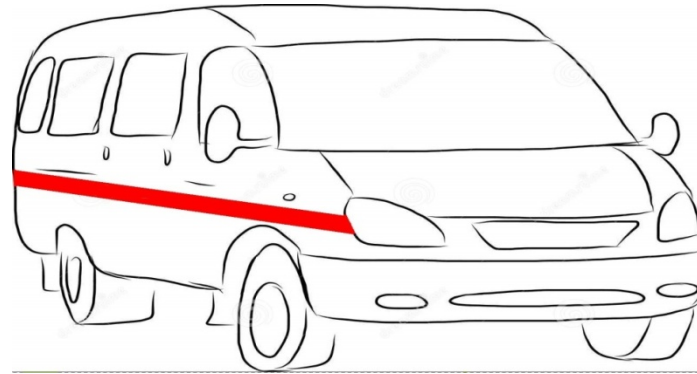




Louage incrémental : problème posé, étude de la solution combinatoire « gloutonne »



Réalisé par:

❖ **Mohamed HASSINE**
mohamedhassinef35@gmail.com

❖ **Philippe CANALDA**
philippe.canalda@femto-st.fr

❖ **Idriss HASSINE**
drisshassine@yahoo.fr

Sommaire

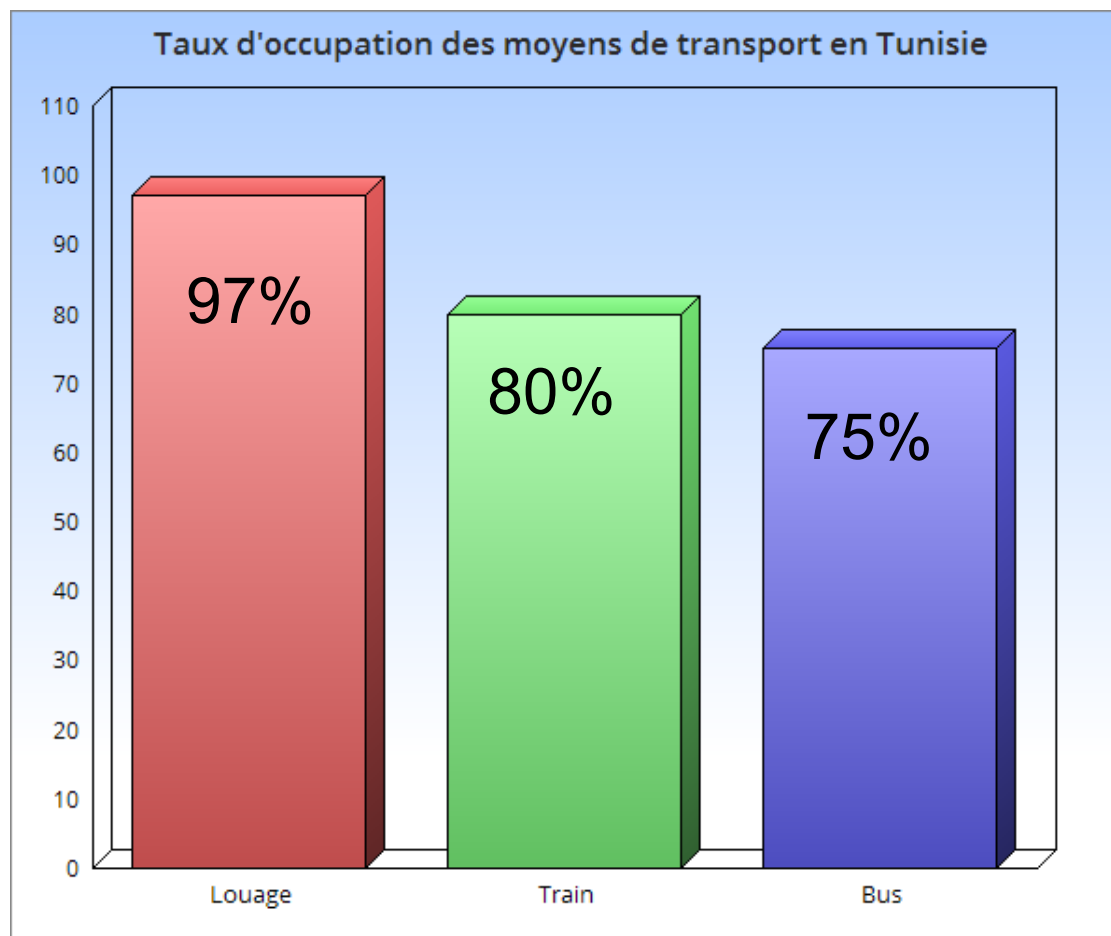
1. LE SYSTÈME DU LOUAGE EN TUNISIE
2. NOTRE PREMIÈRE FORMALISATION DU LOUAGE
 1. ÉTAT DE L'ART
 - 1.1 SYSTÈMES DE TRANSPORT ANALOGUES
 - 1.2 UNE ÉVOLUTION DES DARP, VRP, TSP
 2. REQUÊTE DEMANDE, REQUÊTE OFFRE ET ORGANISATIONS
 3. EXEMPLE PRATIQUE D'APPARIEMENT DES REQUÊTES
 4. LES MULTI CONTRAINTES
 5. LES MULTI OBJECTIFS
3. ALGORITHME PROPOSÉ ET TESTS
 1. GILA : GREDDY INCREMENTAL LOUAGES 'S ALGORITHM
 2. EXEMPLE PRATIQUE D'APPARIEMENT DES REQUÊTES EN APPLIQUANT L'ALGORITHME
 3. TESTS ET ANALYSE DES RÉSULTATS
4. CONCLUSIONS ET PERSPECTIVES

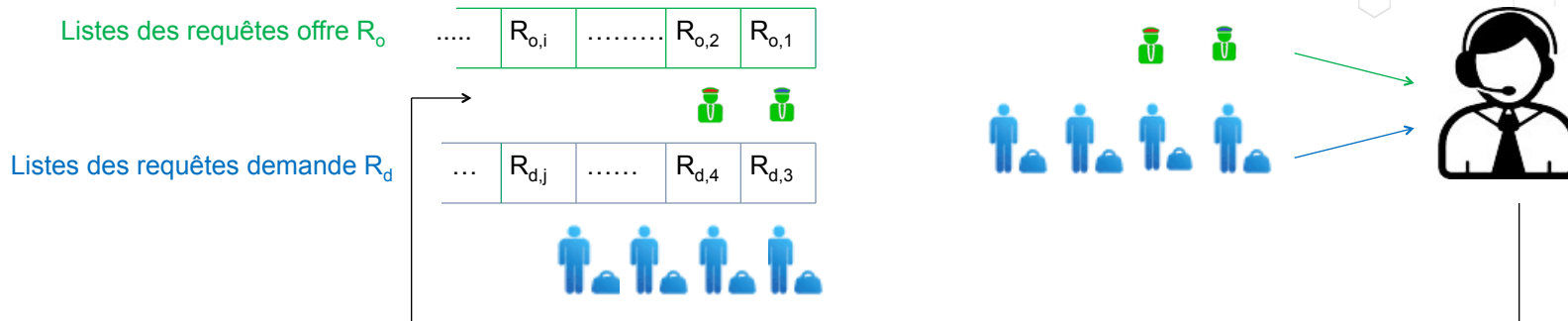


Complémentarité des offres de transport en Tunisie

Le louage est une réponse à une offre de transport publique couvrant l'ensemble du territoire tunisien

- 10000 véhicules de louage en 2014.
- 15 % du kilométrage parcouru dans le transport des personnes par des voitures individuelles est effectué par les louages.
- Un coût pas cher : 4.75 dinars / 60 km (i.e. 0,04€ le km).





✚ **Un chauffeur**, arrivant à la gare, se présente à une personne responsable dans la gare (**opérateur**), pour déclarer sa ligne qu'il souhaite mettre en service.

✚ Les passagers se présentent aussi à l'opérateur dans son guichet pour réserver son trajet et régler son (ou ses) billet(s).

✚ L'opérateur a pour charge de faire coïncider l'offre à la demande.

Service de taxi individuel



Service de taxi collectif



Uber



Service de transport rural



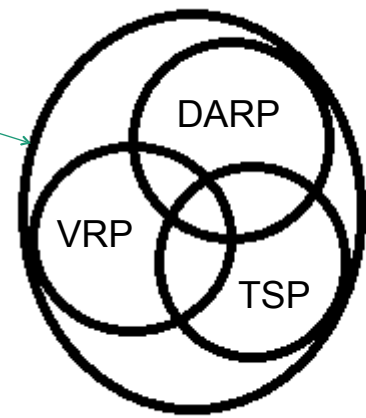
VTC



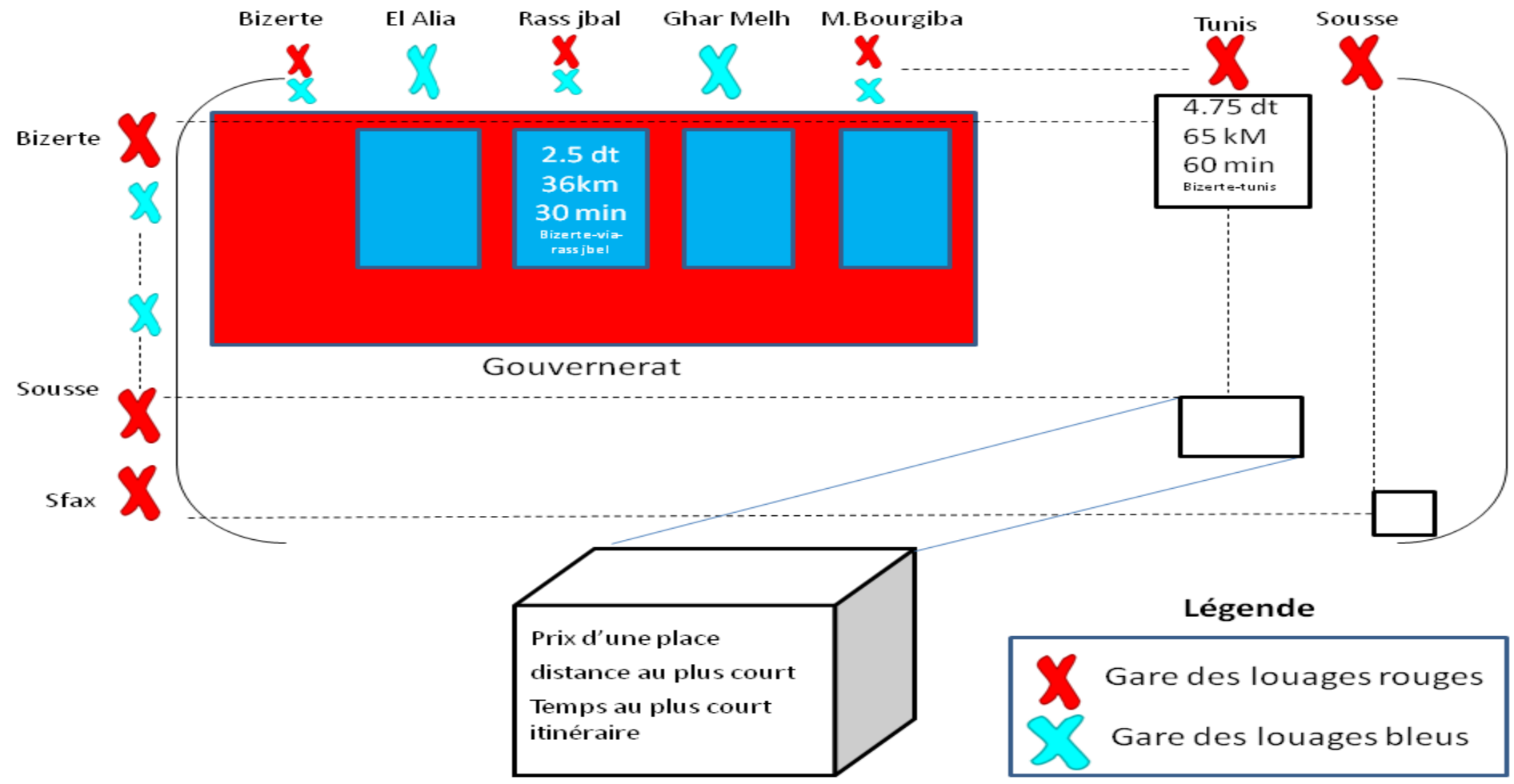
Le problème du louage est un problème multi contraintes et multi objectifs.

On distingue plusieurs problèmes combinatoires classiques appliqués au domaine du transport:
TSP: problème du voyageur de commerce
VRP: problème de tournées des véhicules
DARP: Dial a ride problem

Louage's problem



Matrice origine-destination



Les passagers sont les demandeurs du service, ils interviennent au système par l'envoi d'une liste des requêtes demande (Demands).

$$Demands = \begin{cases} R_{d,1} \\ \vdots \\ R_{d,j} = Itin_{d,j}, NP_{d,j}, M_{d,j}, State_{d,j} \\ \vdots \\ R_{d,\overline{R_d}} \end{cases}$$

$Itin_{d,j}$ désigne le trajet du passager.

$$Itin_{d,j} = \{Pos_{d,j,1}[h_{d,1}^-, h_{d,1}^+, PU_{d,1}, D_{d,1}], \dots, \\ Pos_{d,j,\overline{Itin_{d,j}}}[h_{d,\overline{Itin_{d,j}}}^-, h_{d,\overline{Itin_{d,j}}}^+, PU_{d,\overline{Itin_{d,j}}}, D_{d,\overline{Itin_{d,j}}}]\}$$

$NP_{d,j}$ c'est le nombre des places réservées.

$M_{d,j}$ est le montant payé.

$State_{d,j}$ présente l'état de la requête au cours de son traitement (initialisée, validée, appariée, contractualisée, en cours du voyage, réalisée, non réalisée et archivée).

✚ On définit par Chauffeur un ensemble contenant la liste de tous les chauffeurs tel-que

$$\text{Chauffeur} = \{ \text{Chauffeur}_c \text{ telque } c \in [1, \overline{C}] \} \text{ avec}$$
$$\text{Chauffeur}_c = (\text{matricule}, \text{type de permis}, \text{lignes}_c \dots)$$

✚ On définit par Véhicule un ensemble contenant la liste de tous les véhicules tel-que

$$\text{Véhicule} = \{ \text{Véhicule}_v \text{ telque } v \in [1, \overline{V}] \} \text{ avec}$$
$$\text{Véhicule}_v = (\text{capacité}, \text{assuré}, \text{consommation}, \text{conducteurs} \dots).$$

Les chauffeurs interviennent au système par l'envoi des requêtes offre (Offers).

$$Offers = \begin{cases} R_{o,1} \\ \vdots \\ R_{o,i} = Itin_{o,i}, C_{o,i}, V_{o,i}, PlacesL_{o,i,initial}, PlacesL_{o,i,actuel}, State_{o,i} \\ \vdots \\ R_{o,\overline{R_o}} \end{cases}$$

Chaque requête offre est constitué de :

-L'itinéraire du chauffeur :

$$Itin_{o,i} = \{Pos_{o,i,1}[h_{o,1}^-, h_{o,1}^+, PU_{o,1}, D_{o,1}], \dots, \\ Pos_{o,i,\overline{Itin_{o,i}}}[h_{o,\overline{Itin_{o,i}}}^-, h_{o,\overline{Itin_{o,i}}}^+, PU_{o,\overline{Itin_{o,i}}}, D_{o,\overline{Itin_{o,i}}}] \}$$

-C_{o,i} : c'est le chauffeur c associé à la requête offre R_{o,i}.

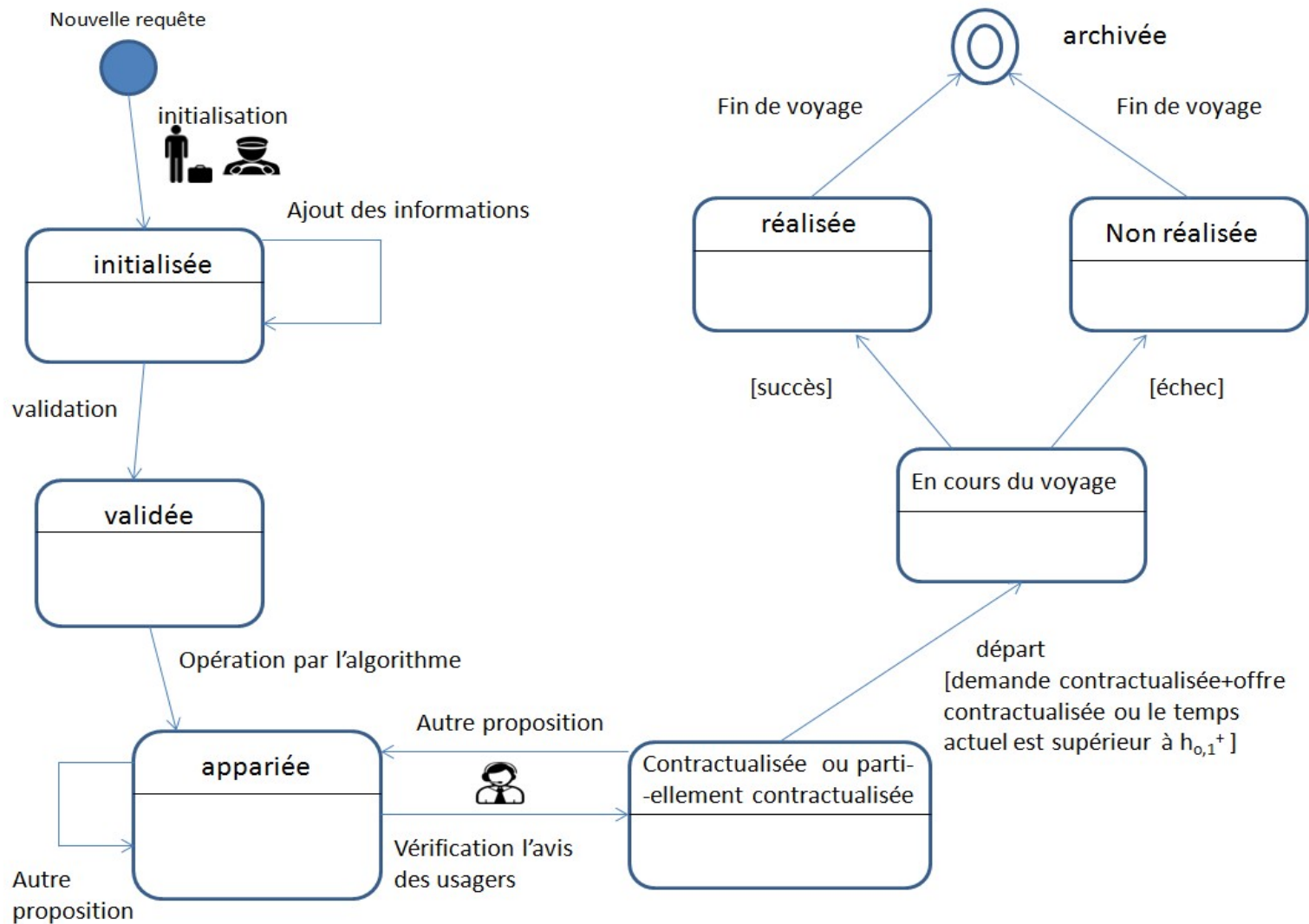
- V_{o,i} : c'est le véhicule v associé à la requête offre R_{o,i}

- PlacesL_{o,i,initial} : est le nombre de places disponible à l'initialisation de R_{o,i}.

-PlacesL_{o,i,actuel} : est le nombre de places disponible à un l'instant actuel du traitement de R_{o,i}.

- State_{o,i} : indique l'état de la requête.

Diagramme d'états transitions des requêtes



Des organisations sont obtenues par une suite d'appariement entre les requêtes offre et les requêtes demande .

Après chaque appariement effectué, on est ramené à un problème de même nature. On définit par **Org** l'organisation contrainte lors d'un nouveau cycle d'appariement.

$$Orgs = \begin{cases} Org_1 \\ \vdots \\ Org_{Cse} = \begin{cases} Cse_1 \\ \vdots \\ Cse_k = Itin_{Cse,k}, Ro_{Cse,k}, ListRd_{Cse,k}, State_{Cse,k}, PlaceO_{Cse,k}, PlaceL_{Cse,k} \\ \vdots \\ Cse_{Org_{Cse}} \end{cases} \\ \vdots \\ Org_{Orgs} \end{cases}$$

- $Itin_{Cse,k}$, c'est l'itinéraire de la course K

$$Itin_{Cse,k} = \{Pos_{Cse,k,1}, \dots, Pos_{Cse,k,p}, \dots, Pos_{Cse,k, \overline{Itin_{Cse,k}}}\}$$

- $Ro_{cse,k}$, c'est la requête offre appariée dans la course K.

- $ListRd_{Cse,k}$: c'est une liste des requêtes demande appariées avec la requête offre.

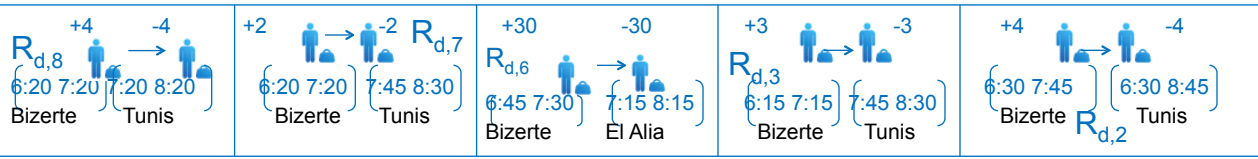
$$ListRd_{Cse,k} = \{Rd_{Cse,k,1}, \dots, Rd_{Cse,k,l}, \dots, Rd_{Cse,k, \overline{R_{Cse,k}}}\}$$

- $State_{Cse,k}$: indique l'état de la course.

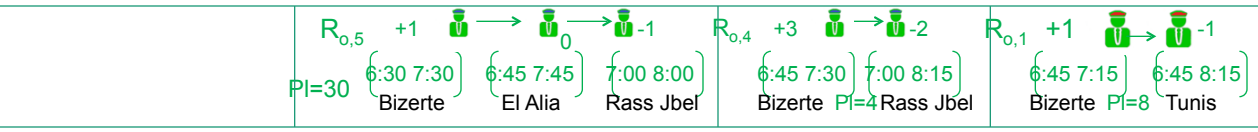
- $PlaceO_{Cse,k}$: indique le nombre de places maximal occupé par les passagers le long de l'itinéraire de Cse_k .

- $PlaceL_{Cse,k}$: c'est le nombre de places libres minimal de l'itinéraire de Cse_k

Illustration d'appariement des requêtes



Requêtes demande



Requêtes offre



Légende

.....Trajet de passager	Chauffeur d'un louage rouge	R_d Requête demande
.....Trajet de chauffeur	Louage à bande bleu	R_o Requête offre
Passager	Louage à bande rouge	PI Places libres
Chauffeur d'un louage bleu	+1 Un chauffeur ou un passager montant (PU)	-1 Un chauffeur ou un passager descendant (D)
L'opérateur		

6:00h

8:30h

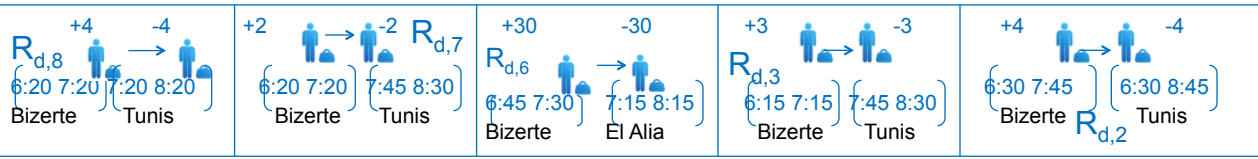
Bizerte

El Alia

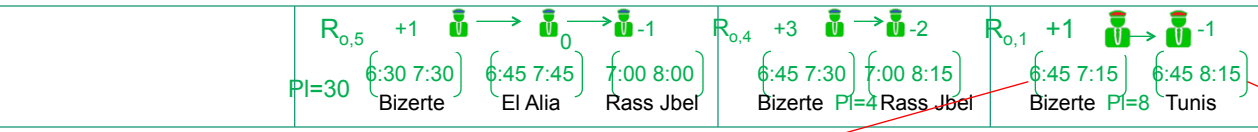
Rass Jbel

Tunis

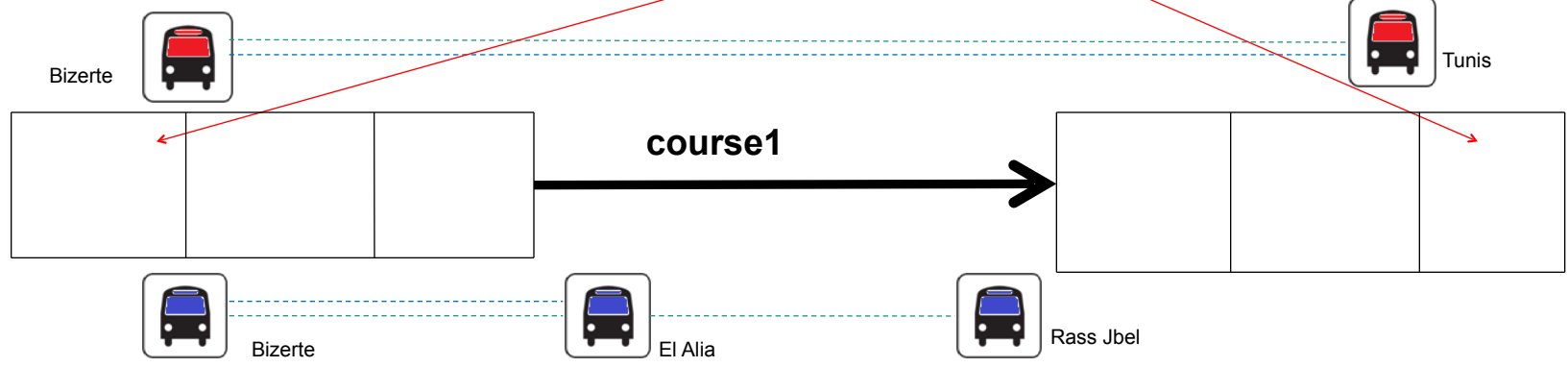
Illustration d'appariement des requêtes



Requêtes demande



Requêtes offre



6:00h

8:30h

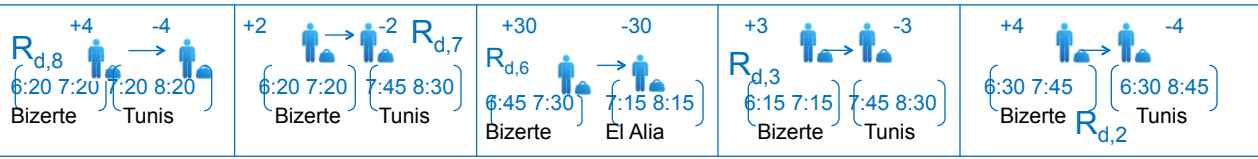
Bizerte

El Alia

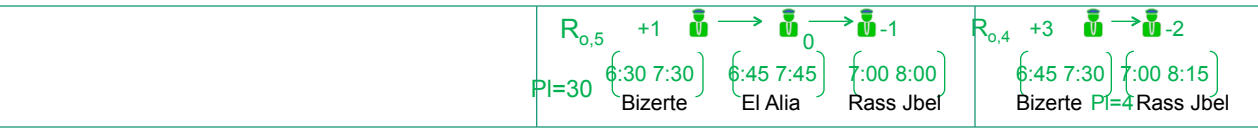
Rass Jbel

Tunis

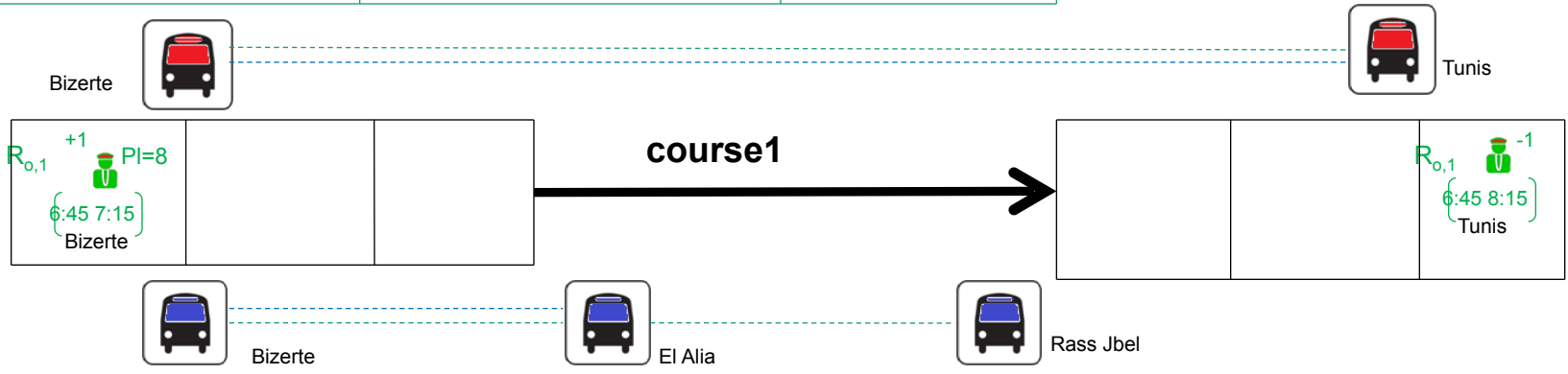
Illustration d'appariement des requêtes



Requêtes demande



Requêtes offre



6:00h

8:30h

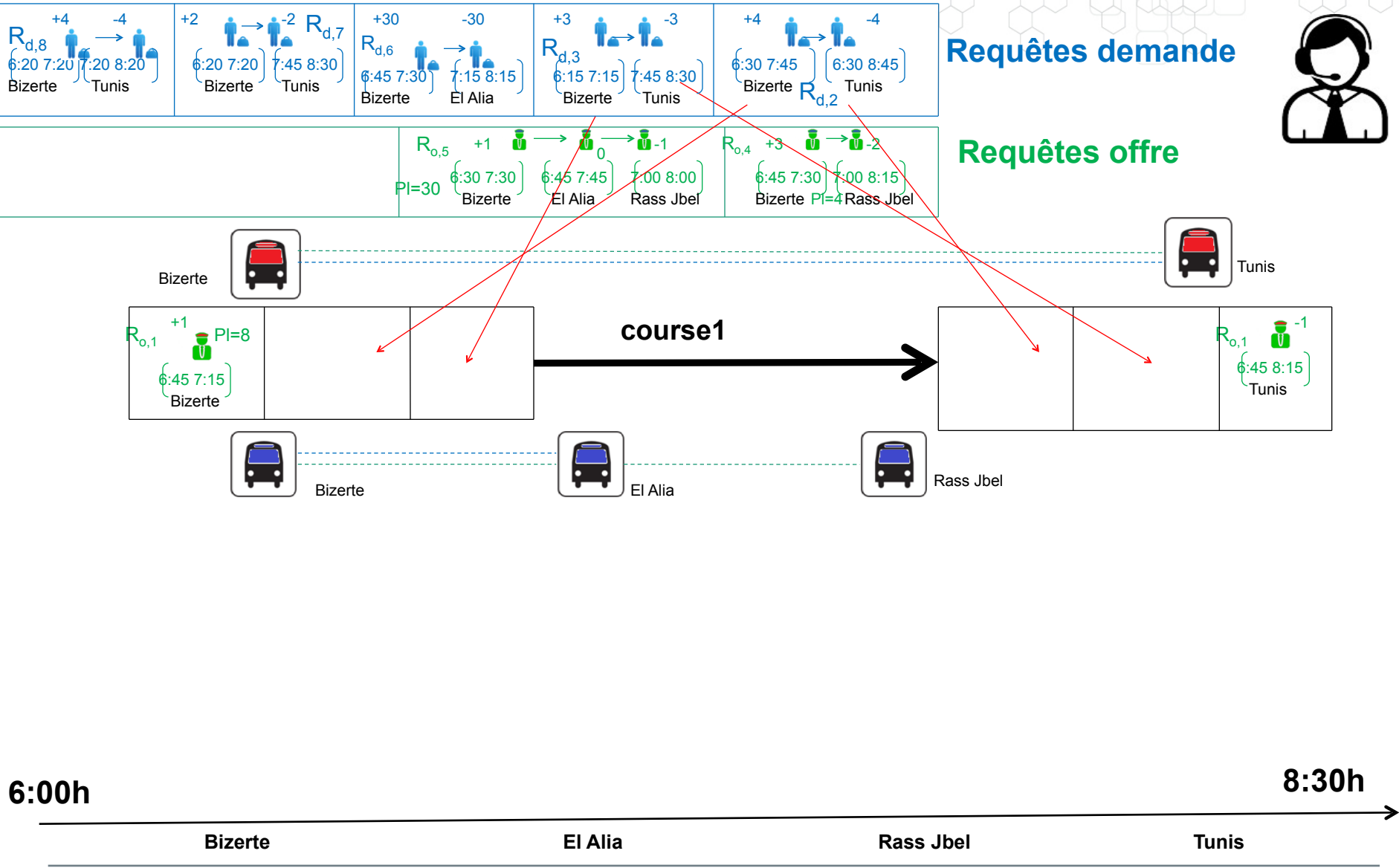
Bizerte

El Alia

Rass Jbel

Tunis

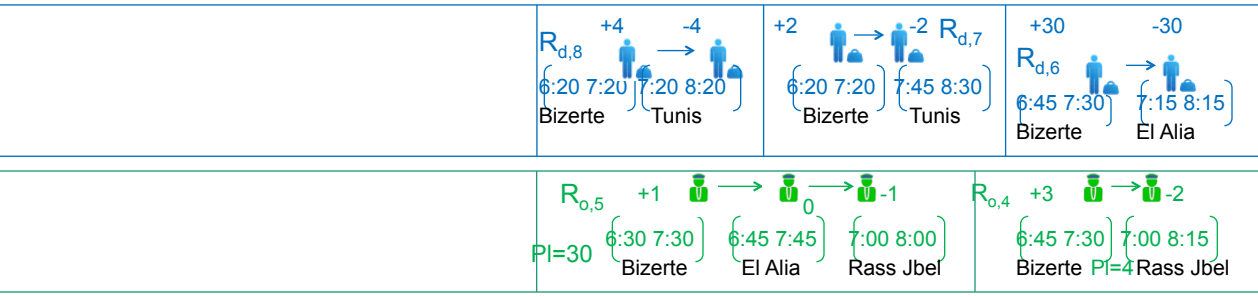
Illustration d'appariement des requêtes



6:00h

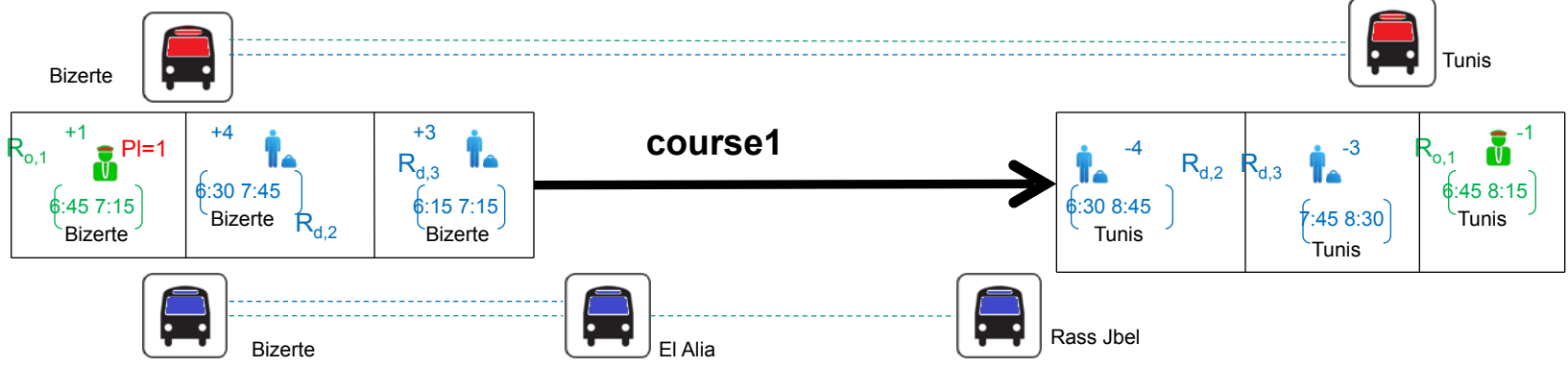
8:30h

Illustration d'appariement des requêtes



Requêtes demande

Requêtes offre



6:00h

8:30h

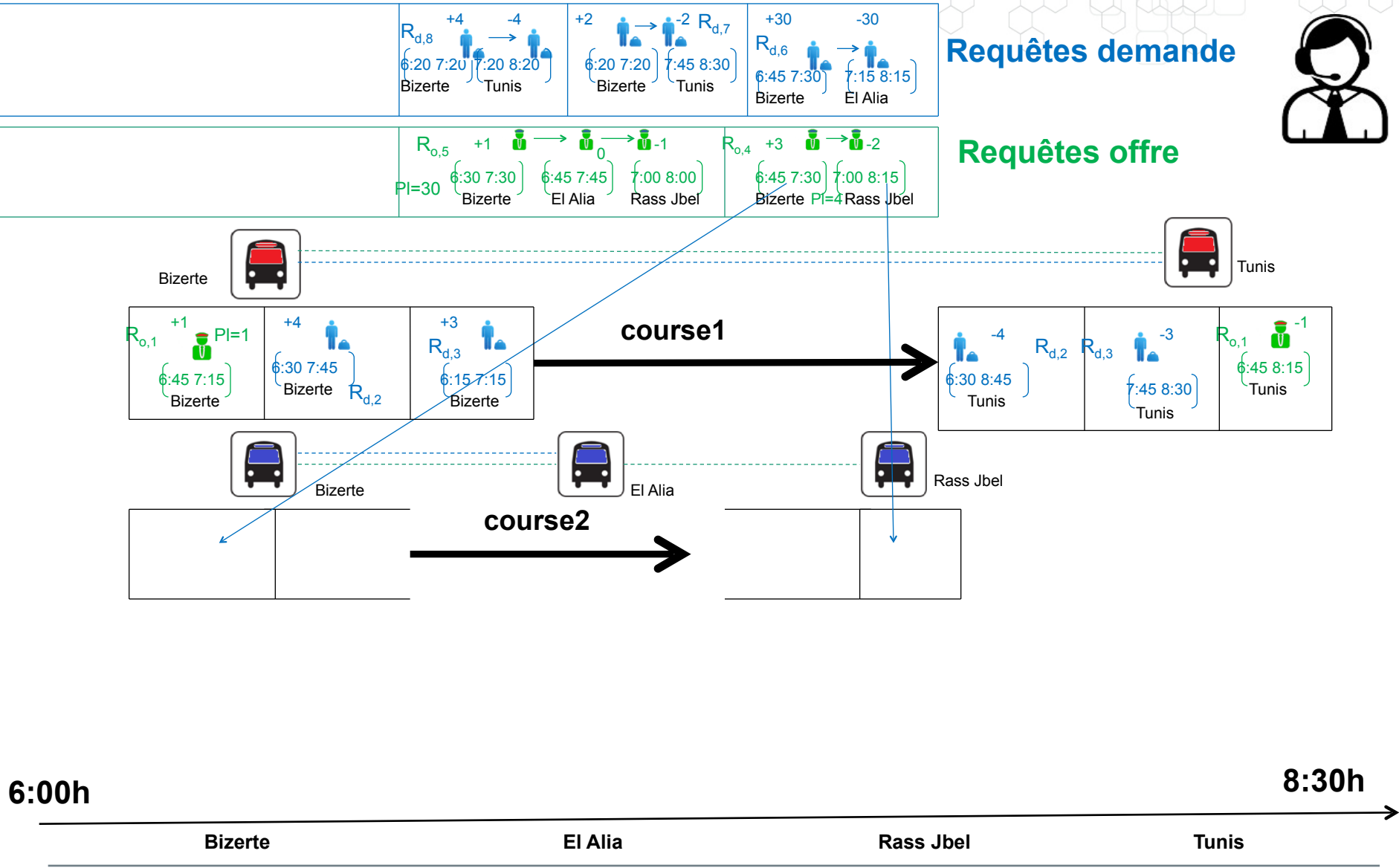
Bizerte

El Alia

Rass Jbel

Tunis

Illustration d'appariement des requêtes



6:00h

8:30h

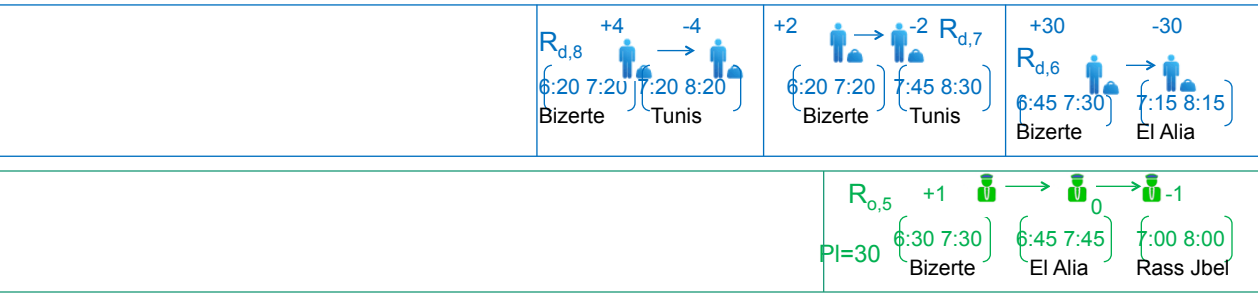
Bizerte

El Alia

Rass Jbel

Tunis

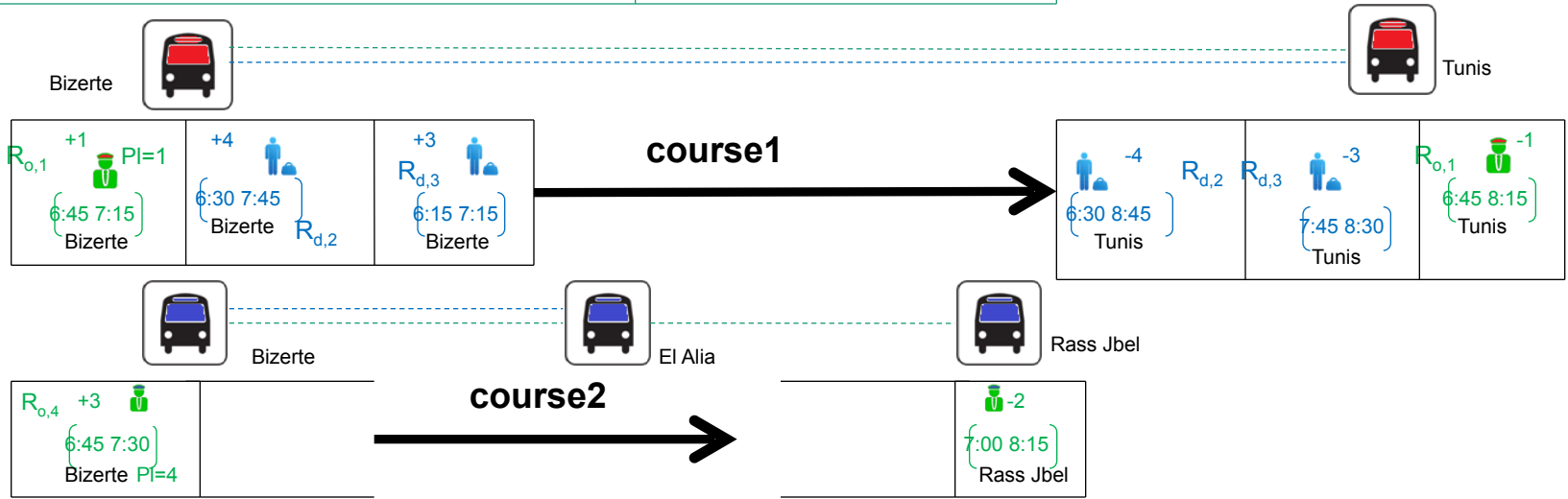
Illustration d'appariement des requêtes



Requêtes demande



Requêtes offre



6:00h

8:30h

Bizerte

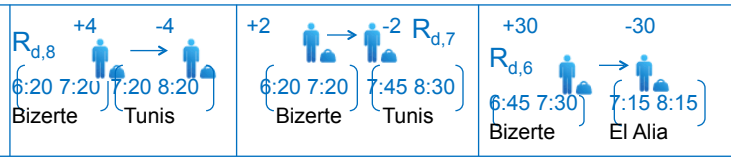
El Alia

Rass Jbel

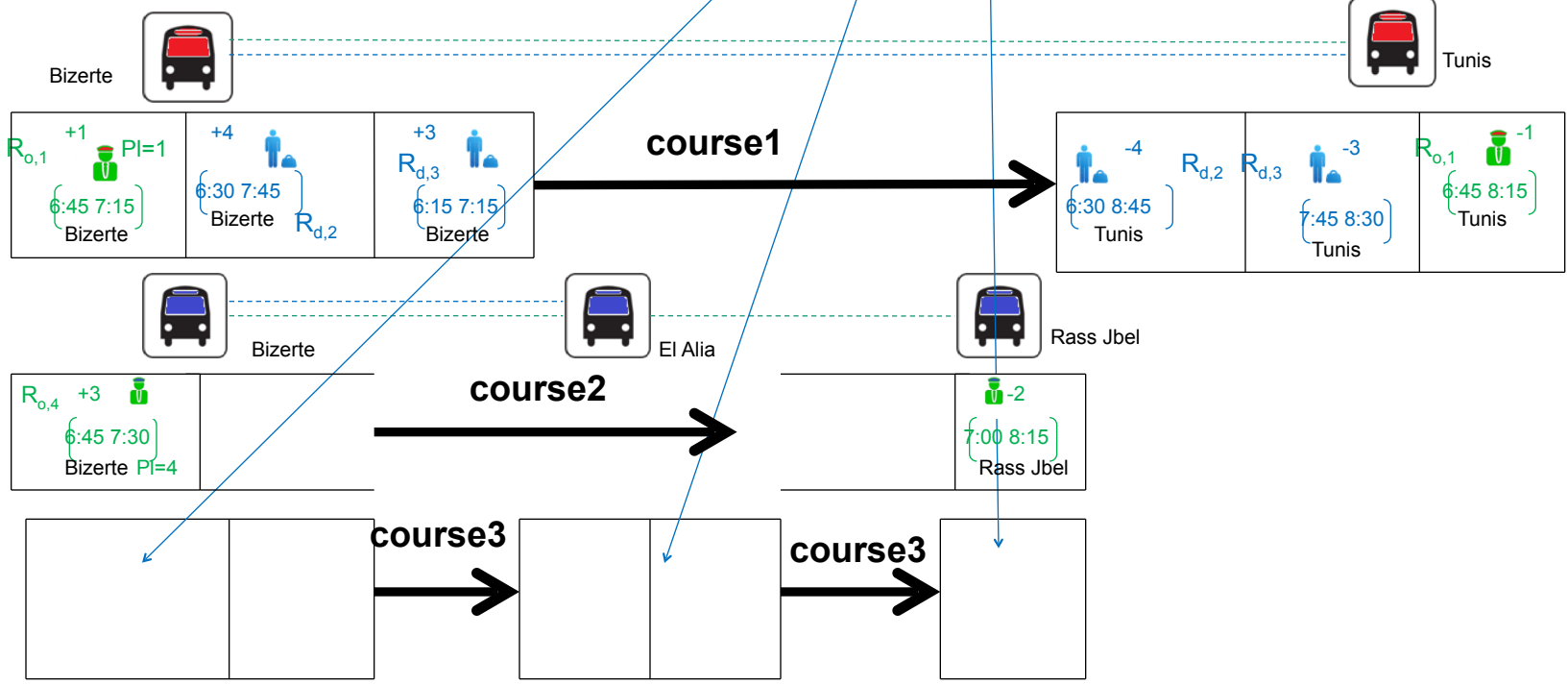
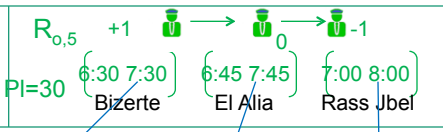
Tunis

Illustration d'appariement des requêtes

Requêtes demande



Requêtes offre



6:00h

8:30h

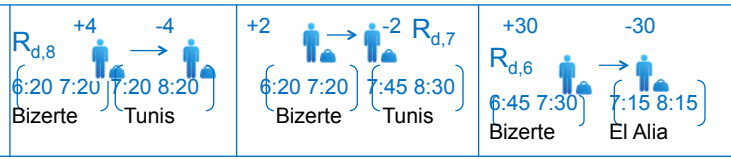
Bizerte

El Alia

Rass Jbel

Tunis

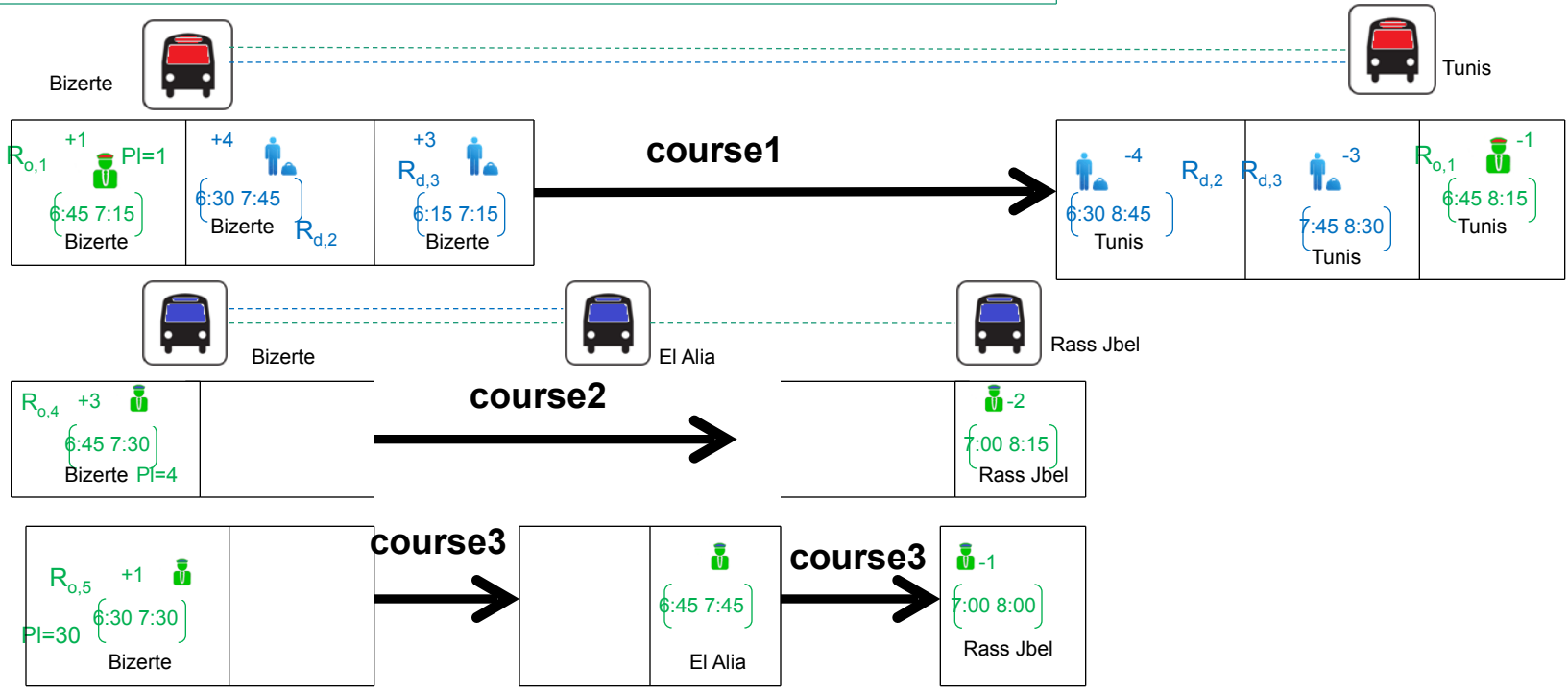
Illustration d'appariement des requêtes



Requêtes demande



Requêtes offre



6:00h

8:30h

Bizerte

El Alia

Rass Jbel

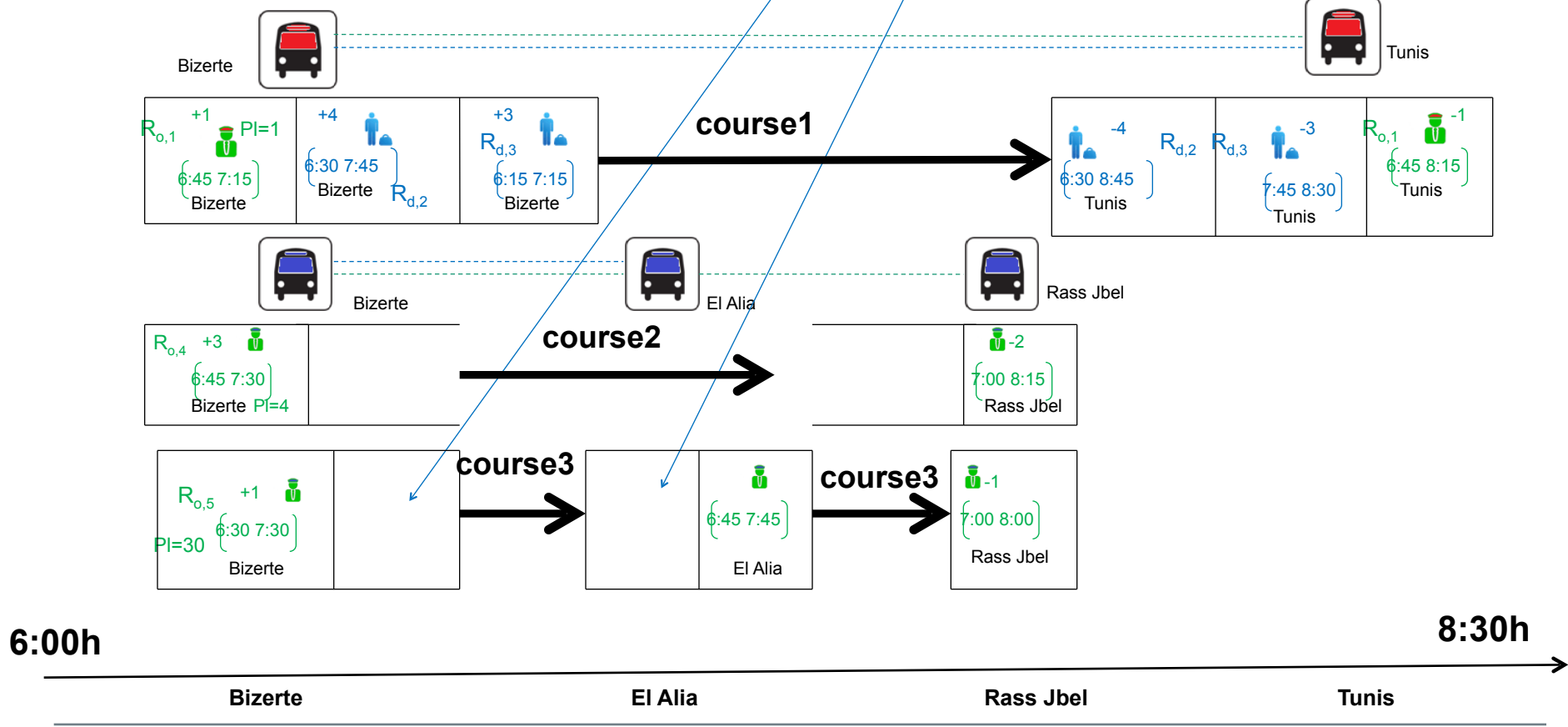
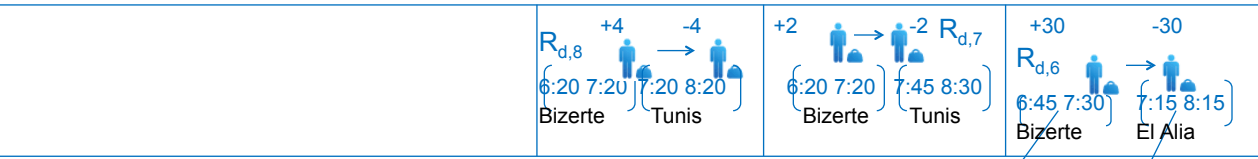
Tunis

Illustration d'appariement des requêtes

Requêtes demande



Requêtes offre



6:00h

8:30h

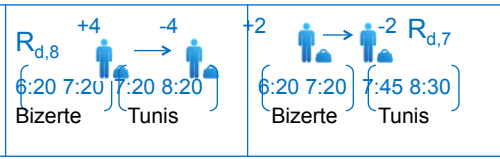
Bizerte

El Alia

Rass Jbel

Tunis

Illustration d'appariement des requêtes

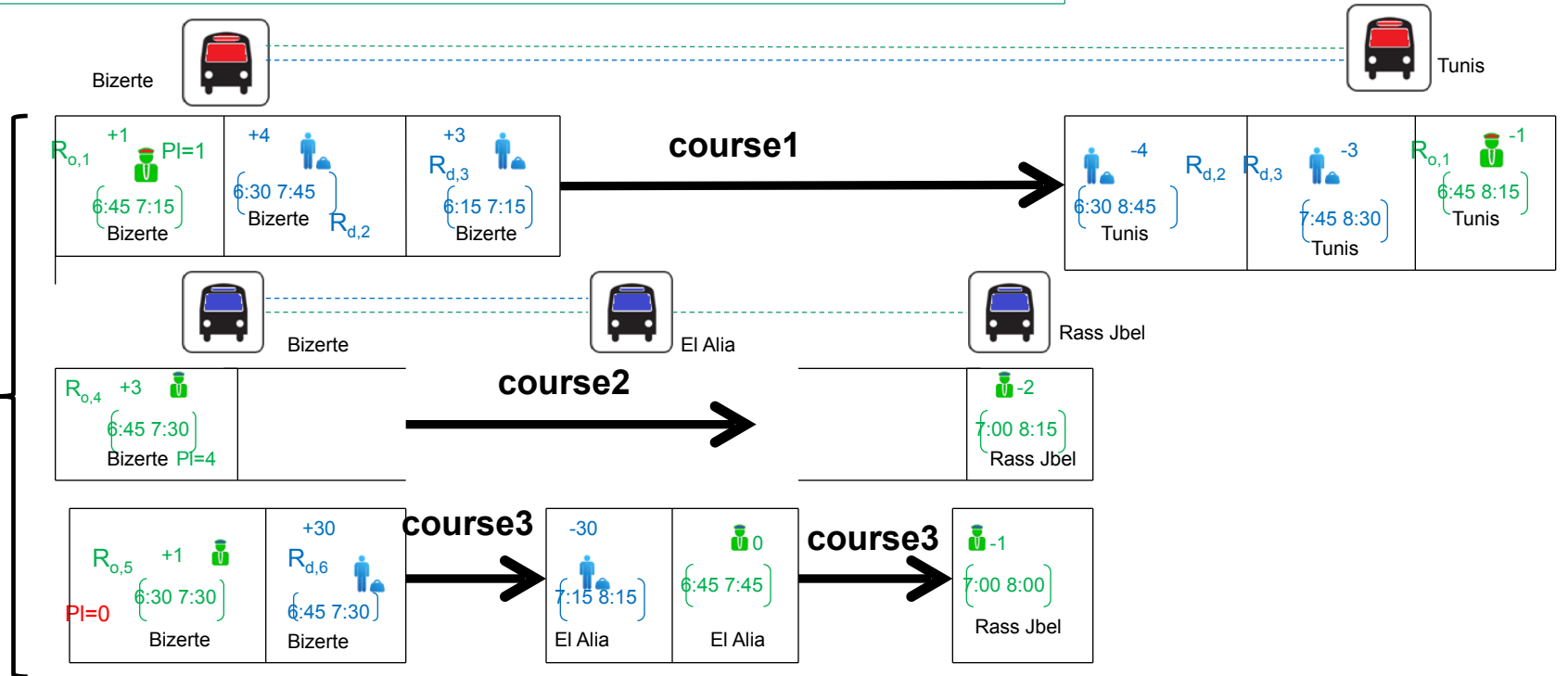


Requêtes demande



Requêtes offre

ORGANISATION



6:00h

8:30h

Bizerte

El Alia

Rass Jbel

Tunis

Les multi contraintes

Le système du louage fait intervenir un ensemble d'acteurs. Ces acteurs sont liés à plusieurs contraintes qu'on va les énumérer dans cette partie.

1. Les contraintes de cohérence et de normalisation des fenêtres de temps.
 - contrainte de normalisation des fenêtres de temps au sein d'une requête offre
 - contrainte de cohérence d'une fenêtre de temps d'une position d'une requête demande avec la même position de la course
2. Les contraintes des lignes de louage
3. Les contraintes des itinéraires
4. Les contraintes de préservation des ordres des positions des itinéraires au sein de R_o , R_d et Org
 - par rapport à Org
 - par rapport à Orgs
5. Les contraintes de préservation des ordres des requêtes dans Org et Orgs.
6. Les contraintes de transition des états.
7. Les contraintes de capacité
8. Les contraintes de prix

Les multi contraintes

Les contraintes des lignes de louage

On définit l'ensemble \mathbf{Ligne}_c la liste des lignes associées à un chauffeur c .

$$\mathbf{Lignes}_c = \{ \mathbf{Ligne}_{c,l} \} \quad \forall c \in \mathbf{Ligne}_{c,l} = Pos_{C,c,l,1}[h_1^-, h_1^+], \dots, Pos_{C,c,l,p}[h_p^-, h_p^+], \dots, \\ Pos_{C,c,l, \overline{\mathbf{Ligne}_{c,l}}}[h_{\overline{\mathbf{Ligne}_{c,l}}}^-, h_{\overline{\mathbf{Ligne}_{c,l}}}^+]$$

On définit par \mathbf{Lignes} la liste des lignes du système.

$$\mathbf{Lignes} = \{ \mathbf{Ligne}_{nl \in [1, \overline{\mathbf{Lignes}}]} \} = \bigcup_{c=1}^{\overline{C}} \left(\bigcup_{l=1}^{\overline{\mathbf{Ligne}_c}} \mathbf{Ligne}_{c,l} \right)$$

13

Contrainte des lignes à opérer par un chauffeur :

Le nombre maximal de lignes à opérer par un chauffeur est $\mathbf{NbMaxLpC}$. Ce nombre ne peut pas dépasser 3 et il est fixé par un arrêté.

$$\forall c \in [1, \overline{C}], \quad \mathbf{NbMaxLpC} = 3$$

Les multi contraintes

Les contraintes liant les itinéraires des requêtes offre et des requêtes demande qui sont initialisées avec les lignes

Pour une requête offre

On définit par $\prod_B(A)$ un symbole qui signifie la projection de A sur B. Pour une requête offre initialisée, la projection de l'itineraire associé sur les positions et les fenêtres de temps coïncide avec les positions d'une ligne existante dans la liste **Lignes**.

$$\forall R_{o,i \in [1, \overline{R_o}]} \text{ et } R_{o,i}.state = \text{"initialisée"}, \exists nl \in [1, \overline{Ligne}]$$

$$\prod_{pos_{o,p}[h_p^-, h_p^+]} (Itin_{o,i}) = Ligne_{nl}$$

Pour une requête demande

✚ Pour une requête demande initialisée, la projection de l'itinéraire associé sur les positions et les fenêtres de temps est égale ou incluse aux positions d'une ligne existante dans la liste **Lignes**.

$$\forall R_{d,j \in [1, \overline{R_d}]} \text{ et } R_{d,j}.state = \text{"initialisée"}, \exists nl \in [1, \overline{Ligne}]$$

$$\prod_{pos_{d,p}[h_p^-, h_p^+]} (Itin_{d,j}) \subseteq Ligne_{nl}$$

Les multi objectifs

Le louage est un système multi-objectifs. Chacun des intervenants a sa vision pour les objectifs. Dans cette partie on va énumérer l'ensemble de ces objectifs

1. maximisation du taux d'occupation
2. maximiser les requêtes offre et les requêtes demande contractualisées
3. respecter la liste FIFO des requêtes
4. maximiser l'écart type des temps d'attente des usagers
5. minimiser les temps totaux des courses
6. maximiser le gain des chauffeurs

Les multi objectifs

Maximiser le nombre des requêtes demande et des requêtes offre contractualisées

✚ maximiser le nombre des requêtes offre et des requêtes demande contractualisées :
ce qui revient à maximiser $\overline{R_{o,con}}$ et $\overline{R_{d,con}}$ et par suite minimiser le nombre des requêtes offre et des requêtes demande non contractualisées et $\overline{R_{o,!con}}$ et $\overline{R_{d,!con}}$

✚ Respecter la liste FIFO des requêtes.

Minimiser les temps totaux des courses dans une organisation

✚ Le temps total TT (Total Time) de voyage de toutes les courses présentes dans une organisation doit être minimum.

$$\forall Org_{Cse \in [1, \overline{Orgs}]} \in Orgs,$$

$$\forall Itin_{Cse, k \in [1, \overline{OrgCse}]} = Pos_{Cse, k, 1}, \dots, Pos_{Cse, k, p}, \dots, Pos_{Cse, k, \overline{ItinCse, k}}$$

$$TT = \sum_{Cse=Cse_1}^{\overline{Cse_{OrgCse}}} \sum_{p=1}^{\overline{ItinCse, k}} M_{o-d}(Pos_{Cse, k, p}, Pos_{Cse, k, p+1}) \cdot t$$

Mono-objectif associé

✚ Dans la formalisation du problème de louage, on a adopté une approche multi-objectifs.

✚ Notre algorithme **GILA (Greedy Incremental Louage's Algorithm)** est un algorithme mono-objectif (objectif réalisé : ordre totale).

✚ Cet objectif principal est la maximisation du nombre des requêtes contractualisées en respectant leur ordre d'arrivée.

✚ **GILA** génère une seule organisation à chaque itération.

Description de l'algorithme

Algorithm 1 Greedy louage's incremental algorithm($Offers_{init}$ O_{init} , $Demands_{init}$ D_{init} , $Organization$ Org)

```

1: //validation des listes des requêtes offre et des requêtes demande initialisées
2:  $O_v \leftarrow ValidateInitializedRequests(O_i)$  //contrainte 3.2
3: //validateInitializedRequests est une fonction qui assure la validation d'une liste de requêtes
   offre ou des requêtes demande
4:  $D_v \leftarrow ValidateInitializedRequests(D_i)$  //contrainte 3.2
5: //initialisation de la liste des organizations
6: if  $Org! = Null$  then
7:    $Orgs = Concat(Orgs, Org)$ 
8: else
9:    $Orgs = new Organization()$ 
10: end if
11: //ajout des offres validées dans la première organisation à la suite des courses de l'organisa-
   tion existante
12:  $Orgs.Org_1 \leftarrow Concat(Org, O_v)$ 
13:  $PcourseOrgs \leftarrow Orgs$ 
14:  $PitinOrgs \leftarrow PcourseOrgs$ 
15:  $PpositinOrgs \leftarrow PitinOrgs$ 
16:  $PD_v \leftarrow D_v$ 
17: while  $PcourseOrgs! = Null$  &&  $!(PcourseOrgs.état = "contractualisée")$  do
18:   while  $PD_v! = Null$  &&  $!(PD_v.état.equals("contractualisée"))$  do
19:      $PPosD_v \leftarrow D_v$ 
20:     while  $PPosD_v! = Null$  &&  $PpositinOrgs! = Null$  do
21:       if  $IsInserted(PPosD_v, PpositinOrgs, PpositinOrgs.next())$  then
22:         //fonction qui vérifie l'insertion avec respect des contraintes 3.12, 3.13, 3.21, 3.22//
23:          $ValidateLocalTimeWindow(PPosD_v, PpositinOrgs.next())$ 
24:          $PropagationOfTimeWindow(PPosD_v, PpositinOrgs, PpositinOrgs.next())$ 
25:          $Update(PPosD_v, PpositinOrgs, PpositinOrgs.next())$ 
26:         //fonction qui vérifie met à jour l'itinéraire avec respect des contraintes 3.12 et
           3.13//
27:          $PpositinOrgs \leftarrow PPosD_v$ 
28:          $PPosD_v \leftarrow PPosD_v.next()$ 
29:       else if  $PPosD_v = Null$  then
30:         //une demande a été injecté

```

```

31:    $D_{con} \leftarrow D_{con} + PD_v$ 
32:    $D_v \leftarrow D_v - PD_v$ 
33:    $PpositinOrgs.PlaceO = PpositinOrgs.PlaceO + PD_v.NP$ 
34:    $PpositinOrgs.PlaceL = PpositinOrgs.PlaceL - PD_v.NP$ 
35:    $PD_v \leftarrow PD_v.next()$ 
36:    $PPosD_v \leftarrow PD_v$ 
37: else
38:   //insertion impossible et avancement du pointeur sur la prochaine requête demande
39:    $PD_v \leftarrow PD_v.next()$ 
40:    $PPosD_v \leftarrow PD_v$ 
41: end if
42: end while
43:    $PD_v \leftarrow PD_v.next()$ 
44: end while
45:  $PCourseOrgs \leftarrow PCourseOrgs.next()$ 
46:  $PitinOrgs \leftarrow PCourseOrgs$ 
47:  $PpositinOrgs \leftarrow PitinOrgs$ 
48: end while
49: return  $Orgs$ 

```

Algorithm 2 ValidateLocalTimeWindow(POS $PPosD_v, POS$ $PpositinOrgs.next()$)

```

1: if  $PPosD_v.h^+ + M_{o-d}(PPosD_v, PpositinOrgs.next()).t \geq PpositinOrgs.next().h^+$  then
2:    $PPosD_v.h^+ \leftarrow PpositinOrgs.next().h^+ - M_{o-d}(PPosD_v, PpositinOrgs.next()).t$ 
3: end if
4: if  $PPosD_v.h^- + M_{o-d}(PPosD_v, PpositinOrgs.next()).t \geq PpositinOrgs.next().h^-$  then
5:    $PpositinOrgs.next().h^- \leftarrow PPosD_v.h^- + M_{o-d}(PPosD_v, PpositinOrgs.next()).t$ 
6: end if

```

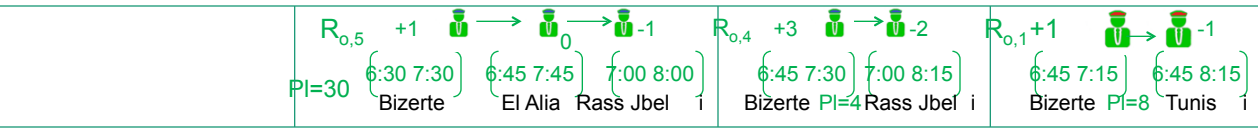
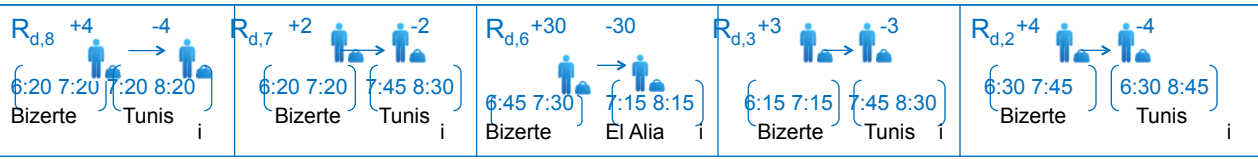
Algorithm 3 PropagationOfTimeWindow(POS $PPosD_v, POS$ $PpositinOrgs, POS$ $PpositinOrgs.next()$)

```

1: if  $PPosD_v.h^- + M_{o-d}(PPosD_v, PpositinOrgs.next()).t \geq PpositinOrgs.next().h^-$  then
2:   /*Propagation à droite*/
    $PpositinOrgs.next().h^- \leftarrow PPosD_v.h^- +$ 
    $M_{o-d}(PPosD_v, PpositinOrgs.next()).t$ 
    $PPosD_v.h^+ \leftarrow PpositinOrgs.next().h^+ -$ 
    $M_{o-d}(PPosD_v, PpositinOrgs.next()).t$ 
3: else if  $PPosD_v.h^+ - M_{o-d}(PpositinOrgs, PPosD_v).t \leq PpositinOrgs.h^+$  then
4:   /*Propagation à gauche*/
    $PpositinOrgs.h^+ \leftarrow PPosD_v.h^+ -$ 
    $M_{o-d}(PpositinOrgs, PPosD_v).t$ 
    $PPosD_v.h^- \leftarrow PpositinOrgs.h^- +$ 
    $M_{o-d}(PpositinOrgs, PPosD_v).t$ 
5: end if

```

Illustration d'appariement des requêtes en appliquant GILA



Légende

i initialisée	-----Trajet de passager	Chauffeur d'un louage rouge	R_d Requête demande
v validée	-----Trajet de chauffeur	Louage à bande bleu	R_o Requête offre
c contractualisée	Passager	Louage à bande rouge	PI Places libres
pc partiellement contractualisée	Chauffeur d'un louage bleu	+1 Un chauffeur ou un passager montant (PU)	-1 Un chauffeur ou un passager descendant (D)
	L'opérateur		

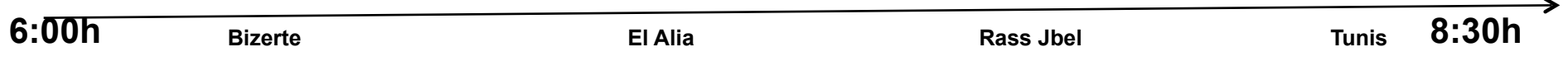
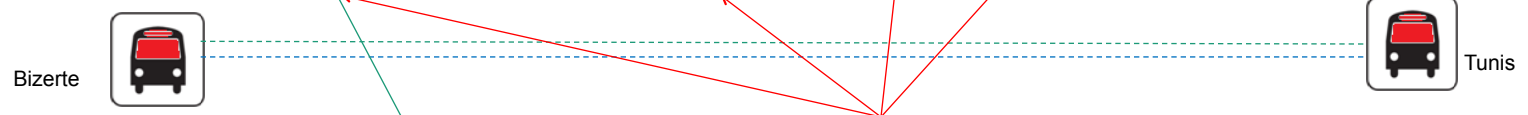
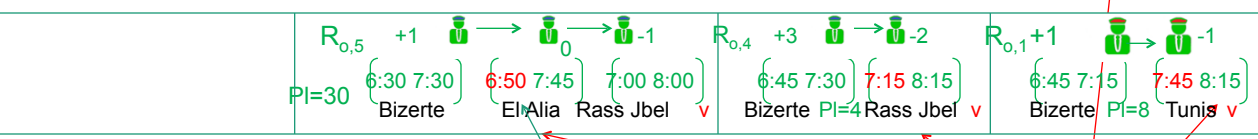
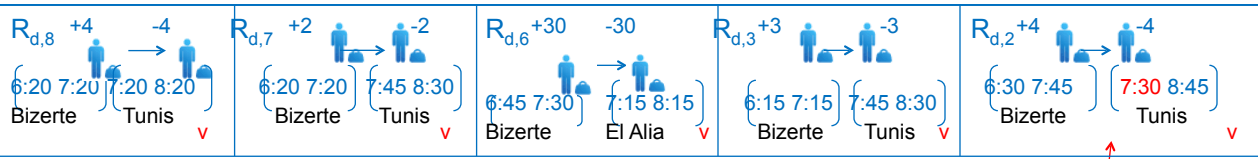


Illustration d'appariement des requêtes en appliquant GILA



Normalisation des fenêtres de temps (validation des R)
ValidateInitializedRequests()



$$Pos_{R_{o,5,1}} \cdot h_v^- = \max(Pos_{R_{o,5,1}} \cdot h_i^-, Pos_{R_{o,5,1}} \cdot h_i^- + d(Pos_{R_{o,5,0}}, Pos_{R_{o,5,1}}), Pos_{R_{o,5,2}} \cdot h_i^- - d(Pos_{R_{o,5,1}}, Pos_{R_{o,5,2}}))$$

Normalisation des fenêtres de temps dans le cas d'un itinéraire avec vias

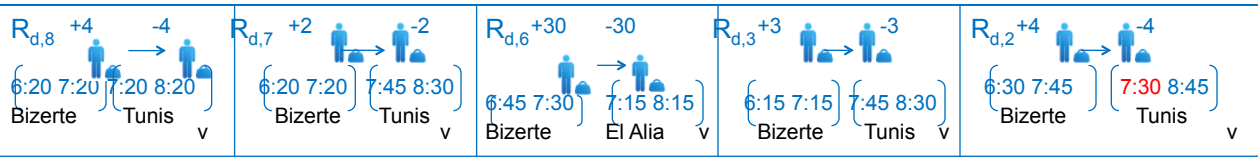


Requêtes demande

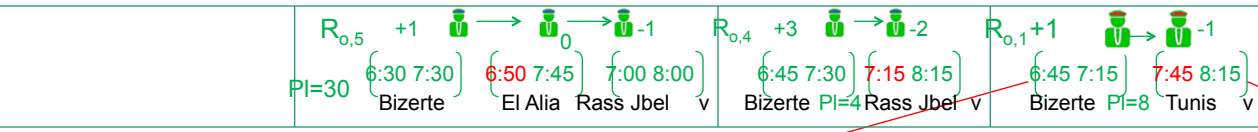
Requêtes offre



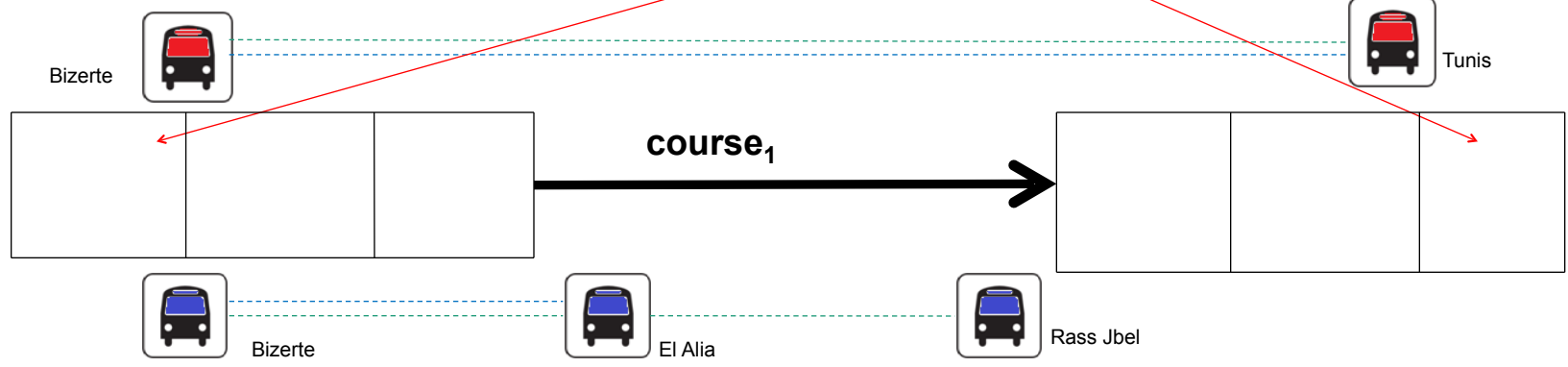
Illustration d'appariement des requêtes en appliquant GILA



Requêtes demande



Requêtes offre



6:00h

8:30h

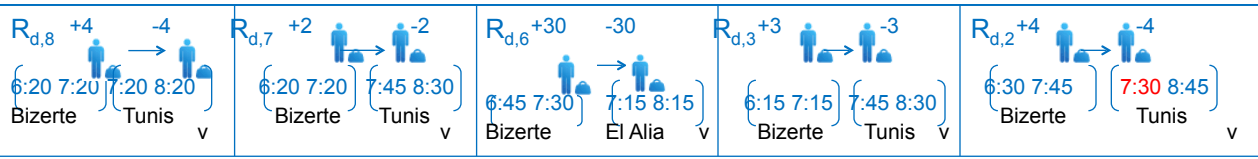
Bizerte

El Alia

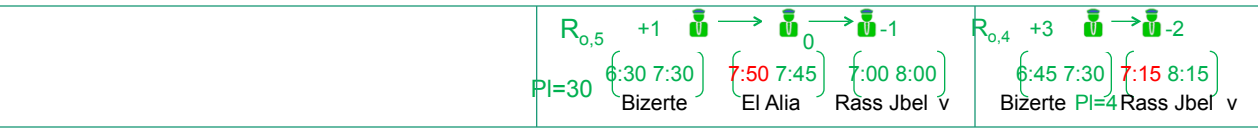
Rass Jbel

Tunis

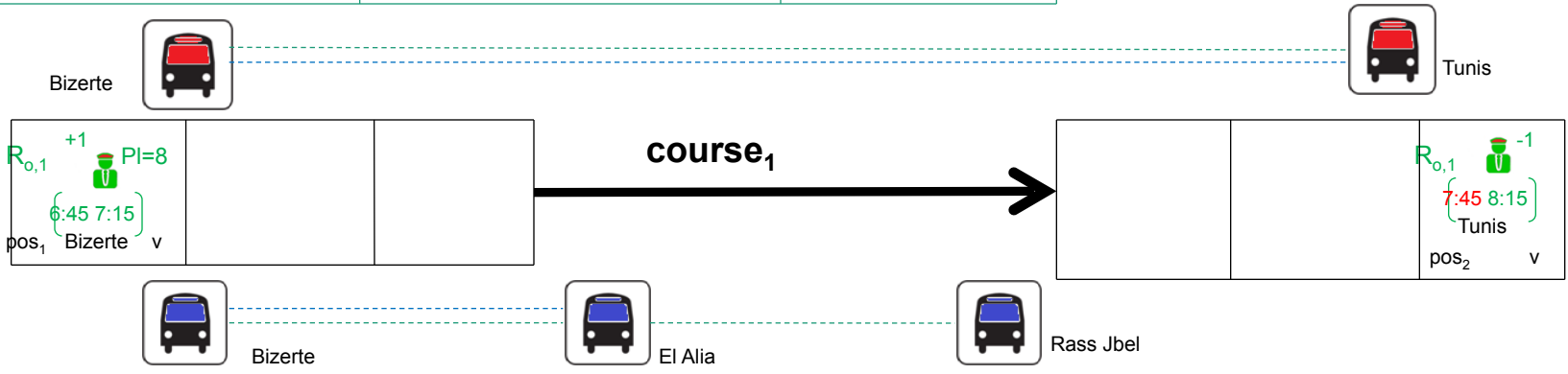
Illustration d'appariement des requêtes en appliquant GILA



Requêtes demande



Requêtes offre



6:00h

8:30h

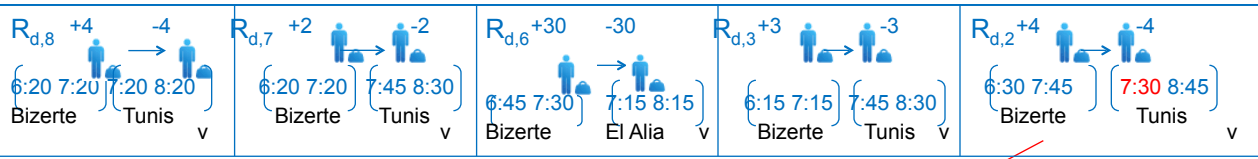
Bizerte

El Alia

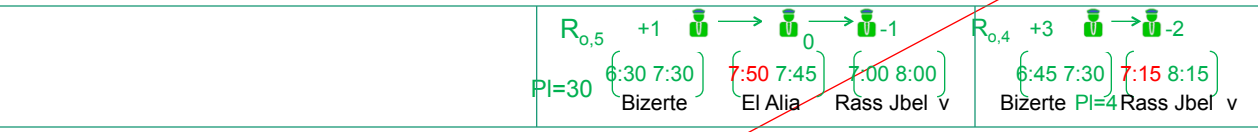
Rass Jbel

Tunis

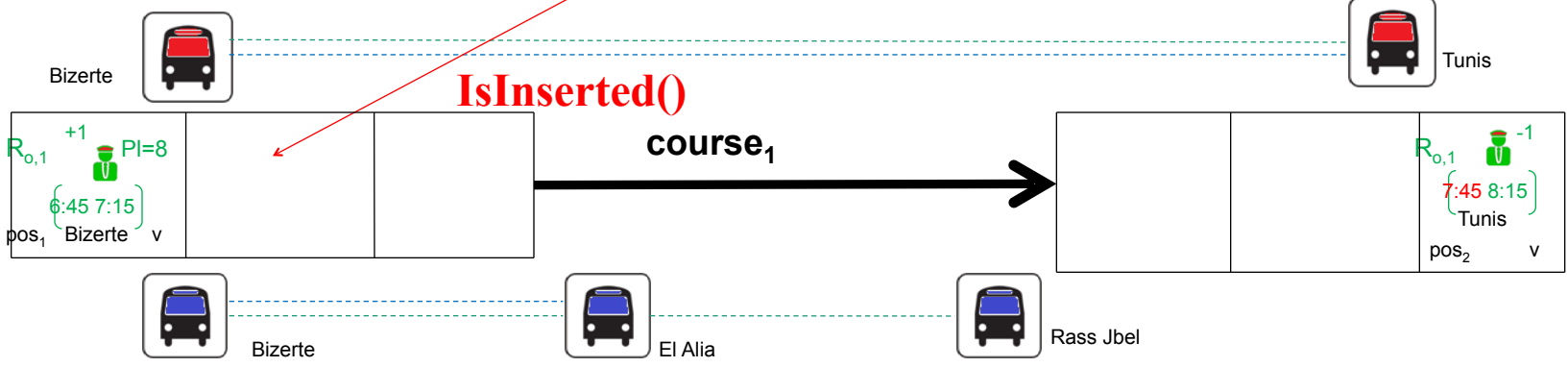
Illustration d'appariement des requêtes en appliquant GILA



Requêtes demande



Requêtes offre



6:00h

8:30h

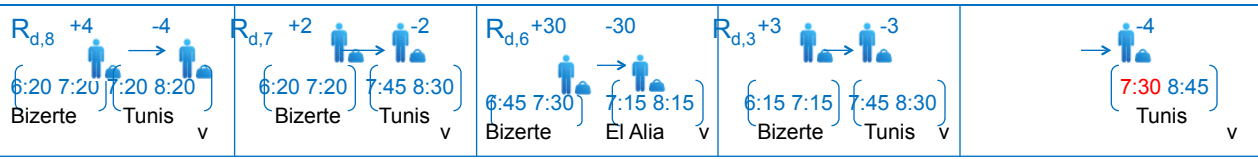
Bizerte

El Alia

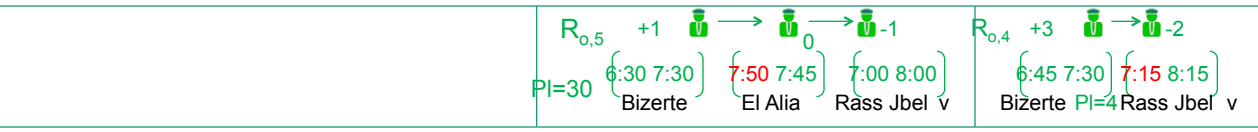
Rass Jbel

Tunis

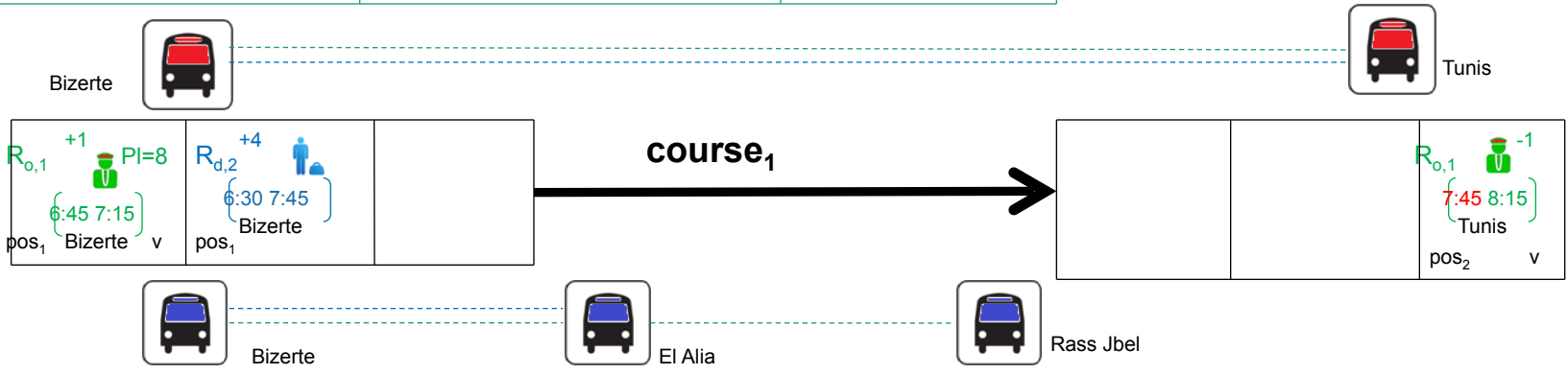
Illustration d'appariement des requêtes en appliquant GILA



Requêtes demande



Requêtes offre



6:00h

8:30h

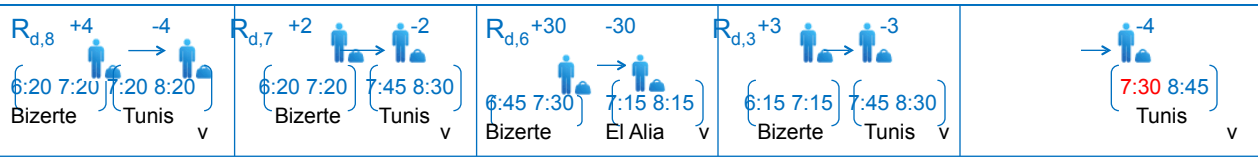
Bizerte

El Alia

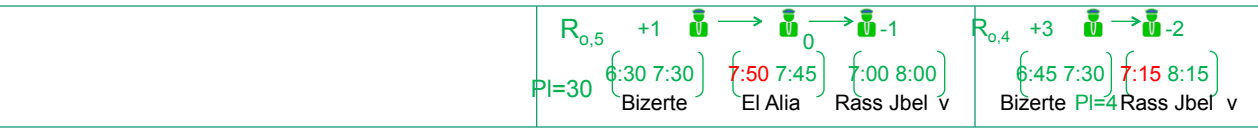
Rass Jbel

Tunis

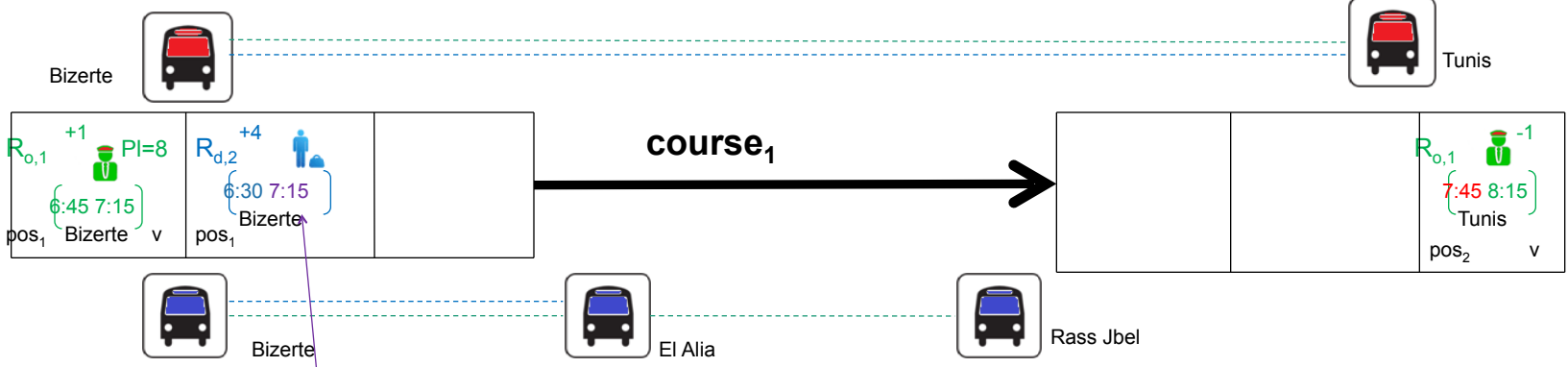
Illustration d'appariement des requêtes en appliquant GILA



Requêtes demande



Requêtes offre



ValidateLocalTimeWindow()

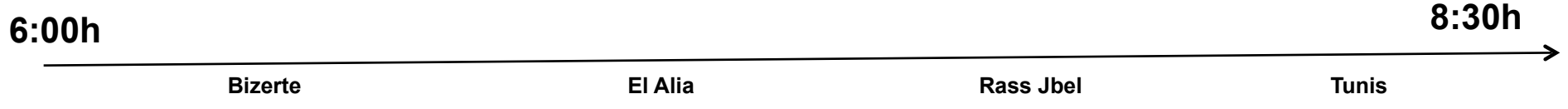
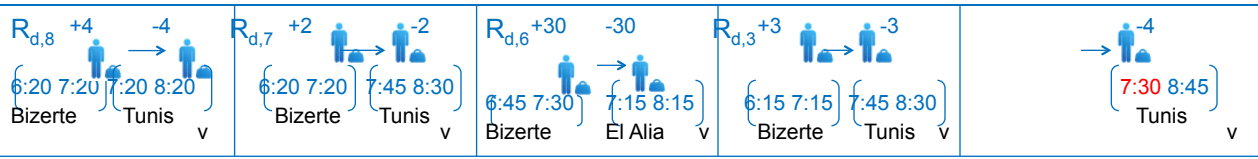
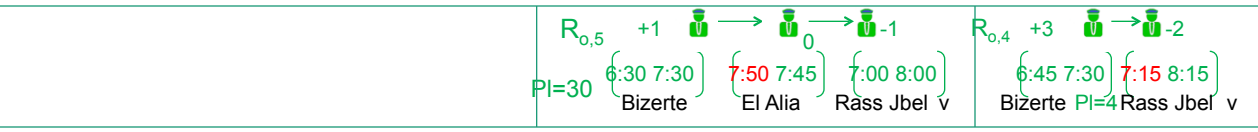


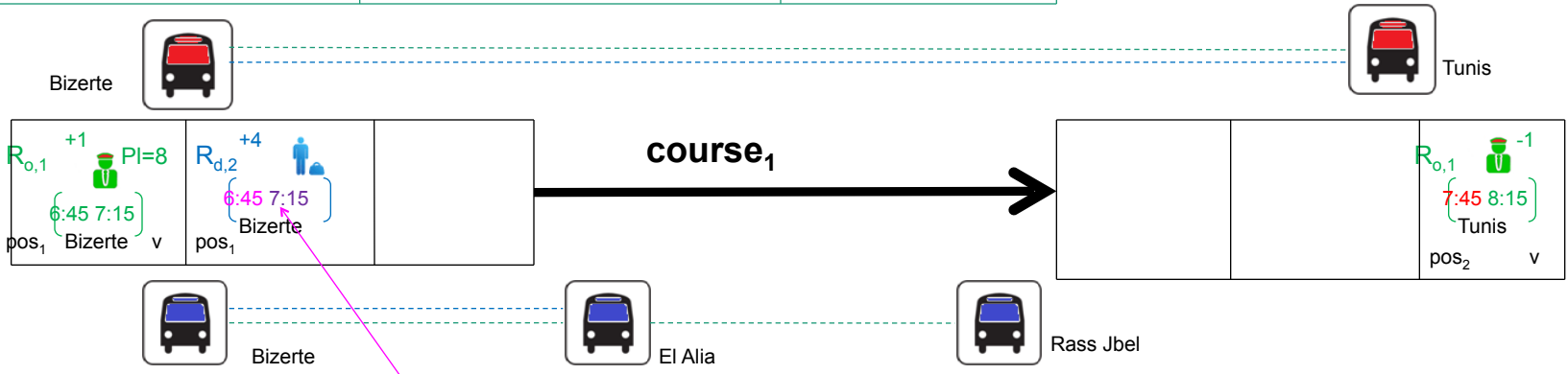
Illustration d'appariement des requêtes en appliquant GILA



Requêtes demande



Requêtes offre



PropagationOfTimeWindow()

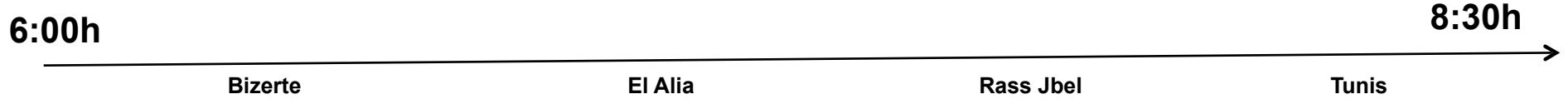
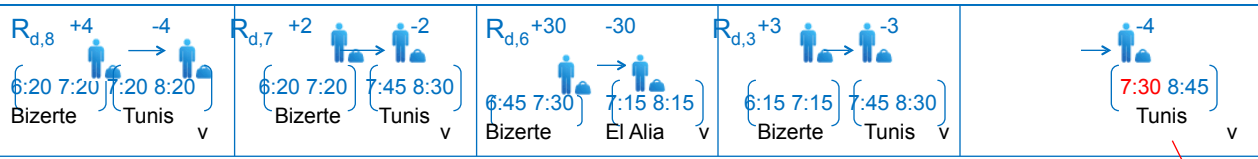
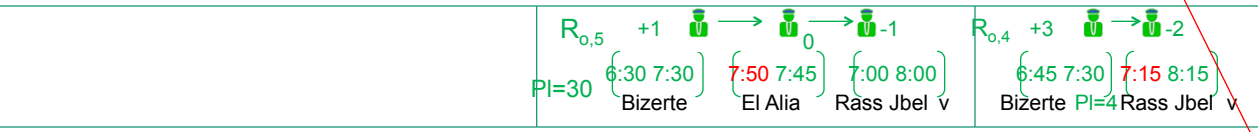


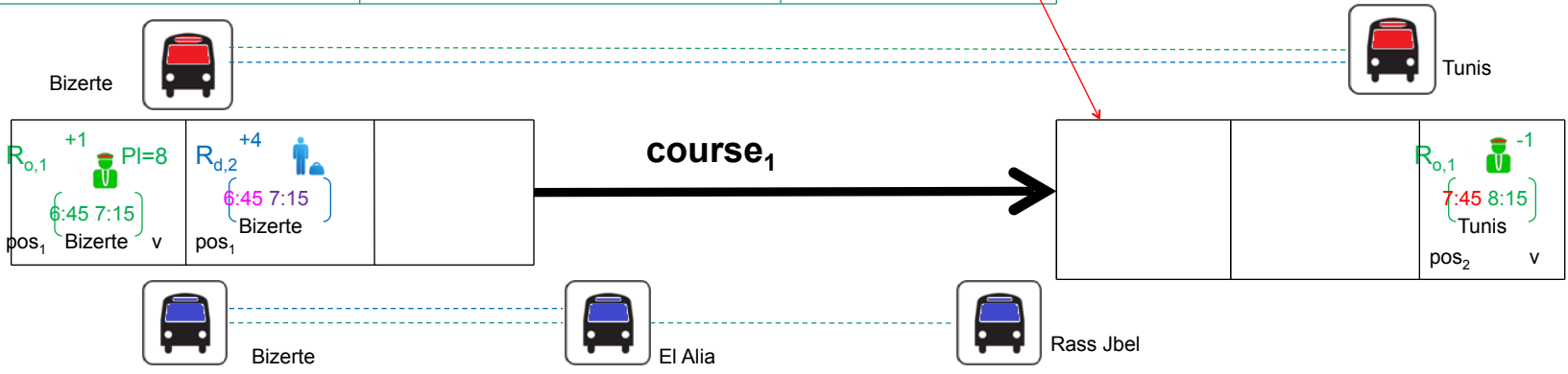
Illustration d'appariement des requêtes en appliquant GILA



Requêtes demande



Requêtes offre



6:00h

8:30h

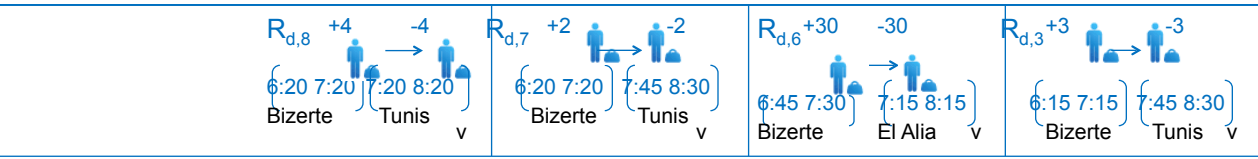
Bizerte

El Alia

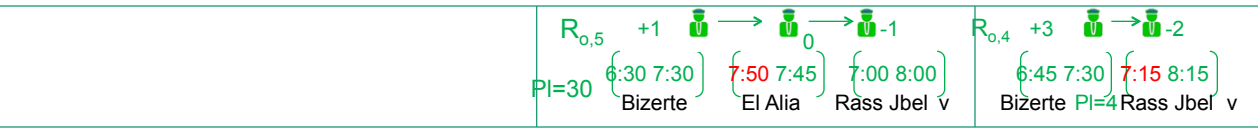
Rass Jbel

Tunis

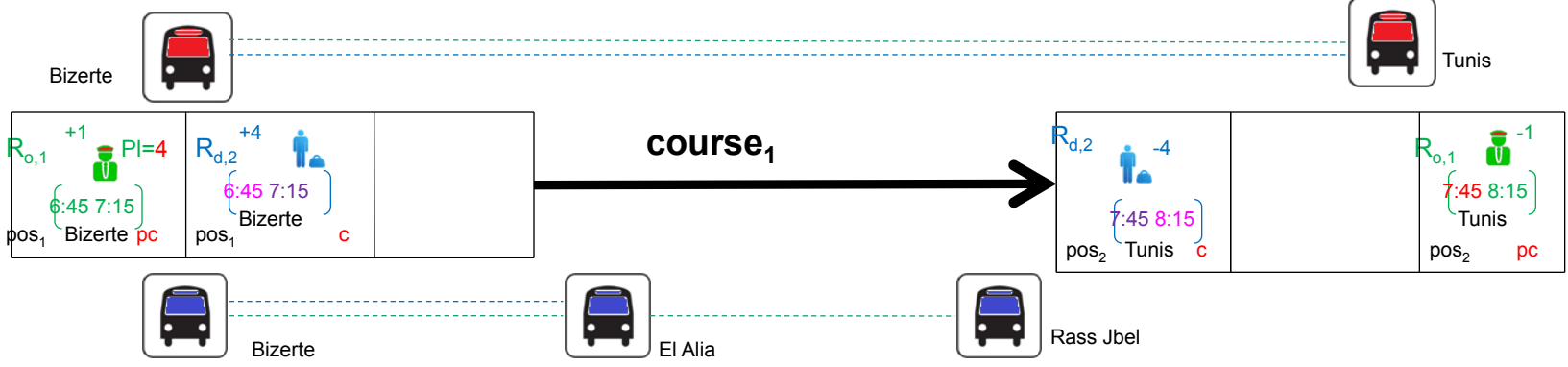
Illustration d'appariement des requêtes en appliquant GILA



Requêtes demande



Requêtes offre



6:00h

8:30h

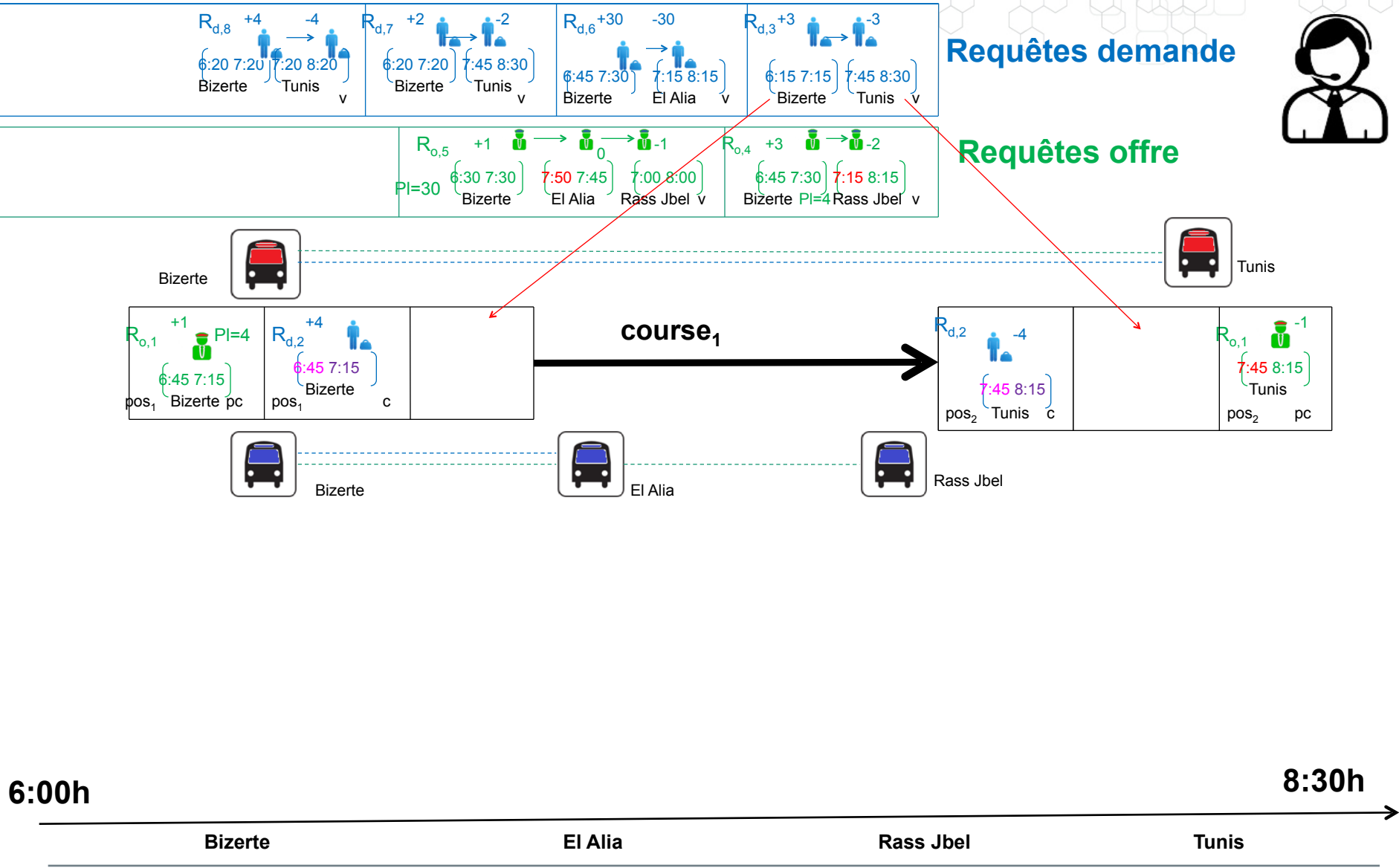
Bizerte

El Alia

Rass Jbel

Tunis

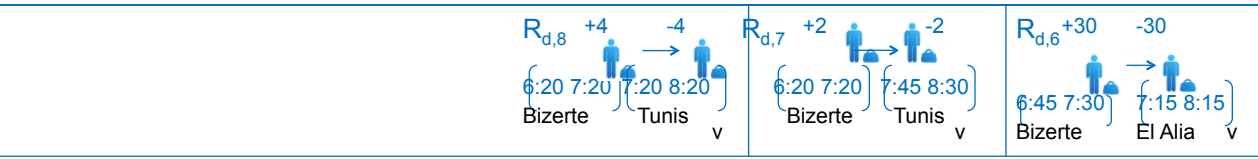
Illustration d'appariement des requêtes en appliquant GILA



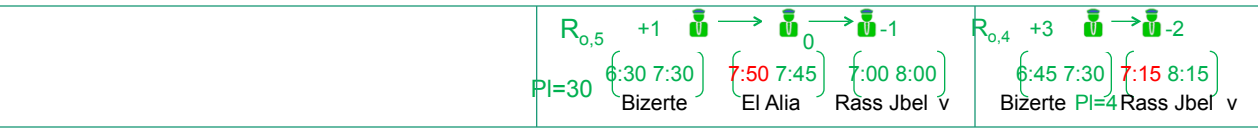
6:00h

8:30h

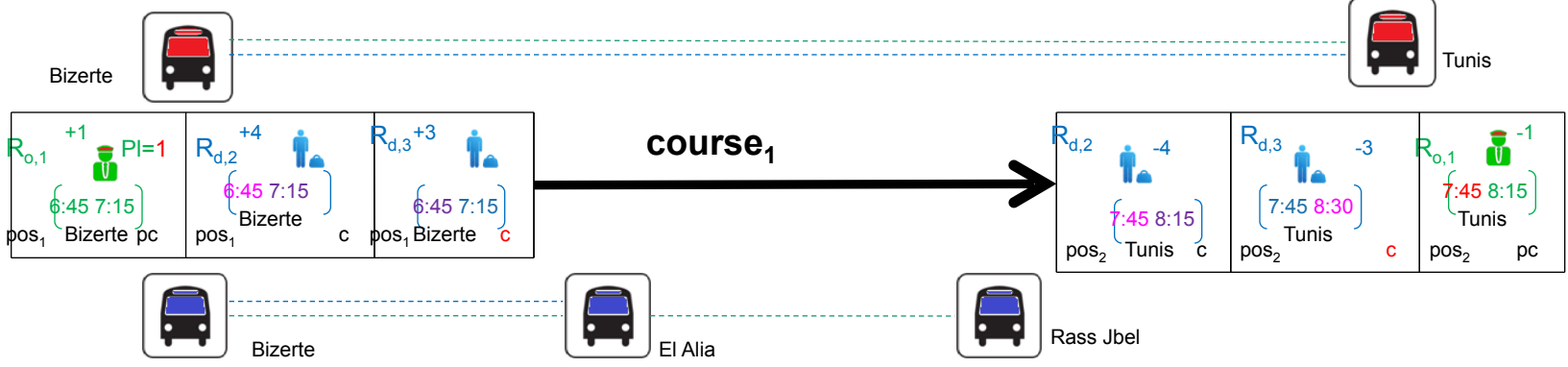
Illustration d'appariement des requêtes en appliquant GILA



Requêtes demande



Requêtes offre



6:00h

8:30h

Bizerte

El Alia

Rass Jbel

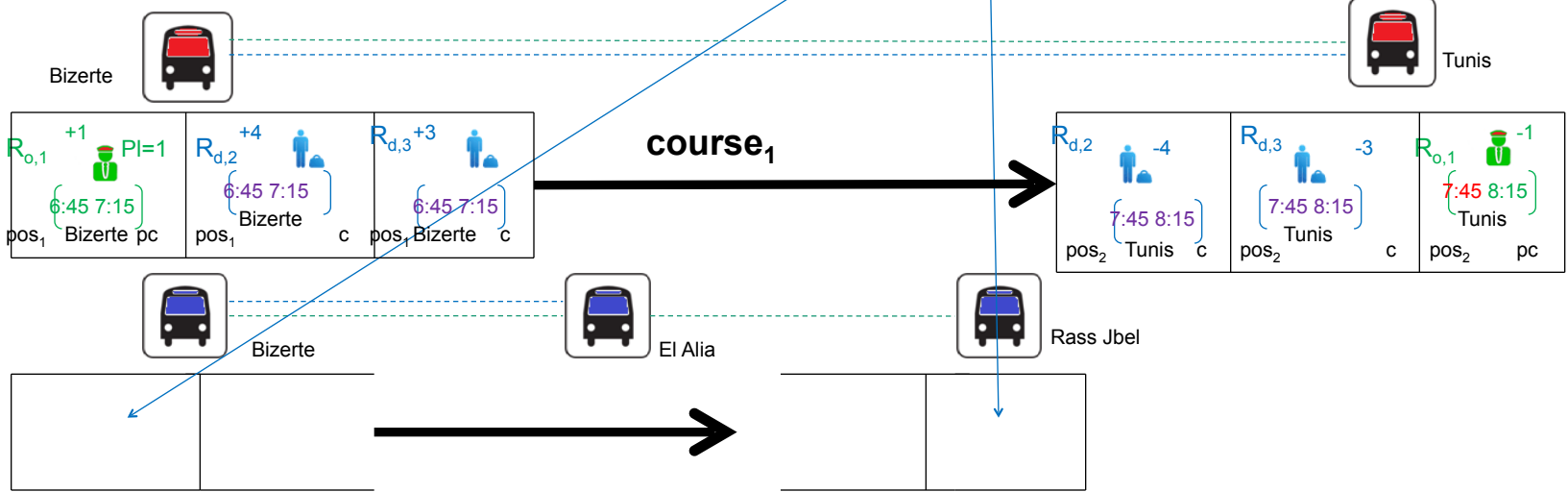
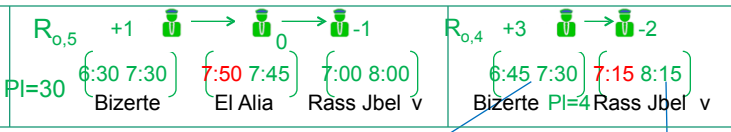
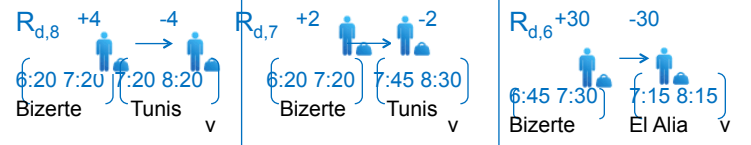
Tunis

Illustration d'appariement des requêtes en appliquant GILA

Requêtes demande



Requêtes offre



6:00h

8:30h

Bizerte

El Alia

Rass Jbel

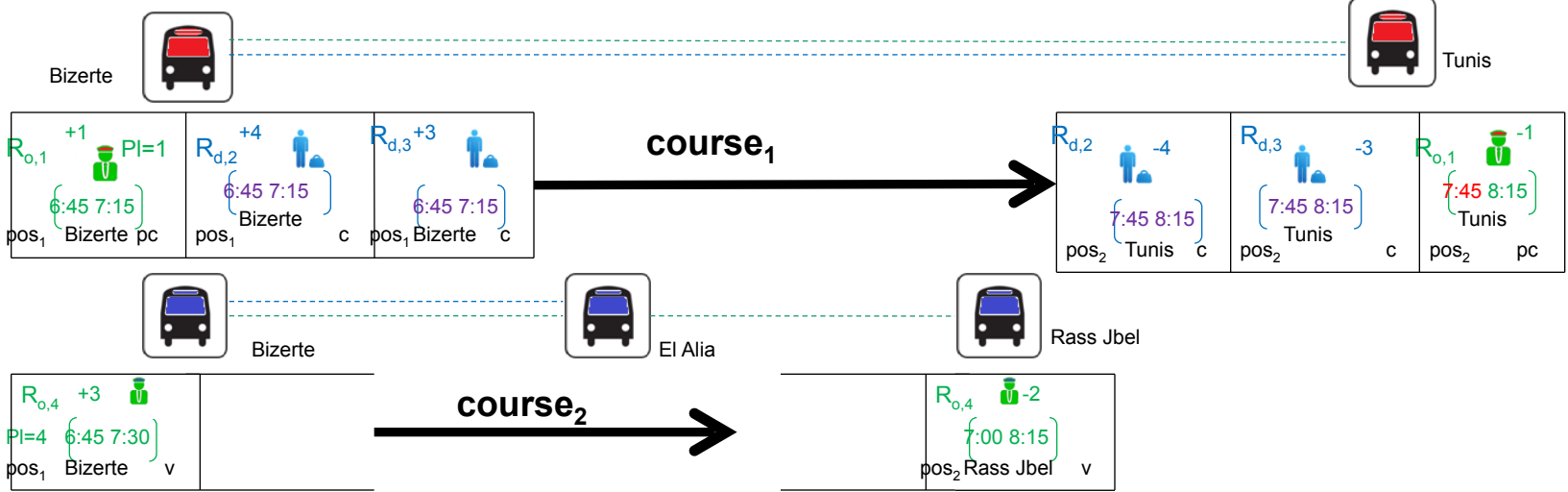
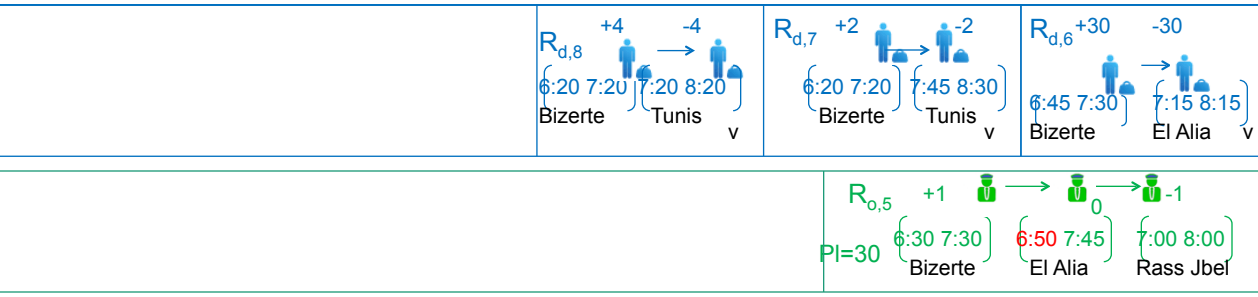
Tunis

Illustration d'appariement des requêtes en appliquant GILA

Requêtes demande



Requêtes offre



6:00h

8:30h

Bizerte

El Alia

Rass Jbel

Tunis

Illustration d'appariement des requêtes en appliquant GILA

Requêtes demande



Requêtes offre

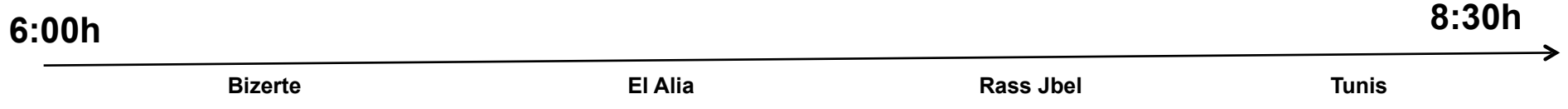
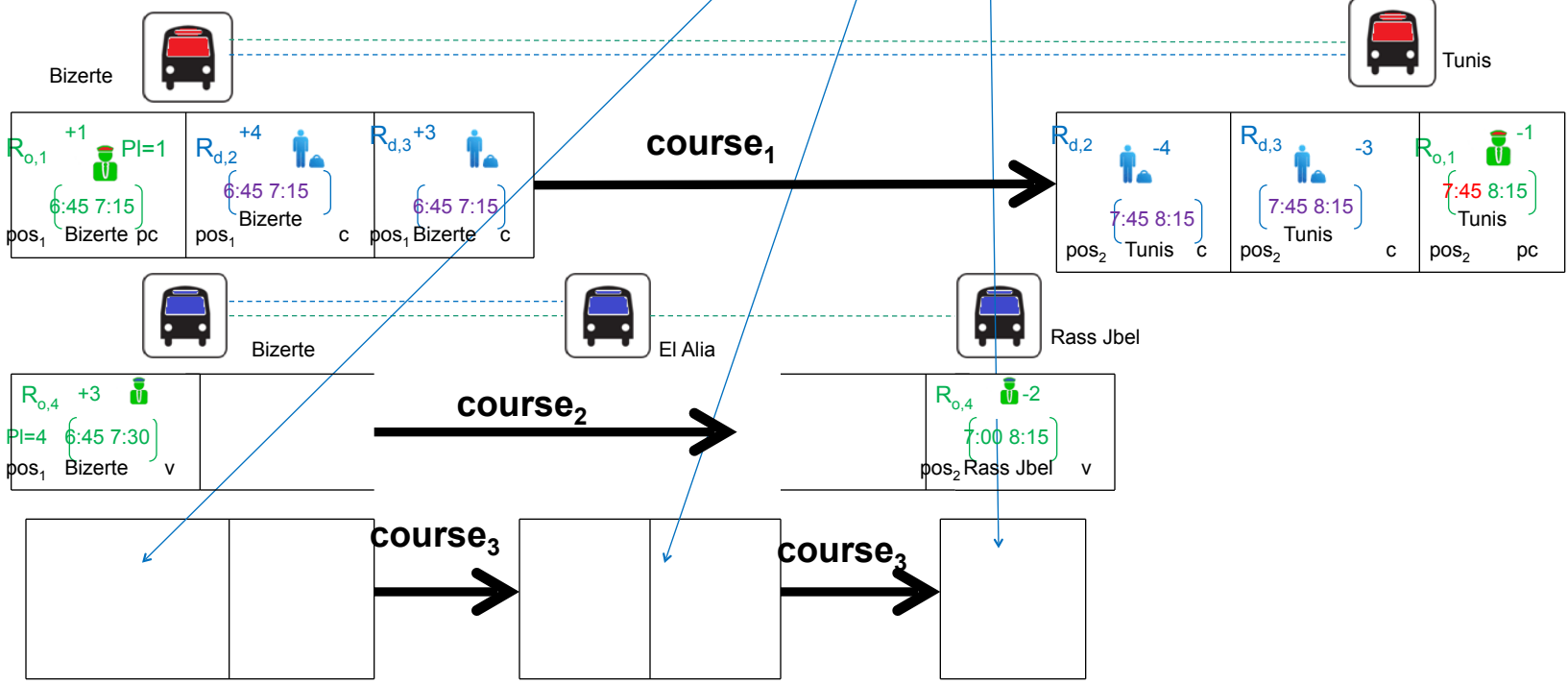
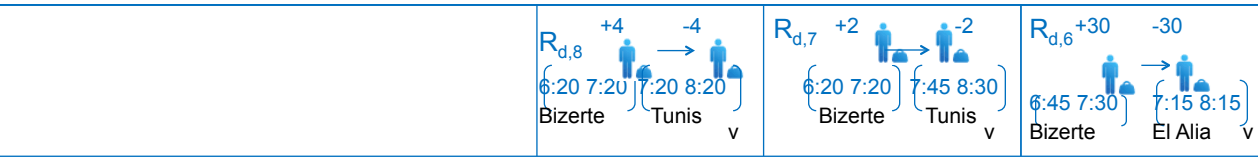


Illustration d'appariement des requêtes en appliquant GILA

Requêtes demande



Requêtes offre

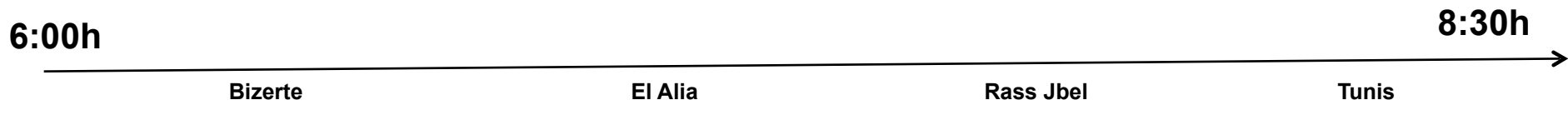
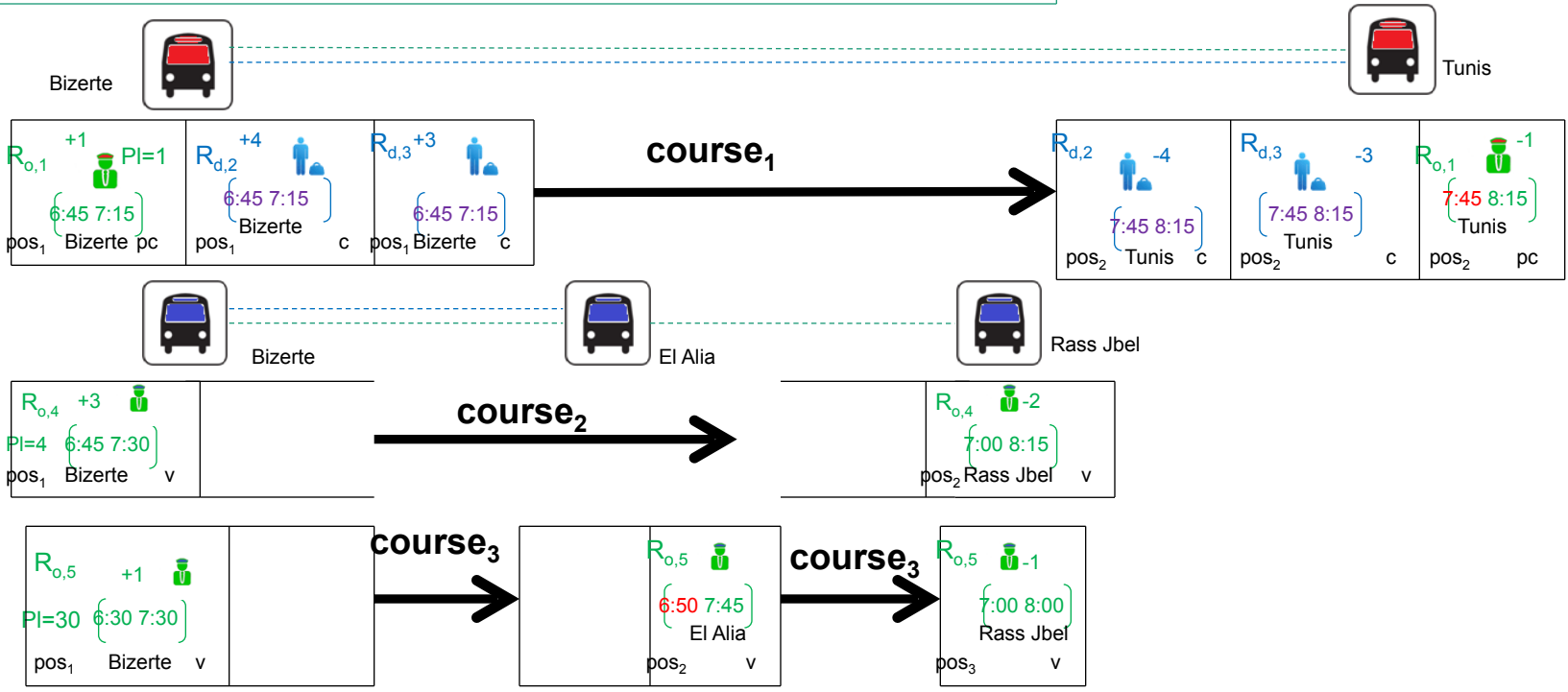
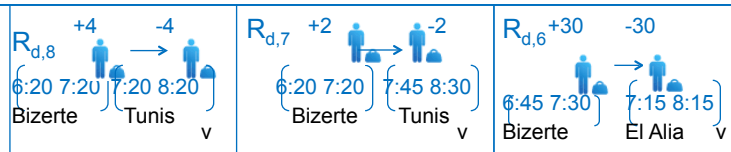
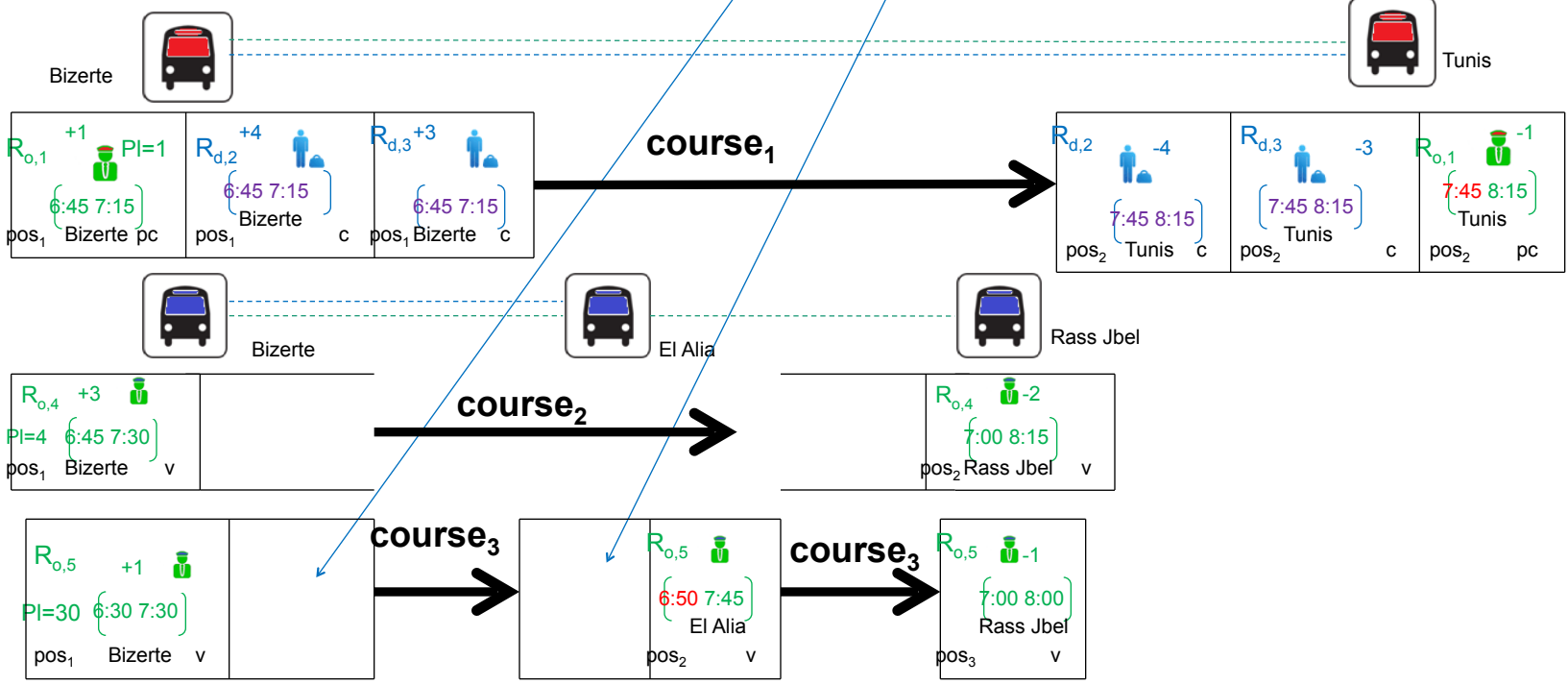
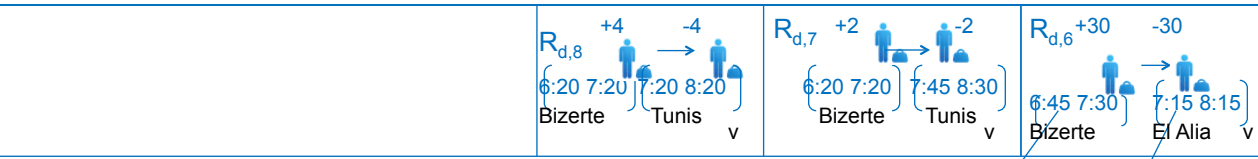


Illustration d'appariement des requêtes en appliquant GILA

Requêtes demande



Requêtes offre



6:00h

8:30h

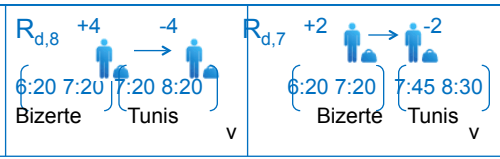
Bizerte

El Alia

Rass Jbel

Tunis

Illustration d'appariement des requêtes en appliquant GILA



Requêtes demande



Requêtes offre

ORGANISATION

Itération 1

6:00h

8:30h

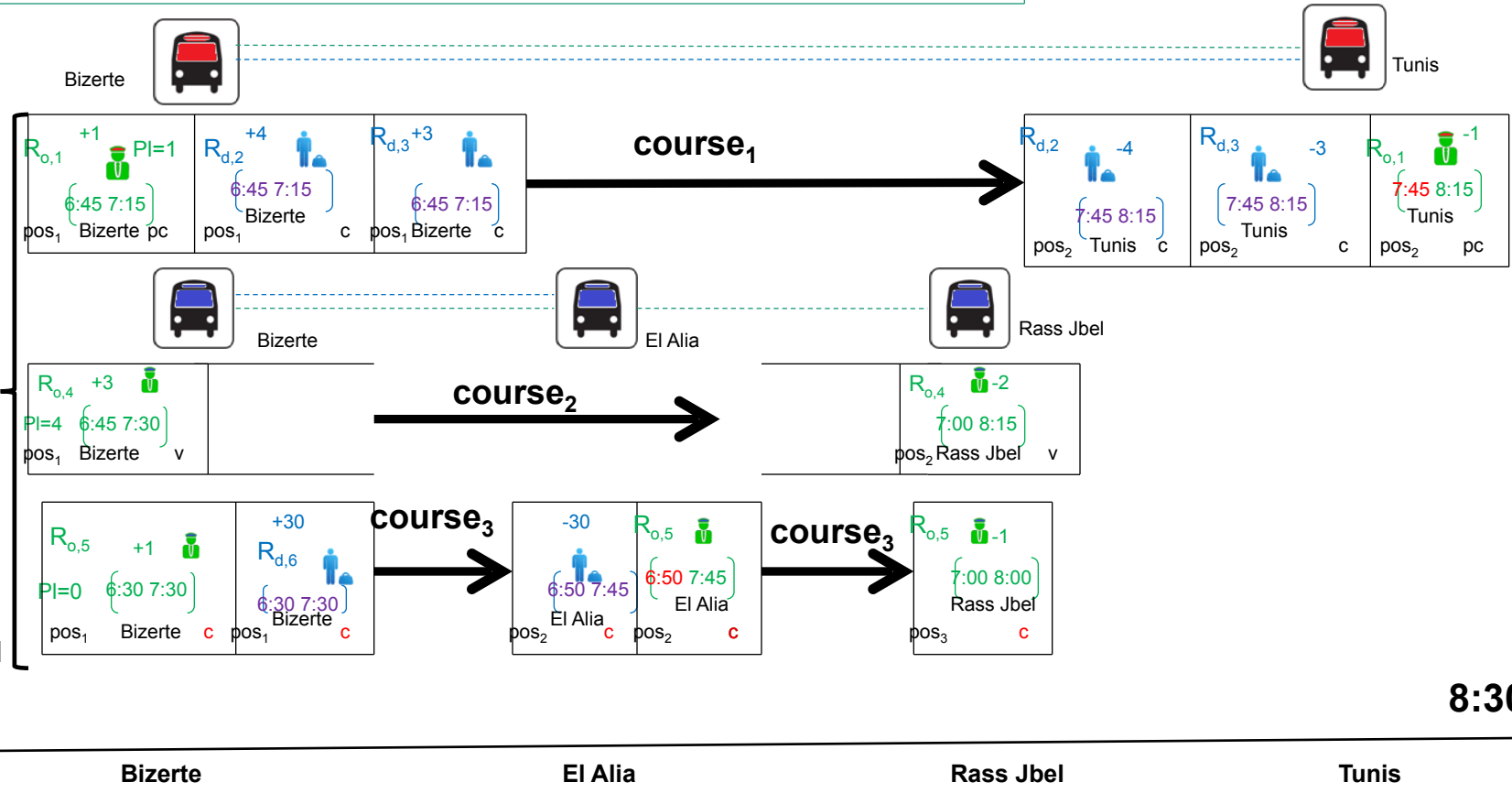
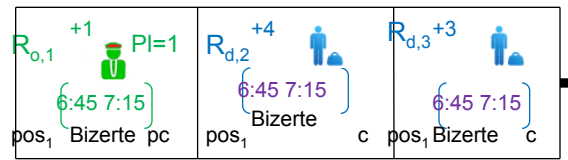
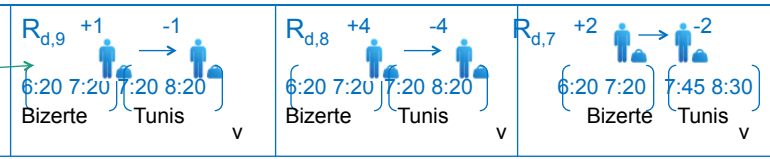


Illustration d'appariement des requêtes en appliquant GILA

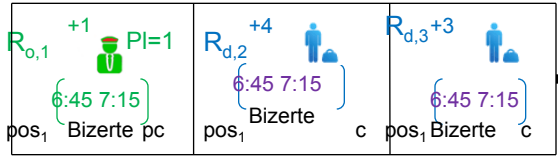
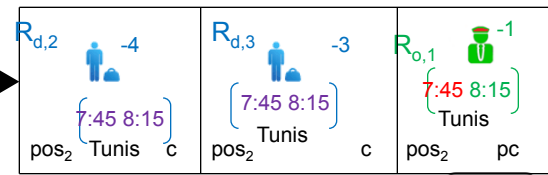
Requêtes demande



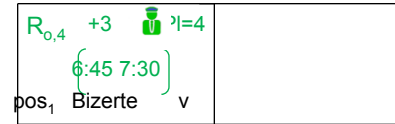
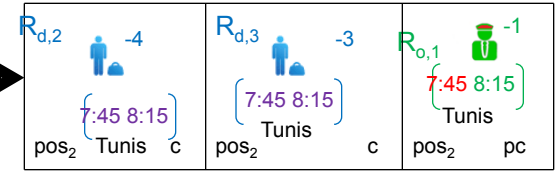
Une nouvelle requête
Demande est reçue
Itération 2



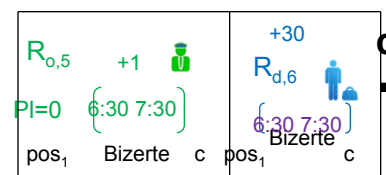
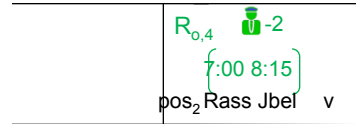
course₁



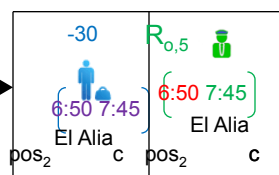
course₁



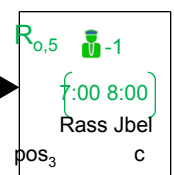
course₂



course₃



course₃



6:00h

8:30h

Bizerte

El Alia

Rass Jbel

Tunis

Illustration d'appariement des requêtes en appliquant GILA

Requêtes demande

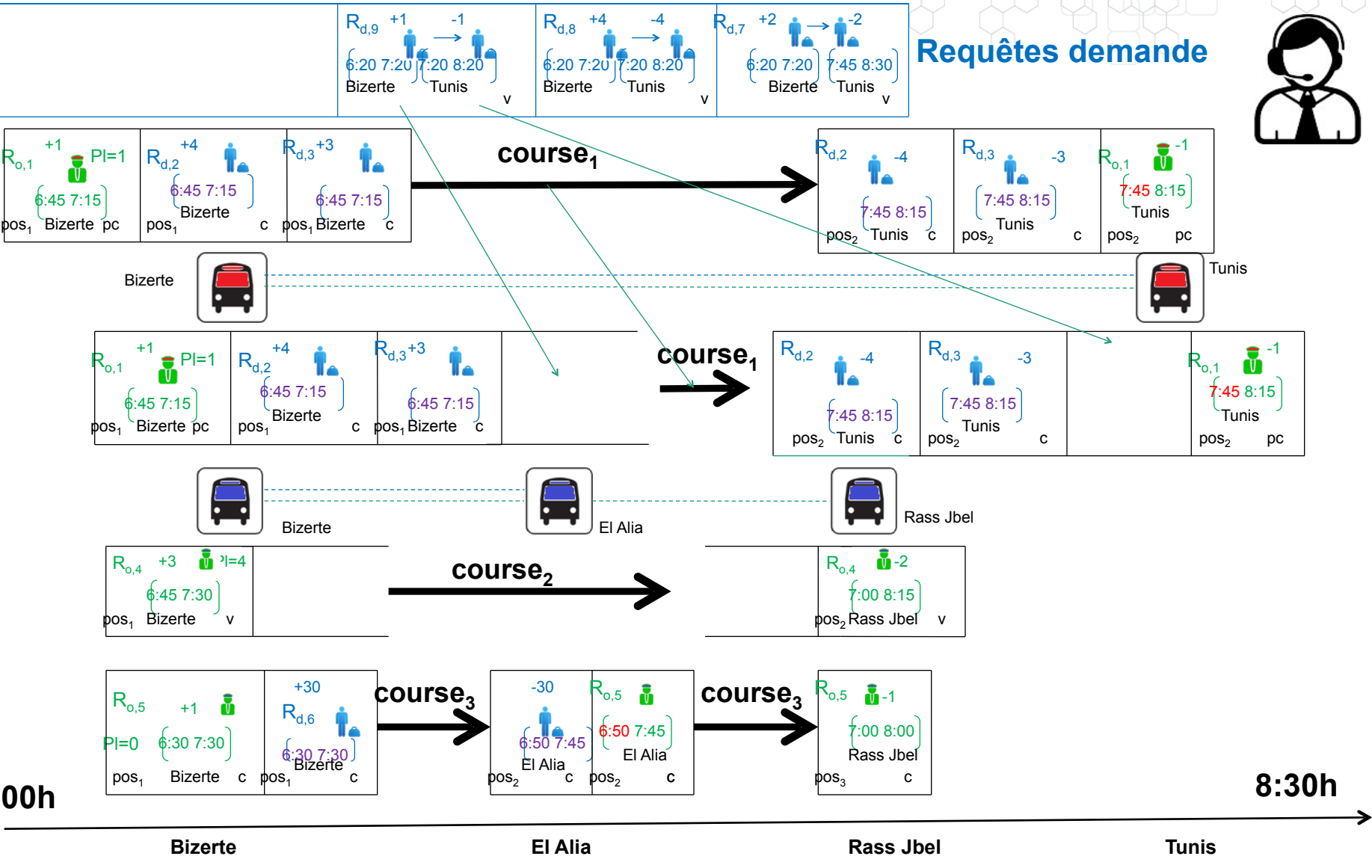
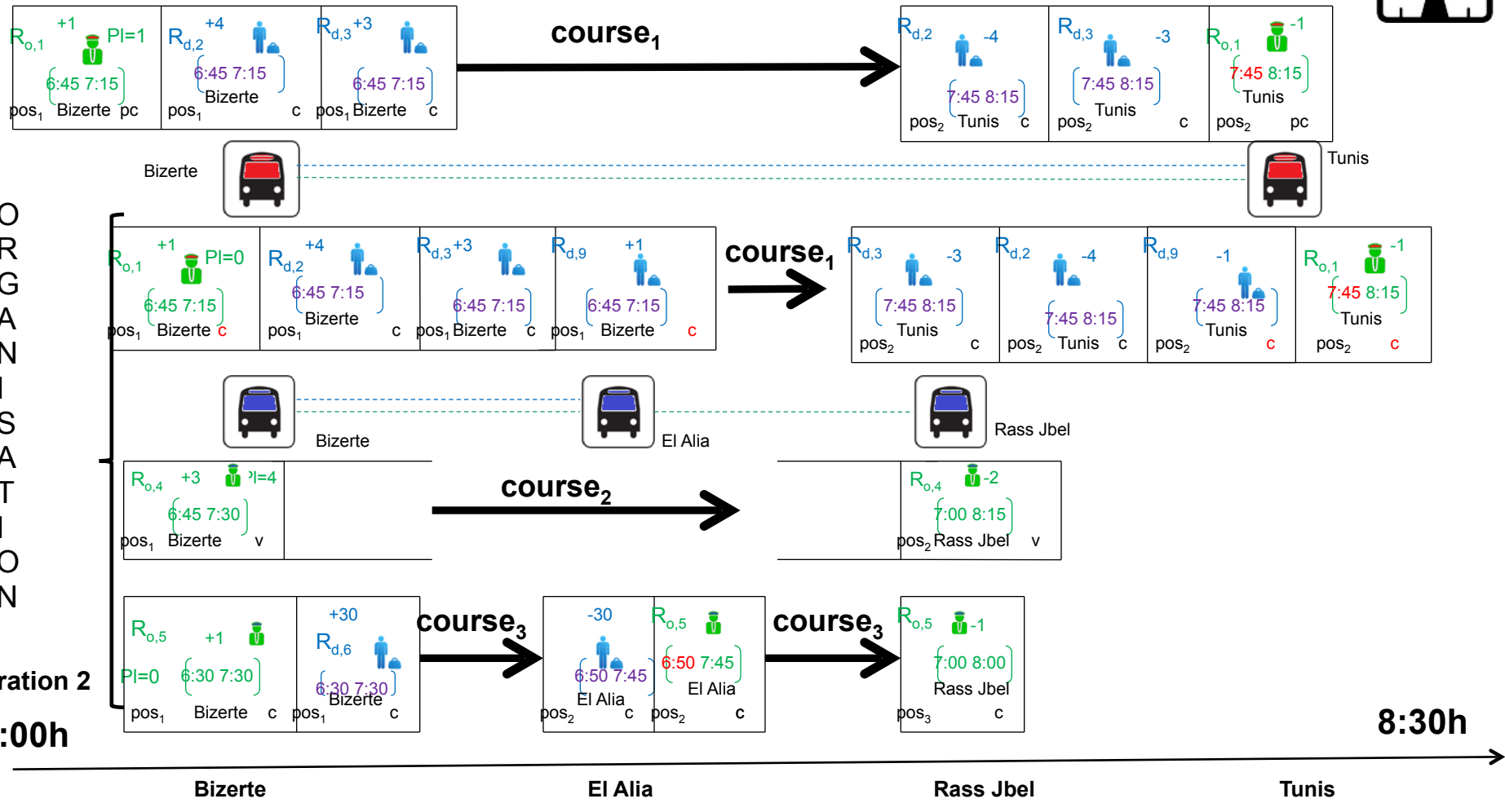
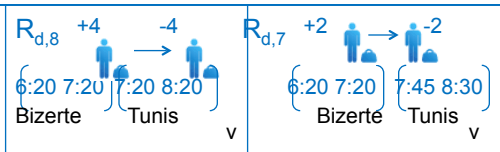


Illustration d'appariement des requêtes en appliquant GILA

Requêtes demande



Tests

Les tests sont effectués à partir d'un fichier de requêtes offre et un fichier de requêtes demande. A partir de la deuxième itération les tests vont être effectués aussi à partir d'un fichier organisation.

La structure d'une ligne du fichier des requêtes demande est la suivante :

idrequetedemande,itineraire{posi[dateauplustot/dateauplustard/pickup/delivery]}, nombredeplaces, montant,etat

```

requête_demandes - Bloc-notes
Fichier Edition Format Affichage ?
idrequetedemande,itineraire{posi[dateauplustot/dateauplustard/pickup/delivery]},nombredeplaces,montant,etat
4,bizerte/6/30/7/45/+1/0_tunis/6/30/8/45/0/-1,1,4.75,initialise
5,bizerte/6/30/7/45/+1/0_tunis/6/30/8/45/0/-1,1,4.75,initialise
6,bizerte/6/30/7/45/+1/0_tunis/6/30/8/45/0/-1,1,4.75,initialise
7,bizerte/6/30/7/45/+1/0_tunis/6/30/8/45/0/-1,1,4.75,initialise
11,bizerte/6/30/7/45/+1/0_tunis/6/30/8/45/0/-1,1,4.75,initialise
12,bizerte/6/30/7/45/+1/0_tunis/6/30/8/45/0/-1,1,4.75,initialise
13,bizerte/6/30/7/45/+1/0_tunis/6/30/8/45/0/-1,1,4.75,initialise
14,bizerte/6/30/7/45/+1/0_tunis/6/30/8/45/0/-1,1,4.75,initialise
15,sousse/6/30/7/30/+1/0_tunis/9/30/10/30/0/-1,1,5.5,initialise
16,sousse/6/30/7/30/+1/0_tunis/9/30/10/30/0/-1,1,5.5,initialise
17,sousse/6/30/7/30/+1/0_tunis/9/30/10/30/0/-1,1,5.5,initialise
18,sousse/6/30/7/30/+1/0_tunis/9/30/10/30/0/-1,1,5.5,initialise
19,sousse/6/30/7/30/+1/0_tunis/9/30/10/30/0/-1,1,5.5,initialise
23,sousse/6/30/7/30/+1/0_tunis/9/30/10/30/0/-1,1,5.5,initialise
24,sousse/6/30/7/30/+1/0_tunis/9/30/10/30/0/-1,1,5.5,initialise
25,sousse/6/30/7/30/+1/0_tunis/9/30/10/30/0/-1,1,5.5,initialise
26,sfax/10/00/11/30/+1/0_tunis/14/15/16/00/0/-2,1,18,initialise
27,sfax/10/00/11/30/+1/0_tunis/14/15/16/00/0/-2,1,18,initialise
28,sfax/10/00/11/30/+1/0_tunis/14/15/16/00/0/-2,1,18,initialise
29,sfax/10/00/11/30/+1/0_tunis/14/15/16/00/0/-2,1,18,initialise
30,sfax/10/00/11/30/+1/0_tunis/14/15/16/00/0/-2,1,18,initialise
31,sfax/10/00/11/30/+1/0_tunis/14/15/16/00/0/-2,1,18,initialise
32,sfax/10/00/11/30/+1/0_tunis/14/15/16/00/0/-2,1,18,initialise
33,sfax/10/00/11/30/+1/0_tunis/14/15/16/00/0/-2,1,18,initialise
34,sfax/10/00/11/30/+1/0_tunis/14/15/16/00/0/-2,1,18,initialise
35,bizerte/6/20/7/20/+3/0_rassjel/7/20/8/20/0/-3,3,7.5,initialise
36,bizerte/6/20/7/20/+3/0_rassjel/7/20/8/20/0/-3,3,7.5,initialise
37,bizerte/6/20/7/20/+3/0_rassjel/7/20/8/20/0/-3,3,7.5,initialise
38,bizerte/6/20/7/20/+3/0_rassjel/7/20/8/20/0/-3,3,7.5,initialise
39,nabeul/6/20/7/20/+1/0_tunis/8/20/9/20/0/-1,1,6.75,initialise
40,nabeul/6/20/7/20/+1/0_tunis/8/20/9/20/0/-1,1,6.75,initialise
41,nabeul/6/20/7/20/+1/0_tunis/8/20/9/20/0/-1,1,6.75,initialise
42,nabeul/6/20/7/20/+1/0_tunis/8/20/9/20/0/-1,1,6.75,initialise
43,tunis/6/15/7/15/+2/0_bizerte/7/45/8/30/0/-2,2,9.5,initialise
44,tunis/6/15/7/15/+2/0_bizerte/7/45/8/30/0/-2,2,9.5,initialise
45,tunis/6/15/7/15/+2/0_bizerte/7/45/8/30/0/-2,2,9.5,initialise
46,tunis/6/15/7/15/+2/0_bizerte/7/45/8/30/0/-2,2,9.5,initialise
47,bizerte/6/45/7/30/+1/0_elalia/7/45/8/30/0/-1,1,1.5,initialise
48,bizerte/6/45/7/30/+1/0_elalia/7/45/8/30/0/-1,1,1.5,initialise
49,bizerte/6/45/7/30/+1/0_elalia/7/45/8/30/0/-1,1,1.5,initialise
50,bizerte/6/45/7/30/+1/0_elalia/7/45/8/30/0/-1,1,1.5,initialise
51,tunis/6/30/7/30/+1/0_sousse/9/30/10/30/0/-1,1,5.5,initialise
52,tunis/6/30/7/30/+1/0_sousse/9/30/10/30/0/-1,1,5.5,initialise
53,tunis/6/30/7/30/+1/0_sousse/9/30/10/30/0/-1,1,5.5,initialise
54,tunis/6/30/7/30/+1/0_sousse/9/30/10/30/0/-1,1,5.5,initialise
55,tunis/6/20/7/20/+1/0_nabeul/8/20/9/20/0/-1,1,6.75,initialise
56,tunis/6/20/7/20/+1/0_nabeul/8/20/9/20/0/-1,1,6.75,initialise
57,tunis/6/20/7/20/+1/0_nabeul/8/20/9/20/0/-1,1,6.75,initialise
58,tunis/6/20/7/20/+1/0_nabeul/8/20/9/20/0/-1,1,6.75,initialise
59,tunis/09/45/10/30/+1/0_sfax/15/00/16/00/0/-1,1,18,initialise
60,tunis/09/45/10/30/+1/0_sfax/15/00/16/00/0/-1,1,18,initialise

```



Plateforme java 1.8.0



IntelliJ IDEA 2016.1.1

Tests

La structure d'une ligne du fichier des requêtes offre est la suivante :

idrequeteoffre, itineraire{posi[dateauplustôt/dateauplustard/pickup/delivery]}, chauffeur {idchauffeur/ typepermis}, vehicule{idvehicule/typevehicule/capacitevehicule}, placeslibresinitiales, placeslibresactuelles, etat

```

requetes_offres - Bloc-notes
Fichier Edition Format Affichage ?
idrequeteoffre,itineraire{posi[dateauplustôt/dateauplustard/pickup/delivery]},chauffeur{idchauffeur/typepermis},vehicule{idvehicule/typevehicule/capacitevehicule},placeslibresinitiales,placeslibresactuelles,etat
0,bizerte/6/45/7/15/+1/0_tunis/6/45/8/15/0/-1,0/d,0/rouge/8,8,8,initialise
1,bizerte/6/45/7/30/+1/0_rassibel/7/00/8/15/0/-1,1/d,1/bleu/8,8,8,initialise
2,sousse/6/45/7/15/+1/0_tunis/9/30/10/30/0/-1,3/d,3/rouge/8,8,8,initialise
3,sfax/9/45/10/15/+1/0_tunis/14/15/15/10/0/-1,5/d,5/rouge/8,8,8,initialise
8,nabeul/6/45/7/15/+1/0_tunis/9/15/10/00/0/-1,4/d,4/rouge/8,8,8,initialise
9,tunis/6/45/7/15/+1/0_bizerte/6/45/8/15/0/-1,6/d,2/rouge/8,8,8,initialise
10,bizerte/6/30/7/30/+1/0_elalia/6/45/7/45/0/0_rassibel/7/00/8/00/0/-1,2/d,6/bleu/8,8,8,initialise
20,tunis/6/45/7/15/+1/0_sousse/9/30/10/30/0/-1,8/d,7/rouge/8,8,8,initialise
21,tunis/6/45/7/15/+1/0_nabeul/9/15/10/00/0/-1,9/d,8/rouge/8,8,8,initialise
22,tunis/9/45/10/15/+1/0_sfax/14/15/15/10/0/-1,10/d,9/rouge/8,8,8,initialise
    
```

La structure d'une ligne du fichier organisation est la suivante:

idcourse, idoffre, itineraire{posi[dateauplustôt/dateauplustard/pickup/delivery]} iddemandes, listDemandesPositions, état, placesoccupées, placeslibres, chauffeur{idchauffeur/ typepermis}, véhicule{idvehicule/typevehicule/capacitevehicule}

```

OrganizationReturned0 - Bloc-notes
Fichier Edition Format Affichage ?
idcourse, idoffre, itineraire{posi[dateauplustôt/dateauplustard/pickup/delivery]} iddemandes, listDemandesPositions, état, placesoccupées, placeslibres, chauffeur
0,0,bizerte/6/45/7/15/+1/0_bizerte/6/30/7/45/+1/0_bizerte/6/30/7/45/+1/0_bizerte/6/30/7/45/+1/0_bizerte/6/30/7/45/+1/0_bizerte/6/30/7/45/+1/0_bizerte/6/30
1,1,bizerte/6/45/7/20/+1/0_bizerte/6/45/7/20/+3/0_bizerte/6/45/7/20/+3/0_rassibel/7/20/8/15/0/-3_rassibel/7/20/8/15/0/-3_rassibel/7/15/8/15/0/-1,35*36,351
2,2,sousse/6/45/7/15/+1/0_sousse/6/30/7/30/+1/0_sousse/6/30/7/30/+1/0_sousse/6/30/7/30/+1/0_sousse/6/30/7/30/+1/0_sousse/6/30/7/30/+
3,3,sfax/9/45/10/15/+1/0_sfax/10/0/11/30/+1/0_sfax/10/0/11/30/+1/0_sfax/10/0/11/30/+1/0_sfax/10/0/11/30/+1/0_sfax/10/0/11/30/+1/0_sfa
4,8,nabeul/6/45/7/15/+1/0_nabeul/6/20/7/20/+1/0_nabeul/6/20/7/20/+1/0_nabeul/6/20/7/20/+1/0_nabeul/6/20/7/20/+1/0_tunis/8/20/9/20/0/-1_tunis/8/20/9/20/0/-
5,9,tunis/6/45/7/15/+1/0_tunis/6/45/7/15/+2/0_tunis/6/45/7/15/+2/0_tunis/6/45/7/15/+2/0_tunis/6/45/7/15/+2/0_bizerte/7/45/8/15/0/-2_bizerte/7/45/8/15/0/-2
6,10,bizerte/6/30/7/20/+1/0_bizerte/6/30/7/20/+3/0_bizerte/6/30/7/20/+3/0_bizerte/6/45/7/30/+1/0_bizerte/6/45/7/30/+1/0_elalia/7/45/7/50/0/-1_elalia/7/45/
7,20,tunis/6/45/7/15/+1/0_tunis/6/30/7/30/+1/0_tunis/6/30/7/30/+1/0_tunis/6/30/7/30/+1/0_tunis/6/30/7/30/+1/0_sousse/9/30/10/30/0/-1_sousse/9/30/10/30/0/-
8,21,tunis/6/45/7/15/+1/0_tunis/6/20/7/20/+1/0_tunis/6/20/7/20/+1/0_tunis/6/20/7/20/+1/0_nabeul/8/20/9/20/0/-1_nabeul/8/20/9/20/0/-1
9,22,tunis/9/45/10/15/+1/0_tunis/9/45/10/30/+1/0_tunis/9/45/10/30/+1/0_sfax/15/0/15/10/0/-1_sfax/15/0/15/10/0/-1_sfax/14/15/15/10/0/-1,59*60,59|tunis/9/45
    
```

Tests

Premier tableau de tests



première itération						
config test	$\overline{R_{o,c}}$	$\overline{R_{o,pc}}$	$\overline{R_{o,v}}$	$\overline{R_{d,c}}$	$\overline{R_{d,v}}$	temps d'exécution (ms)
<i>config1</i> : paramOffre = 10/10/0/0/0, paramDemande = 50/50/0/0	5	4	1	47	3	23
<i>config2</i> : paramOffre = 20/20/0/0/0, paramDemande = 100/100/0/0	12	4	4	92	8	43
<i>config3</i> : paramOffre = 100/100/0/0/0, paramDemande = 500/500/0/0	49	10	31	364	136	379
<i>config4</i> : paramOffre = 500/500/0/0/0, paramDemande = 2500/2500/0/0	268	48	184	2028	472	2892
<i>config5</i> : paramOffre = 1000/1000/0/0/0, paramDemande = 5000/5000/0/0	537	94	369	4088	912	7361
<i>config6</i> : paramOffre = 2000/2000/0/0/0, paramDemande = 10000/10000/0/0	1075	186	739	8208	1792	14747

Le nombre des requêtes offre présente le 1/5 des requêtes demande

Deuxième tableau de tests



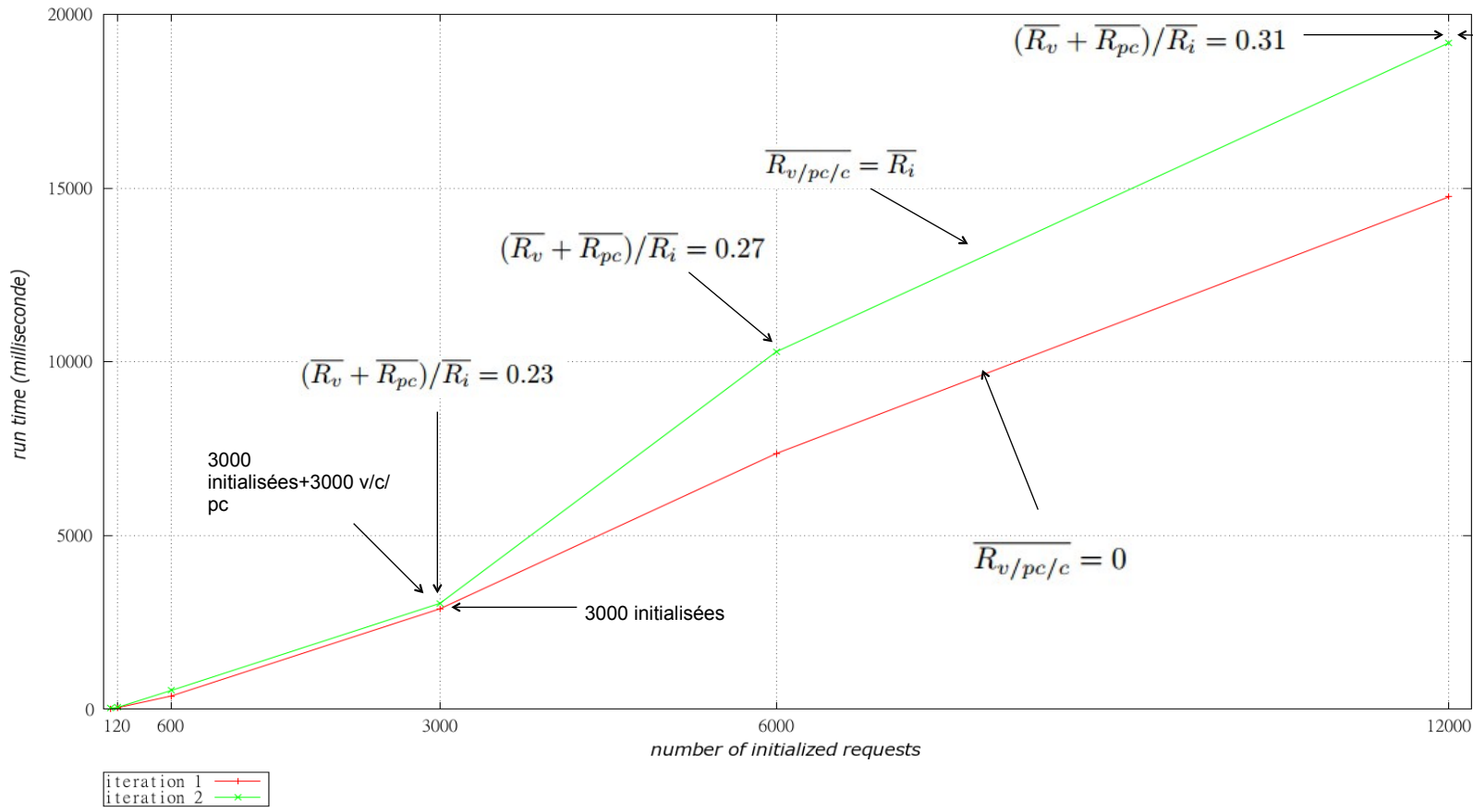
deuxième itération						
config test	$\overline{R_{o,c}}$	$\overline{R_{o,pc}}$	$\overline{R_{o,v}}$	$\overline{R_{d,c}}$	$\overline{R_{d,v}}$	temps d'exécution (ms)
<i>config1</i> : paramOffre = 20/10/1/5/4, paramDemande = 100/50/3/47	10	8	2	93	7	27
<i>config2</i> : paramOffre = 40/20/4/12/4, paramDemande = 200/100/8/92	24	10	6	187	13	50
<i>config3</i> : paramOffre = 200/100/31/49/10, paramDemande = 1000/500/136/364	120	26	54	882	118	538
<i>config4</i> : paramOffre = 1000/500/184/298/48, paramDemande = 5000/2500/472/2028	593	98	309	4416	584	3048
<i>config5</i> : paramOffre = 2000/1000/369/537/94, paramDemande = 10000/5000/912/4088	1075	186	739	8208	1792	10290
<i>config6</i> : paramOffre = 4000/2000/739/1075/186, paramDemande = 20000/10000/1792/8208	1613	278	2109	12328	7672	19183

Légende

config1: paramOffre = $\overline{R_o}/\overline{R_{o,i}}/\overline{R_{o,v}}/\overline{R_{o,c}}/\overline{R_{o,pc}}$; paramDemande = $\overline{R_d}/\overline{R_{d,i}}/\overline{R_{d,v}}, \overline{R_{d,c}}$

Analyse des résultats

run time according to the number of initialized requests for iteration 1 and iteration 2



24000 requêtes
4000 louage
40% du nombre des louages en 2014
Temps d'exécution 19s

l'évolution linéaire du temps d'exécution affirme la complexité théorique de l'algorithme glouton qui est de l'ordre de $O(n)$.

Conclusions

- ✚ Présentation du système de louage en Tunisie.
- ✚ Formalisation du problème multi-contraintes et multiobjectifs.
- ✚ Implémentation d'un algorithme d'appariement glouton incrémental et l'évaluation des performances de cet algorithme.

Perspectives

- ✚ Preuve de prototype en intégrant ce calculateur au sein d'un système de gestion des courses des louages.



Merci de votre attention

