported(euros)*
individual taxi	vehicle service	urban area	++	0,16
shared taxi	vehicle service	urban area	++	0,06
rural transport	vehicle service	rural area	++	0,05
TCD	private vehicle	urban area	+	2.0
Uber	private vehicle	urban area	+	1.4
Louage	vehicle service	urban+rural area	+++	0.04

TABLE I
Comparative table of public and private transport services based on the IV in Tunisia an their main concurrent in the world.

The louage is therefore a combination of DRT and TC. The praticed prices are very competitive (0,04 euros per Km traveled). This service is reactive and available between 5am to 8pm. It covers almost the whole Tunisian territory by offering a complementary service to the public transportation (bus, train, plane...). This probably makes it the best among all Tunisian public transport's offers.

B. Inventory of compliant combinatorial problems

We briefly introduce major combinatorial problems, possibly targeting the Louage's problem.
The **Traveling Salesman Problem** consider a group of cities. It consists in visiting once each city (a single time) while minimizing travel distances and returning to the starting point. This is an NP-complete problem which can derivate to N-TSP problem where the objective is also to minimize the number N of vehicles required. An exact solution remains limited to small size (less than 20 trips).
The **Vehicle Routing Problem** involves planning of delivery routes with lower cost to serve a set of geographically dispersed customers, while respecting the vehicle capacity constraints and time windows. This combinatorial optimization problem has, both the characteristics of resource allocations (expense distributions in trucks: bin packing), and sequence of itinerary construction (TSP).
The **Problem of planning a Dial-a-Ride** service is known as the problem of access to a turn (DARP) [1]. This is a specific version of the pickup and delivery problem with time windows, which comes from the vehicle routing problem descriped in Dantzig et Ramser (1959) [2]. It consist to serve as a set of requests using a fleet of vehicles.
The **Demand Responsive Transportation Problem** is a new mode (or service) of transportation that aims to make more flexible transport system [3]. It is an intermediate mode between bus and taxi forwhich itineraries are either fixed either adaptative but considering a set of stops (composing a flexble line). DRT is suitable for rural areas where transit service is not always available.

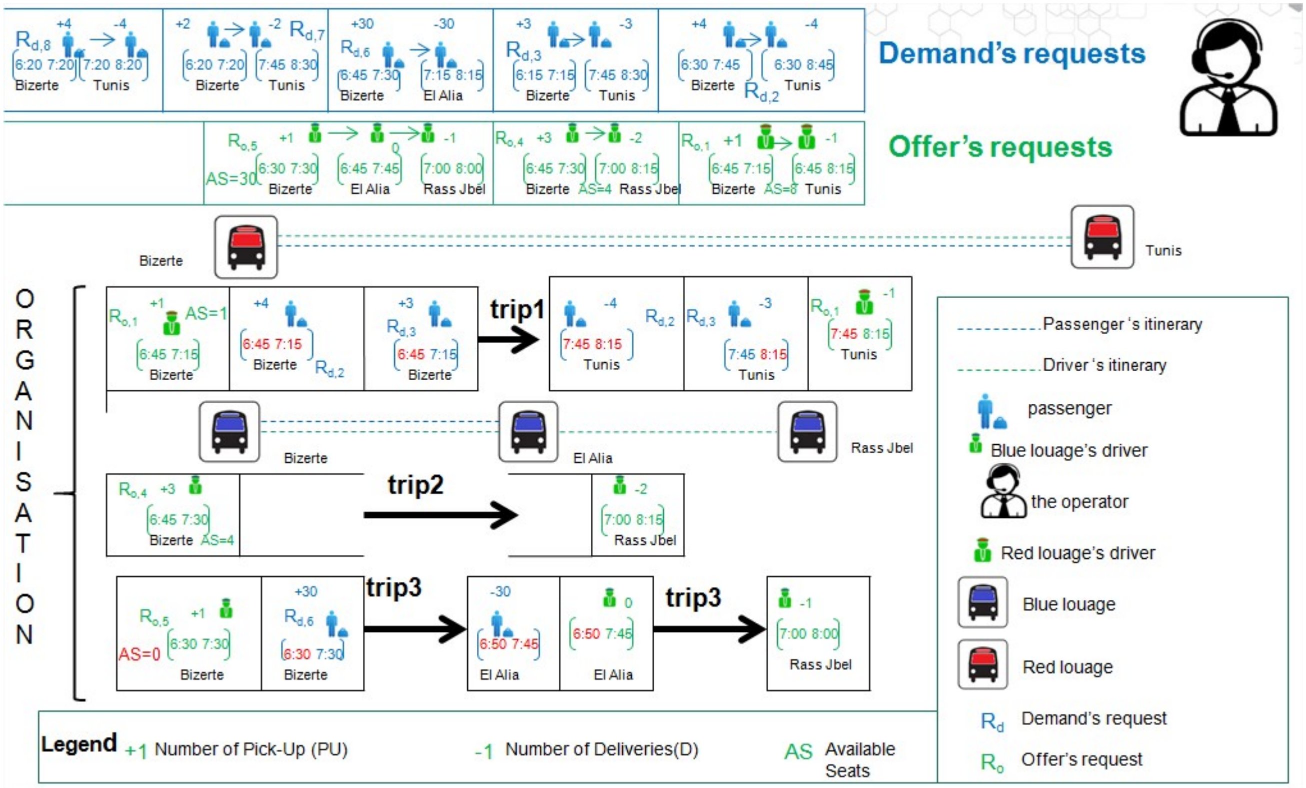

Fig. 1. Assignment's illustration between passengers and drivers at a red and blue Louage's station.

With the Louage, the source position does not coincide with the destination. The Louage has to follow the shortest itinerary, and not necessarily to stop at each crossing point. So the TSP is not fully compliant with the Louage's problem. For the Louage, the subject of delivery is the passenger. As for the VRP, it also has dynamic time windows with multi-deposits. In definitive, the classical VRP is not well suited to the resolution of our problem. With the DARP, an offer request can have Vias, but also the deman request may have Vias. DRTP is not suitable for the louage because we must consider additional properties (multi-deposit, split pick-up and delivery [4][5]).
None of the problems entirely fit the problem of Louage. So we can not apply the well known methods which address them efficiently. Indeed the Louage remains a problem limited in termes of constraints and objectives but it is a remarkable combination with specific constraints.

IV. Louage's problem modelling

In the sequel we propose the first modelling of the Louage Problem (LP for short). We formalize the main constraints per actor, and the objectives of LP' solution. In our formalization, we use an extended origin-destination matrix M_{o-d}. The rows and columns of this matrix present the different positions of the system (stations, vias). This matrix (figure 2) qualifies the displacement between two positions with various criteria among shortest time, shortest distance, price per place[12].

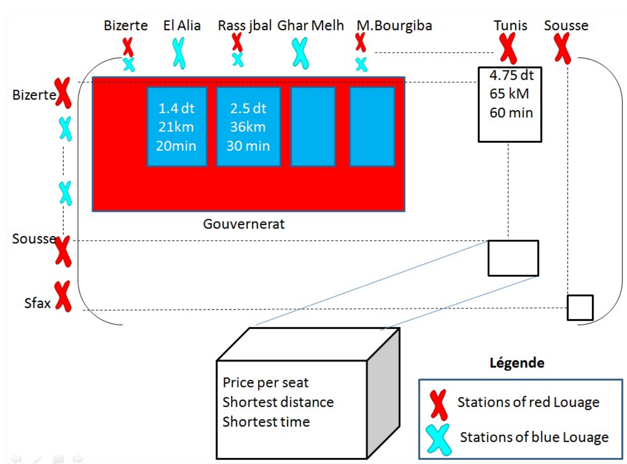

Fig. 2. Louage origin-destination matrix

A. Demand's request(s) of a passenger

Passengers are the applicants for the service. They address their requests *Demands* to the operator.

$$Demands = \begin{cases} R_{d,1} \\ \dots \\ R_{d,j} = \quad Itin_{d,j}, NS_{d,j}, A_{d,j}, State_{d,j} \\ \dots \\ R_{d,\overline{R_d}} \end{cases}$$

Demands is composed of $\overline{R_d}$ demand's requests.

$$Itin_{d,j} = \{Pos_{d,j,1}[h^-_{d,j,1}, h^+_{d,j,1}, PU_{d,j,1}, D_{d,j,1}], ..., \\ Pos_{d,j,\overline{Itin_{d,j}}}[h^-_{d,j,\overline{Itin_{d,j}}}, h^+_{d,j,\overline{Itin_{d,j}}}, PU_{d,j,\overline{Itin_{d,j}}}, D_{d,j,\overline{Itin_{d,j}}}]\}$$

designates the itinerary associated to passenger j. It is mainly composed by an origin position $Pos_{d,j,1}$ and a destination position $Pos_{d,j,\overline{Itin_{d,j}}}$. Each position is assigned from the earliest starting date $h^-_{d,j}$, the one at the latest $h^+_{d,j}$, the number of passengers boarding $PU_{d,j}$ (Pick-Up) and the number of passengers descending $D_{d,j}$ (Delivery).
$NS_{d,j}$ is the number of reserved seats[2].
$A_{d,j}$ is the amount paid by the passenger j (cf. [12]).
$State_{d,j}$ designates the state of the demand's request j among : initialized, validated (and normalized), matched (proposition), contractualized, running trip, executed or not executed, and archived (see figure 3).

B. Offer's request(s) of a driver

Drivers intervene in the system by declaring their offers' requests *Offers* to an operator, if present.

$$Offers = \begin{cases} R_{o,1} \\ \dots \\ R_{o,i} = Itin_{o,i}, D_{o,i}, V_{o,i}, AS_{o,i,initial}, AS_{o,i,current}, State_{o,i} \\ \dots \\ R_{o,\overline{R_o}} \end{cases}$$

Offers is composed of $\overline{R_o}$ offer's requests. Similarly to the previous section, $Itin_{o,i}$ designates the itinerary of driver j:

$$Itin_{o,i} = \{Pos_{o,i,1}[h^-_{o,i,1}, h^+_{o,i,1}, PU_{o,i,1}, D_{o,i,1}], ..., \\ Pos_{o,i,\overline{Itin_{o,i}}}[h^-_{o,i,\overline{Itin_{o,i}}}, h^+_{o,i,\overline{Itin_{o,i}}}, PU_{o,i,\overline{Itin_{o,i}}}, D_{o,i,\overline{Itin_{o,i}}}]\} \quad (1)$$

$D_{o,i}$ is the driver associated to the offer $R_{o,i}$ with $D_{o,i} \in [1, \overline{D}]$.
We define the set *Drivers* such as
$Drivers = \{Driver_d, d \in [1, \overline{D}]\}$ with
$Driver_d$=(registration number, type of permit, lines$_d$...).
$V_{o,i}$ is the vehicle associated to the offer $R_{o,i}$ with $V_{o,i} \in [1, \overline{V}]$.
We define the set *Vehicles* such as
$Vehicles = \{Vehicle_v, v \in [1, \overline{V}]\}$ with
$Vehicle_v = (capacity, insurance, consumption, drivers...)$.
$AS_{o,i,initial}$ is the number of available seats at the initialization of request offer i with $AS_{o,i,initial} \leq$ capacity of $V_{o,i} - 1$ (the seat occupied by the driver).
$AS_{o,i,current}$ is the current number of available seats of request offer i. This value takes into account the matched demand's requests[3].
$State_{o,i}$ is the state of the offer i. The state differs from demand's state with the partially construalized state which corresponds to a current offer with $AS_{o,i,current} > 0$. Figure 3 represents the transition-state diagram of a dynamic Louage's process. When a passenger expresses a new demand request

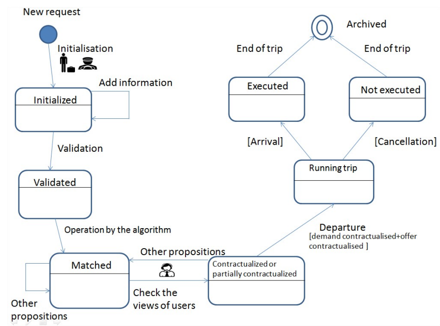

Fig. 3. Louage's transition-state diagram

(respectively offer request expressed by a Louage's driver), the request is in the status "initialized". As such, the request's information can be modified. The following validation step normalizes the dynamic time windows, and verify if the "normalized" request is correct according to itinerary constraints and point-to-point displacements. If "validated", the "new" request (or a group of sorted new requests) within, or without, a set of "contractualized/partially-contractualized" offers and demands, is matched. So that the matching algorithm operates and proposes an organization (constituted of trip) to be partially contractualized (only for offer, when it remains available seats) or contractualized (for all demands and for offers having no more available seats). The matching algorithm may be incrementally applied when new request(s) occurs. If passenger(s), driver and operator accept this proposition, states of matched requests become either "contractualized" for a demand, or "contractualized" or "partially contractualized" for an offer. An offer request's status is "contractualized" not only when all seats are reserved, but also when all available seats are assumed by the driver (this happens when the current date is greater than the latest date of the departure). Then, the request status become "running trip". Afterwards, depending on the cancellation of an offer request or the achievement of the trip, one have the two corresponding states "not executed" or "executed". Finally, at the end of the trip, the state becomes "archived".

C. Operator and organizations

A solution to the Louage's problem may be a list *Orgs* of a number $\overline{Orgs}$ possible organizations [4]. Each organisation $Org_{t \in [1, \overline{Orgs}]}$ is formed by a trips' list. Each trip is composed of an offer with a current status, a list of contractualized demands (this list may be empty if the offer request is only validated)

[2]A Louage's customer pay for his personal seat, his acompanying persons, and eventually the places not yet occupied, so that the line's vehicle can nstart as soon that all place are paid or paid and occupied. A driver can decide to leave his origin station to avoid an exceptional long waiting time, at lunch time, or for shorten the last day-trip.

[3]At a certain *time* of the system or at any *state* succeeding the initialisation state request.

[4]Our formalism proposes several solutions but, our coming implementation only produces one solution at each invocation.

and a proposed generated/computed itinerary.

$$Orgs = \begin{cases} Org_1 \\ \dots \\ Org_t = \begin{cases} Trip_{t,1} \\ \dots \\ Trip_{t,k} = Itin_{t,k}, Ro_{t,k}, ListRd_{t,k}, \\ State_{t,k}, OS_{t,k}, AS_{t,k} \\ \dots \\ Trip_{t,\overline{Org_{Trip}}} \end{cases} \\ \dots \\ Org_{\overline{Orgs}} \end{cases}$$

$Itin_{t,k}$ is the kth itinerary of the organization Org_t. If the status offer is validated, the itinerary k coincides with the normalized offer's itinerary. $Itin_{t,k} = \{Pos_{t,k,1}, \dots, Pos_{t,k,p}, \dots Pos_{t,k,\overline{Itin_{t,k}}}\}$.
The initial and destination positions of this itinerary coincide with those of the initial offer. The intermediate positions corresponds, one-to-one, to the positions belonging to positions issued from contractualized demands itineraries.
$Ro_{t,k}$ is the associated offer to $Trip_{t,k}$. The status of the request becomes contractualized when the reserved seats (of all the demands) are equal to the initial available seats of the vehicle's offer.
$ListRd_{t,k}$ is a list of matched demand's requests
$ListRd_{Trip,k} = \{Rd_{Trip,k,1}, ..., Rd_{Trip,k,l}, ..., Rd_{Trip,k,\overline{R_{Trip,k}}}\}$.
$State_{t,k}$, the trip's state.
$OS_{t,k}$, the maximum occupied seats along $Trip_k$ itinerary. This number is strictly less than the associated vehicle's capacity.
$AS_{t,k}$, the minimum number of available seats along trip's itinerary $Trip_k$ with $AS_{t,k} \in [0, AS_{o,i,initial} - OS_{t,k}]$.

D. Multi-constraints to be satisfied

Normalization of dynamic time windows An initialized offer's request becomes valid after the consistency check of its time windows. We pose $M_{o-d}(Pos_p, Pos_{p+1}).t$ the trip's duration between the position p and the position $p+1$. The sum of the earliest departure's date (resp. the latest departure's date) of the position p with the trip's duration is less or equal than the earliest departure's date (resp. latest departure's date) of the position $p+1$.

$$\begin{aligned} \forall \quad & R_{o,i\in[1,\overline{R_o}]} \in Offers \quad \wedge \quad State_{o,i} = \text{"validated"}, \\ & \forall \quad p \in [1, \overline{Itin_{o,i}}],\ h_p^+ + M_{o-d}(Pos_p, Pos_{p+1}).t \le h_{p+1}^+ \\ & \qquad \wedge\ h_p^- + M_{o-d}(Pos_p, Pos_{p+1}).t \le h_{p+1}^- \end{aligned} \tag{2}$$

This constraint remains valid $\forall \quad R_{d,j\in[1,\overline{R_d}]} \in Demands$.
Expoitation card lines and computed itineraries We define by $Lines_d$ the lines' list associated with a driver d with $d \in [1, \overline{D}]$, $\overline{D}$ the size of D the set of drivers, and $l \in [1,3]$ with 3 the maximum lines' number of any exploitation's card.

$$\begin{aligned} & Lines_d = \{Lines_{d,l}\} \quad with \\ & \quad Lines_{d,l} = Pos_{l,1}[h_1^-, h_1^+], ..., Pos_{l,p}[h_p^-, h_p^+], ..., \\ & \qquad Pos_{l,\overline{Lines_{d,l}}}[h^-_{\overline{Lines_{d,l}}}, h^+_{\overline{Lines_{d,l}}}] \end{aligned} \tag{3}$$

$Lines_{d,l}$ is similar to an itinerary modulo the Pick-up and Delivery attributes (cf. 1). We define by *Lines* the system lines' list. This list is the union of all the lines associated with the drivers.

$$Lines = \{Lines_{nl\in[1,\overline{Lines}]}\} = \bigcup_{d=1}^{\overline{D}} \left(\bigcup_{l=1}^{\overline{Line_d}} Lines_{d,l} \right) \tag{4}$$

The line's maximum number operated by a driver is $NbMaxLpD$. This number can not exceed 3 [5].

$$\forall \quad d \in [1, \overline{D}], \quad \overline{Lines_d} \in [1, NbMaxLpD] \tag{5}$$

Preservation of the positions' orders in the itinerary For each organization $Org_{t\in[1,\overline{Orgs}]}$, there will be an appropriateness between the positions' order of the trip's itinerary and the positions' order of the initial itinerary of the associated offer and all eventual matched demands (if any). Hence, the first itinerary's position (resp. the last one) of the offer coincides with the first position (resp. the last one) of the proposed itinerary in the tth organization's trip.

$$\begin{aligned} \forall \quad & Itin_{o,i\in[1,\overline{R_o}]} \in R_{o,i}, \quad \forall \quad Trip_{t,k} \in Org_t \\ & If \quad R_{o,i}.State \subset \{partially contractualized, contractualized\} \\ & \quad So\ \exists\ Itin_{t,k} = Pos_{t,k,1}, \dots, Pos_{t,k,p}, \dots, Pos_{t,k,\overline{Itin_{t,k}}} \\ & \qquad such \quad as \begin{cases} Pos_{t,k,1} = Pos_{o,i,1} \\ Pos_{t,k,\overline{Itin_{t,k}}} = Pos_{o,i,\overline{Itin_{o,i}}} \\ (\forall e, f \in [2, \overline{Itin_{o,i}} - 1] \quad and \\ \quad \forall g, h \in [2, \overline{Itin_{t,k}} - 1] \\ (e > f, \quad g > h), \\ \nexists (Pos_{t,k,g} = Pos_{o,i,f}) \quad and \\ \quad \nexists (Pos_{t,k,h} = Pos_{o,i,e}) \end{cases} \end{aligned} \tag{6}$$

This constraint is applied also $\forall \quad R_{d,j\in[1,\overline{R_d}]} \in Demands$. However the different positions of the demand will be inserted between the origin position and the destination position of the trip's itinerary.
Vehicle fleet with heterogeneous capacities In any offer, the number of initial available seats must be strictly less than the capacity of the associated vhicle.

$$\forall \quad R_{o,i} \in Offers, \quad AS_{o,i,initial} < Vehicle.cap \tag{7}$$

Similarly, in an organization proposed by the system, the maximum occupied seats' number during the trip [6] must be strictly less than the capacity of the associated vehicle.

$$\forall \quad Trip_k \in Org_t, \quad OS_{t,k} < Vehicle.cap \tag{8}$$

Atmost one request appearance in any $Trip_{t,k}$ of an Org_t A demand cannot be matched in two different trips of a same organization. As a consequence, a demand's request with 4 reserved seats will not be matched with 2 vehicles (that mean 2 offers in a same organization). As a property deriving from this constraint, is the no occurence of a partially contractualized state in any demand appearing in a trip.

$$\begin{aligned} \forall \quad & Org_{t\in[1,\overline{Orgs}]} \in Orgs, \forall \quad Trip_{k\in[1,Trip_{\overline{Org_t}}]} \in Org_t, \\ & \forall \quad Rd_{t,k,l\in[1,\overline{Rd_{t,k}}]} \in ListRd_{t,k}, \\ & \qquad Rd_{t,k,l}.State = \text{"contractualized"} \end{aligned} \tag{9}$$

Similarly, an offer appears exactly in one trip of each organization unless this offer is not yet validated.

[5]This lines' number operable by a driver is fixed by the law number 33 edited on 19/04/2004 relating to the road transport's organization [9].
[6]At each stop of the itinerary of a vehicle, we calculate the number of occupied seats.

E. Considered objectives

Objectives are financial, qualitative and quantitative.

Maximizing the occupancy rate of all the vehicles of an Org_t. The primary objective is to maximize the vehicle occupancy. That corresponds to minimize the number of unoccupied seats in vehicles of at least validated offer's request. It is also equivalent to maximize the numbers (OR) of Pick-Up at all itineraries' positions constituting the trips of each Organization.

$$\forall \quad Org_{t\in[1,\overline{Orgs}]} \in Orgs,$$
$$\forall \quad Itin_{t,k\in[1,\overline{Org_t}]} = Pos_{t,k,1},\ldots,Pos_{t,k,p},\ldots,Pos_{t,k,\overline{Itin_{t,k}}}$$
$$OR = \sum_{Trip=t_1}^{t_{\overline{Org_t}}} \sum_{p=1}^{\overline{Itin_{t,k}}} Pos_{t,k,p}.PU \quad (10)$$

Maximizing the number of contractualized request This objective consists in maximizing the number of contractualized offers and demands, i.e. maximizing the sum of $R_{o,con}$ and $R_{d,con}$ or minimizing the sum of $R_{o,!con}$ and $R_{d,!con}$. This goal must be achieved by respecting the requests' FIFO list (the order) of offers and demands expressed to the operator of Louage's service at the station.

Maximizing the standard deviation of users's waiting time and minimizing trips' total times We pose (WT) the Waiting Time of a passenger or a driver in a proposed computed itinerary. The waiting time delay, at a station, must have the widest margin to guarantee a passenger to access and realize his trip. Consequently, this objective participate to robustness, as to maximize the wating times' standard deviation in the itinerary (noted by σ).

$$\sigma = \sqrt{\left(\frac{1}{(\overline{Itin_{t,k}})}\right) \times \sum_{p=1}^{\overline{Itin_{t,k}}} \left(WT_p - \sum_{p=1}^{\overline{Itin_{t,k}}} \frac{WT_p}{(\overline{Itin_{t,k}})}\right)^2} \quad (11)$$

Also, we pose TT the Total travel's Time of trips which compose an organisation. Our goal is to minimize TT.

$$\forall \quad Org_{t\in[1,\overline{Orgs}]} \in Orgs,$$
$$\forall \quad Itin_{t,k\in[1,\overline{Org_t}]} = Pos_{t,k,1},\ldots,$$
$$Pos_{t,k,p},\ldots Pos_{t,k,\overline{Itin_{t,k}}}$$
$$TT = \sum_{t=t_1}^{t_{\overline{Org_t}}} \sum_{p=1}^{\overline{Itin_{t,k}}} M_{o-d}(Pos_{t,k,p}, Pos_{t,k,p+1}).t \quad (12)$$

Maximizing the drivers's gain Since the price paid by the passenger is fixed, our objective consists to maximize the drivers' gain. We define by $GainD$ this gain.

$$GainD = \sum_{t=t_1}^{t_{\overline{Org_t}}} \sum_{l=1}^{\overline{Rd_{t,k}}} Rd_{t,k,l}.A \quad (13)$$

V. Louage matching process illustration

In the sequel, we illustrate the Louage's matching process. To illustrate it, we focus on the trip1 generation of illustration of Figure 1 (refer to page 3). Initially, all initialized requests are inserted into a FIFO lists. The very first initialized trip consists of the origin and destination positions of $R_{o,1}$. This is an offer to vehiculate 8 passengers from Bizerte to Tunis. Then, the next 2 initialized requests are demands $R_{d,2}$ and $R_{d,3}$ which fit into the first trip as they share the same origin and the same destination, with coinciding time windows, coherent requested seats with vehicle's available seats (associated with $R_{o,1}$). $R_{d,2}$ occupies 4 seats and $R_{d,3}$ 3 seats. After the matching, $R_{d,2}$ and $R_{d,3}$ become contractualized and $R_{o,1}$ becomes partially contractualized since there is still a free seat. At this step, requests $R_{d,7}$ and $R_{d,8}$ can not be matched to $R_{o,1}$, and therefore to the trip 1, because they require too much seats compared with available one's. Similarly to the previous trip, the second trip is initialized with the origin and destination positions of the second offer. This is an offer from Bizerte to Ras Jebel with $AS = 4$. No valid demands share the positions involved. So, this offer remains pending (this offer is just in validated status). When treating $R_{o,3}$ which proposes to desserve Bizerte and Rass Jbel through a via (El Alia), $R_{d,6}$ becomes matched. At this point of the illustration, one notice the multiple and iterative call of a matching algorithm. Contractualization at the end of a first call becomes a constraint on any subsequent calls. The proposed result consists of a single organization, itself composed of three trips. The first is partially contractualized, the second is only validated and the third is contractualized.

VI. Greedy Incremental Louage's Algorithm

In our modelling of the Louage's Problem, we presented a multi-objectives approach based on maximizing the occupancy rate, minimizing the total times of organization's trips, etc., but our proposed GILA algorithm is a single-objective algorithm. This mono-objective is achieved by prioritizing four objectives. The main objective is to maximize the number of contracted requests respecting their arrival's order [7] in the matching process. GILA generates one organization during each iteration, the first one which reaches this single objective and also respects the constraints cited at the formalization stage. The algorithm 1 GILA has as input a list of offers initialized by drivers, a list of demands initialized by passengers and a facultative constraint organization composed by a list of trips already generated, proposed and by the way either contractualized or partially contractualized.

The organization constraint is empty at the first occurrence of GILA's call. Then GILA handles a previous selected organization considered as constraint. The incremental process involved corresponds to iterative calls of GILA by accepting a set of new validated requests with a contextual [partially] contractualized organization as input.

At the very beginning of the algorithm, function **validateInitializedRequests()** is launched and returns a validated requests list respecting the normalization's constraint of time windows (constraint 2). The first organization is constituted by the existing organization *constraint*. If this organization does not existe, a new organization is created. The (new) validated offers are inserted as trips to the organization. The matching process begins by browsing trips' lists and the corresponding list of validated, but not contracted, requests (offers). For the first trip $PtripOrgs$, we position a pointer $PitinOrgs$ along the itinerary, and we browse the list of validated demands pointing

[7] GILA which maximizes the number of satisfied requests by respecting the order constraints is equivalent to an exact resolution satisfying the multi-constraints of the defined louage problem.

Algorithm 1 GILA($Offers_{init}$ O_{init}, $Demands_{init}$ D_{init}, $Organization$ Org)

```
//validation/normalization of initialized demands and offers
O_v ← ValidateInitializedRequests(O_i) //constraint 2
D_v ← ValidateInitializedRequests(D_i) //constraint 2
//initializing the list of organizations
if Org! = Null then
    Orgs = Concat(Orgs, Org)
else
    Orgs = new  Organization()
end if
//inserting (new) validated offers to the first organization
Orgs.Org_1 ← Concat(Org, O_V)
PtripOrgs ← Orgs
PitinOrgs ← PtripOrgs
PpositinOrgs ← PitinOrgs
PD_v ← D_v
while PtripOrgs! = Null  &&  !(PtripOrgs.state.equals("contractualized")) do
    while PD_v! = Null  &&  !(PD_v.state.equals("contracted")) do
        PPosD_v ← D_v
        while PPosD_v! = Null  &&  PpositinOrgs! = Null do
            //function that checks the insertion with respect to constraints 7,8
            if VerifValidLocalTW(PPosD_v, PpositinOrgs, PpositinOrgs.next())
            then
                PropagationOfTW(PPosD_v, PpositinOrgs, PpositinOrgs.next())
                //function that updates the itinerary with respect to constraint 6
                Update(PPosD_v, PpositinOrgs, PpositinOrgs.next())
                PpositinOrgs ← PPosD_v
                PPosD_v ← PPosD_v.next()
            else
                // local insertion contraint unvalidated
                PD_v ← PD_v.next()
                PPosD_v ← PD_v
            end if
        end while
        if PPosD_v = Null then
            //a demand has been inserted
            D_con ← D_con + PD_v
            D_v ← D_v − PD_v
            PpositinOrgs.OS = PpositinOrgs.OS + PD_v.SN
            PpositinOrgs.AS = PpositinOrgs.AS − PD_v.SN
            PD_v ← PD_v.next()
            PPosD_v ← PD_v
        end if
    end while
    PtripOrgs ← PtripOrgs.next()
    PitinOrgs ← PtripOrgs
    PpositinOrgs ← PitinOrgs
end while
return Orgs
```

to the first position $PPosD_v$ of the itinerary of the current demand PD_v. At this step, we call the function **VerifValidLocalTW()** which checks whether the position's inclusion is possible between two positions of the trip's itinerary.

Algorithm 2 Boolean VerifValidLocalTW(POS $PPosD_v$,POS $PpositinOrgs$,POS $PpositinOrgs.next()$)

```
if PpositinOrgs.h⁻ + M_{o−d}(PpositinOrgs, PPosD_v).t ≤ PPosD_v.h⁺ and Max(PPosD_v.h⁻, PpositinOrgs.h⁻ + M_{o−d}(PpositinOrgs, PPosD_v).t) + M_{o−d}(PPosD_v, PpositinOrgs.next()) ≤ PpositinOrgs.next().h⁺ then
    return  true
else
    return  false
end if
```

The property of this condition is held on Dynamic Time Windows' normalization, at each time a position candidate from a demand may be inserted between 2 positions of the trip's computing. If the insertion is considered as possible, a mixt dynamic propagation (forward and backward) of time windows is performed using the fonction **PropagationOfTW()**. Hence, after the insertion, the trip in construction remains normalized, that means that coherences between earliest date of departure of position $p-1$ (resp. latest date of the TW) and earliest date of arrival position $p+1$ (resp. the latest one's) is ensured even considering the position p having being inserted.

Algorithm 3 PropagationOfTW(POS $PPosD_v$,POS $PpositinOrgs$,POS $PpositinOrgs.next()$)

```
if PPosD_v.h⁻ + M_{o−d}(PPosD_v, PpositinOrgs.next()).t ≥ PpositinOrgs.next().h⁻ then
    /*forward propagation*/
    PpositinOrgs.next().h⁻ ← PPosD_v.h⁻ + M_{o−d}(PPosD_v, PpositinOrgs.next()).t
    PPosD_v.h⁺ ← PpositinOrgs.next().h⁺ − M_{o−d}(PPosD_v, PpositinOrgs.next()).t
else if PPosD_v.h⁺ − M_{o−d}(PpositinOrgs, PPosD_v).t ≤ PpositinOrgs.h⁺ then
    /*backward Propagation*/
    PpositinOrgs.h⁺ ← PPosD_v.h⁺ − M_{o−d}(PpositinOrgs, PPosD_v).t
    PPosD_v.h⁻ ← PpositinOrgs.h⁻ + M_{o−d}(PpositinOrgs, PPosD_v).t
end if
```

The itinerary of the trip is updated by applying the function **Update ()**. Once this position is treated (from the demand), we treat the next one. If the local insertion condition insertion is $false$, we evaluate if the insertion of this demand is feasible in another organization's trip. If all the positions of the demand are inserted, the demand request is matched with the running process' trip. We go on with the next request. when the processed trip becomes contractualized, or if we have already treated all the demands, we position $PtripOrgs$ on the next partially contratcualized, or only validated, trip, to proceed as above. Browsing through all the trips, we lead a new organization $Orgs$. The greedy aspect is inferred from the fact that the solution is built step by step and without backtracking, and is respectfull of constraints and hierarchical mono-objective. The first insertion of a demand in a trip, and thus within an offer, is revealed as the correct one. The insertion of a demand's position between 2 positions of the trip's itinerary in treatment is sometimes possible in multiple places. The last place of insertion appears, in the context of the constraints of this Louage's problem, to be always the best one. Among the reasons why we attest GILA's algorithm is a greedy version of LP, considering the mono-objective function which aims at maximizing the number of contractualized requests by respecting their order of expression to the operator.

GILA is incremental. During the first call or iteration, GILA only deals with initialized requests. During another iteration it_n $with$ $n>1$, the organization proposed by it_{n-1} is considered as a constraint to satisfy. The incrementality aims to satisfy no-matched requests and ameliorate other organizations' quality. Table II shows an example of execution of GILA, focussing on the computation of time windows (C), and subsequently the progressive elaboration of the trip 1 of our illustration. This execution demonstrates the dynamic aspect of GILA.

Positions of Trip1 (illustration 1)	$R_{o,1}.Pos_1$	$R_{d,2}.Pos_1$	$R_{d,3}.Pos_1$	$R_{d,2}.Pos_2$	$R_{d,3}.Pos_2$	$R_{o,1}.Pos_2$
Input datas −− *Initialized TW of Trip1′s requests*	$[6h45\ \ 7h15]_i$	$[6h30\ \ 7h45]_i$	$[6h15\ \ 7h15]_i$	$[6h30\ \ 8h45]_i$	$[7h45\ \ 8h30]_i$	$[6h45\ \ 8h15]_i$
Input datas −− *Normalized TW of validated requests*	$[6h45\ \ 7h15]_v$	$[6h30\ \ 7h45]_v$	$[6h15\ \ 7h15]_v$	[**7h30** 8h45]$_v$	$[7h45\ \ 8h30]_v$	[**7h45** 8h15]$_v$
Start of GILA −− *initialize trip with an Offer*	[6h45 7h15]					[7h45 8h15]
insert $R_{d,2}.Pos_1$ *and Forward TW′s propagation*	[6h45 7h15]	[6h30 **7h15**]				[7h45 8h15]
Backward TW′s propagation	[6h45 7h15]	[**6h45** 7h15]				[7h45 8h15]
insert $R_{d,2}.Pos_2$ *and Forward + Backward TW′s propagation*	[6h45 7h15]	[6h45 7h15]		[**7h45 8h15**]		[7h45 8h15]
insert $R_{d,3}.Pos_1$ *and Forward + Bacward TW′spropagation*	[6h45 7h15]	[6h45 7h15]	[**6h45** 7h15]	[7h45 8h15]		[7h45 8h15]
insert $R_{d,3}.Pos_2$ *and Forward + Backward TW′spropagation*	[6h45 7h15]	[6h45 7h15]	[6h45 7h15]	[7h45 8h15]	[7h45 **8h15**]	[7h45 8h15]
Contractualized trip −− **End of GILA**	$[6h45\ \ 7h15]_C$	$[6h45\ \ 7h15]_C$	$[6h45\ \ 7h15]_C$	$[7h45\ \ 8h15]_C$	$[7h45\ \ 8h15]_C$	$[7h45\ \ 8h15]_C$

TABLE II

GILA'S EXECUTION WHEN INSERTING TWO DEMANDS AND GENERATING (ILLUSTRATION 1) TRIP1'S ITINERARY

VII. FIRST LOUAGE'S INSTANCES SET

The new combinatorial problem formalisation, and the modelling of the greedy and incremental algorithm, require a specific set of instances we propose in the sequel. Table 3 presents synthetically 12 instances of LP. The first column describes the input datas of GILA's callings. 2 type of callings are proposed. The first iteration calling and the 2nd considering organisation contraints. An instance's configuration is defined as follows: $configOffer = \overline{R_o}/\overline{R_{o,i}}/\overline{R_{o,v}}/\overline{R_{o,c}}/\overline{R_{o,pc}}$; $configDemand = \overline{R_d}/\overline{R_{d,i}}/\overline{R_{d,v}}, \overline{R_{d,c}}$ with: $config.n$ is a configuration of an umpteenth iteration, $\overline{R_o}$ is the number of offers, $\overline{R_i}$ the number of initialized offers, $\overline{R_v}$ the number of validated offers, $\overline{R_{pc}}$ the number of partially contractualized offers and $\overline{R_c}$ the number of contractualized offers. We have also the same legend to explain the column two to six to qualify the organisation's result. For each 2nd call of GILA's configuration we consider also, as constraints, input requests that are either contractualized, partially contractualized or only validated. Twelve different configurations are evaluated. The first one consists of 10 initialized offers and 50 initialized demands. Gradually the other configurations allow to evaluate the scalability and to demonstrate the operational incrementality and the linear and greedy behavior of GILA. To show the scalability behavior of GILA, we vary the number of initialized offers corresponding to 20% of initialized demands. The results show the number of validated requests, partially contractualized requests and contractactualized requests obtained, as well as the algorithm's execution time.

The GILA'evaluation is based on a Java's implementation and a running on a common laptop (ASUS K56C). Three files compose the imput flows of GILA's implementation: an offers file, a demands one's, and optionnally an organization one's. Fig. 4 shows 6 curves. The red one shows the execution's evolution time compared to the number of initialized offers. The green curve shows the execution's evolution time compared to the number of initialized demands. The blue curve shows the execution's evolution time compared to the number of initialized offers and demands.

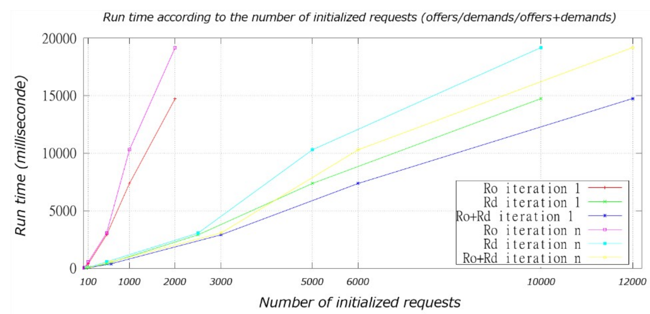

Fig. 4. Execution time's evolution for iterative/functional/scaling up instances of the Louage's problem

The blue curve can be seen as a concatenation of the two others curves. The observation of this curve shows a linear behavior. For 600 initialized requests (500 demands and 100 offers), the matching's execution time is 379 ms. Even for 3000 requests (2500 demands and 500 offers), the execution time remains operational with 2892 ms. The fact that GILA is incremental has no impact on operationnality. These results show that GILA is operational and can be deployed over the Tunisian Louage's network. Even a centralized version of GILA running for all Louage's stations in Tunisia is operational as the number of simulatneous Louage's requests overall the Tunisian territory is of the order of 10000. The other curves show the running time evolution when applying GILA in an incremental context. In these tests, we aggregate the same number of initialized requests and contractualized or partially contractualized requests, plus the organization result provided by previous call of GILA. For example, for 600 new initialized requests, we have also 600 contractualized one's and one organization constitued on either partially contractualized or contratcualized offers and associated contractualized demands. We notice the same linear behavior which confirms the theoretical complexity of the algorithm is linear (in order of O(n)) with $n = m \times p$, with m designates the number of offers and p the number of demands.

VIII. CONCLUSIONS AND PERSPECTIVES

We have proposed the first mathematical formalization of the Louage's problem in Tunisia, as a combination of multi-constraints and multi-objectives. We also presented GILA, a first implementation of the greedy and incremental matching algorithm. Our overall proposition constitutes a first automated

config test	$\overline{R_{o,c}}$	$\overline{R_{o,pc}}$	$\overline{R_{o,v}}$	$\overline{R_{d,c}}$	$\overline{R_{d,v}}$	ms
$config1 : configOffer = 10/10/0/0/0, configDemand = 50/50/0/0$	5	4	1	47	3	23
$config1.n : configOffer = 20/10/1/5/4, configDemand = 100/50/3/47$	10	7	3	96	4	27
$config2 : configOffer = 20/20/0/0/0, paramDemande = 100/100/0/0$	12	4	4	92	8	43
$config2.n : configOffer = 40/20/4/12/4, configDemand = 200/100/8/92$	24	10	6	187	13	50
$config3 : configOffer = 100/100/0/0/0, configDemand = 500/500/0/0$	49	10	31	364	136	379
$config3.n : configOffer = 200/100/31/49/10, configDemand = 1000/500/136/364$	120	26	54	882	118	538
$config4 : configOffer = 500/500/0/0/0, configDemand = 2500/2500/0/0$	268	48	184	2028	472	2892
$config4.n : configOffer = 1000/500/184/298/48, configDemand = 5000/2500/472/2028$	593	98	309	4416	584	3048
$config5 : configOffer = 1000/1000/0/0/0, configDemand = 5000/5000/0/0$	537	94	369	4088	912	7361
$config5.n : configOffer = 2000/1000/369/537/94, configDemand = 10000/5000/912/4088$	1075	186	739	8208	1792	10290
$config6 : configOffer = 2000/2000/0/0/0, configDemand = 10000/10000/0/0$	1075	186	739	8208	1792	14747
$config6.n : configOffer = 4000/2000/739/1075/186, configDemand = 20000/10000/1792/8208$	1613	278	2109	12328	7672	19183

TABLE III
12 INSTANCES' DESCRIPTION OF LOUAGE'S PROBLEM AND ITS QUALITATIVE RESULTS.

and numerical alternative to the operational and empirical practice of Louage's services in Tunisia. Our proposal allows us to foresee a first complete Louage's digitized solution. The algorithmic and incremental version aims at maximizing the number of contractualized requests respecting the FiFo order of demands and offers validates by an operator of the Louage's service. In more details, our study therefore proposed, first the data's formalization of multi-actors : drivers, passengers, vehicles with heterogeneous capacities, operator. Second it proposed a dynamic and incremental process to tackle a DARP-like problem with multiple entries: offer's requests, demands, the contractualized organization issued from a previous call. Different states model a more global incremental Louage's management and clarify the running process to match demands and requests, and also accompagny the execution of itineraries generated by GILA : initialized, validated, matced and proposed, partially contractualized, contractualizeded, executed,... Third, the constraints address dynamic time windows with a normalization of windows, Louage's line, preservation of positions' order, resources, previous engagements. Last but not least, the formalization of the multiple objectives amongs : maximizing the occupancy rate, maximizing the number of contractualized requests, minimizing the total times of trips and maximizing the drivers's gain. The implementation of the proposed incremental solution is based on the algorithm GILA solving exactly generation's problem of the matching between different systems' actors. This algorithm's implementation accepts as input data flows composed of offer's requests, demand's requests, and optionally previous contractualizations between partially contracted offers and contracted demands. This version manages dynamic time windows, source and destination itinerary for red louage's demands and offers, and a list of steps for offer's requests from blue louage. The fleet of vehicles is of variable capacity. The pick-ups and deliveries considered are partial. Another remarkable feature of our system is the automatic generation of distributable itineraries among the matching actors. This point is added to that of an incremental matching which makes it possible to iterate on partially contracted trips or only validated with a renewed or modified stream of offers or demands. We proposed an evaluation of our system to assert its ability to dynamically process cumulatively and centrally all louage's demands from the national territory. A fortiori, the incremental approach thus makes it possible to respond to the louage's operationality at these country's stations. Simulations made it possible to satisfy 20,000 demands and 4,000 offers, and for a response time of our system reduced to a few seconds. The complexity calculation is consistent with the gradual evaluation of input data. Our work constitutes a good evaluation of performance that could be compared with other heuristics for example.

Our future work is divided into two orientations. The first is to realize a mobile web and mobile system to be deployed in one or several Louage's stations (at the behest of the white label products of the company shareandmove solutions [7]). The second is to translate this operational system based on the operation of the individual vehicle into a global public transport system. This Louage is an alternative to the Uber approach, which is subject to tax and unfair competition when applied in different European countries (especially in France).

REFERENCES

[1] Cordeau, Jean. "The dial-a-ride problem: models and algorithms" Annals of Operations Research (2007).
[2] Dantzig, G. B., and Ramser J. H. "The Truck Dispatching Problem." Management Science 6.1 (1959).
[3] JennyMageen and John D. NELSON The evaluation of demand responsive transport services in Europe Journal of Transport Geography 11 (2003).
[4] Parragh S., Doerner K., and Hartl R. (2008a). A survey on pickup and delivery problems. Part I: Transportation between customers and depot. Journal fr Betriebswirtschaft (58), 21-51.
[5] Parragh S., Doerner K.. and Hartl R. (2008b). A survey on pickup and delivery problems. Part II : Transportation between pickup and delivery locations. Journal fr Betriebswirtschaft (58), 81-117.
[6] http://www.covivo.fr/
[7] http://www.shareandmove.fr/
[8] http://www.snt.com.tn/
[9] http://www.legislation.tn/ Law n 33, edited in 2004;arrłt order of the Tunisian Transportation Ministery, 2010, article n 11, page 296.
[10] https://www.uber.com/fr/
[11] http://www.vtcourses.fr/
[12] http://archive.jawharafm.net/tounsia/29364-exclusif–la-liste-des-nouveaux-tarifs-des-taxis-et-des-louages-utica.html