

# Pickup and delivery (un véhicule).

## Données :

- un véhicule de capacité  $C$ ,
- un ensemble de requêtes de type

$$(\langle l_p, a_p, b_p \rangle, \langle l_d, a_d, b_d \rangle, t_{max})$$

$l_p, l_d$  : lieux de prise en charge et de dépose,  
 $(a_p, b_p), (a_d, b_d)$  : plages horaires correspondantes,  
éventuellement contrainte sur les durées des trajets ( $\leq t_{max}$ ),

- une distance  $d$ .

# Pickup and delivery (un véhicule).

**Problème** : trouver un trajet qui respecte les plages horaires et la capacité du véhicule (et les durées des trajets).

**Problème d'optimisation** : trouver le trajet le plus court qui satisfait les contraintes précédentes.

# Pickup and delivery (un véhicule).

## Contraintes :

plages horaires :

- **dépose** : la borne min n'existe pas en général, ou alors c'est pour une correspondance et dans ce cas il faudra que le véhicule arrive dans la plage (pas d'attente),
- **prise en charge** : il peut y avoir des bornes min, qui peuvent générer des temps d'attente : le véhicule attend la personne à l'endroit de prise en charge,

**capacité du véhicule** : contrainte dure,

**durées des trajets** : donnée, contrainte molle.

# Pickup and delivery (un véhicule).



## Optimisation :

- nombre de requêtes satisfaites,
- distance parcourue,
- temps en route,
- temps en charge (payé par les organismes),
- temps arrêté (payé par le transporteur).



# Solution.



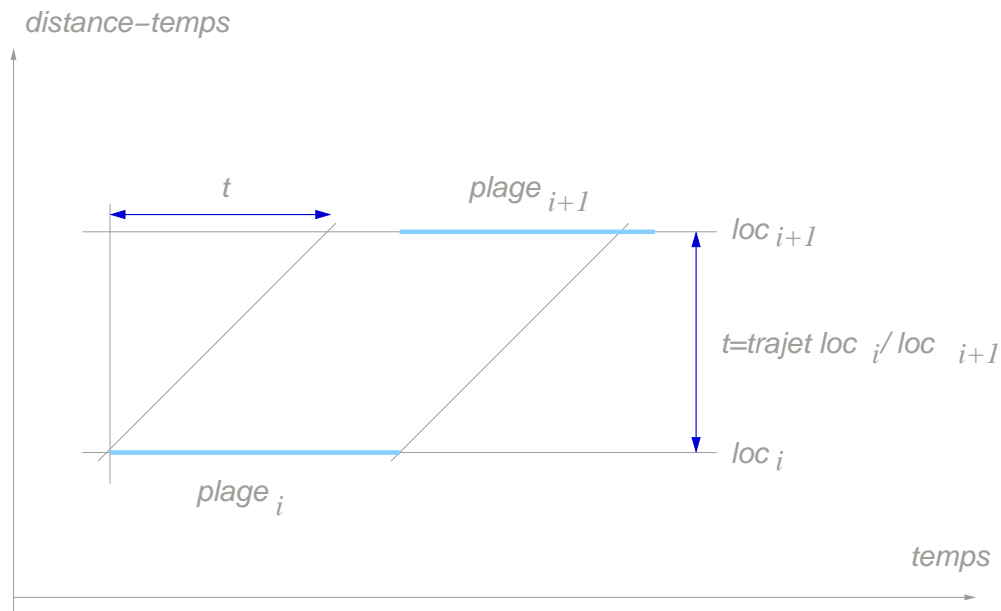
Une suite de  $n$  opérations  $(o_0, o_1, \dots, o_{n-1})$  :

- $o_0$  et  $o_{n-1}$  sont deux opérations bidon (servent à marquer le départ du dépôt initial et le retour au dépôt final),
- les autres opérations marchent par deux : pickup puis delivery,
- à aucun moment le nombre de requêtes en cours n'excède la capacité du véhicule,
- les plages horaires sont respectées (il existe des horaires de parcours qui les respectent).



# Consistance des plages.

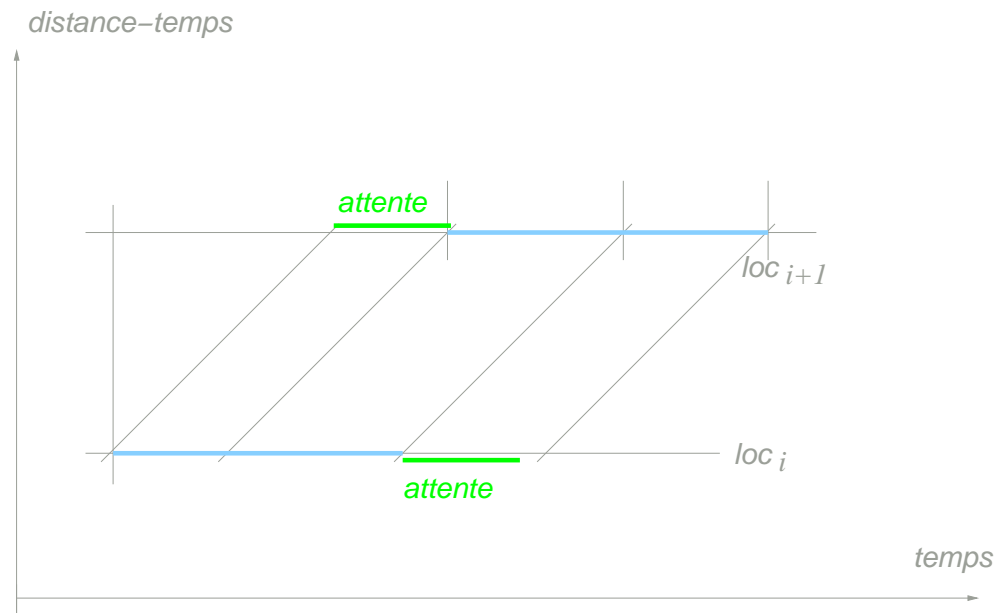
## interaction entre opérations voisines





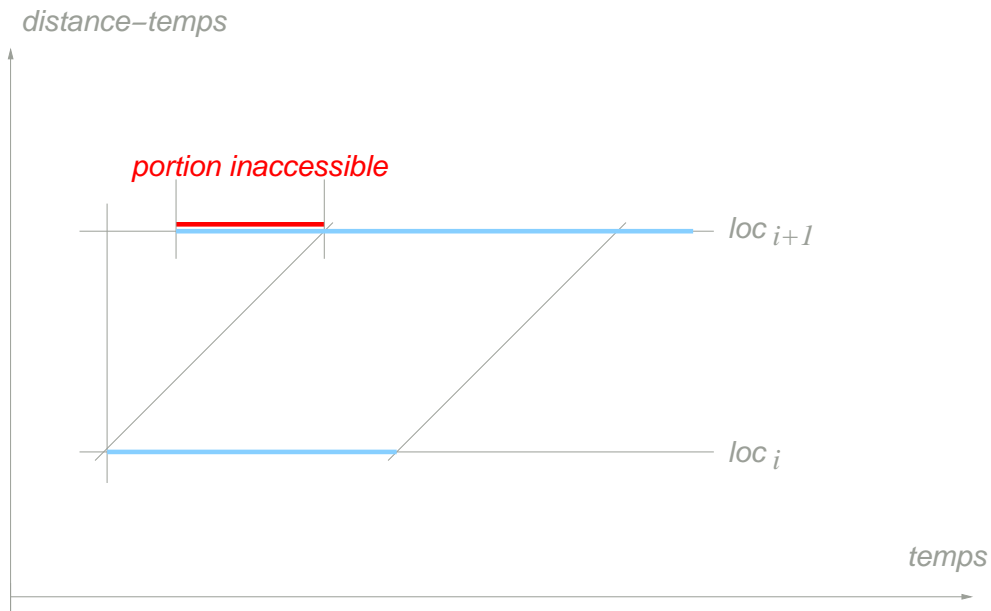
# Consistance des plages.

## • temps d'attente



# Consistance des plages.

- propagation de la borne minimale vers l'avant

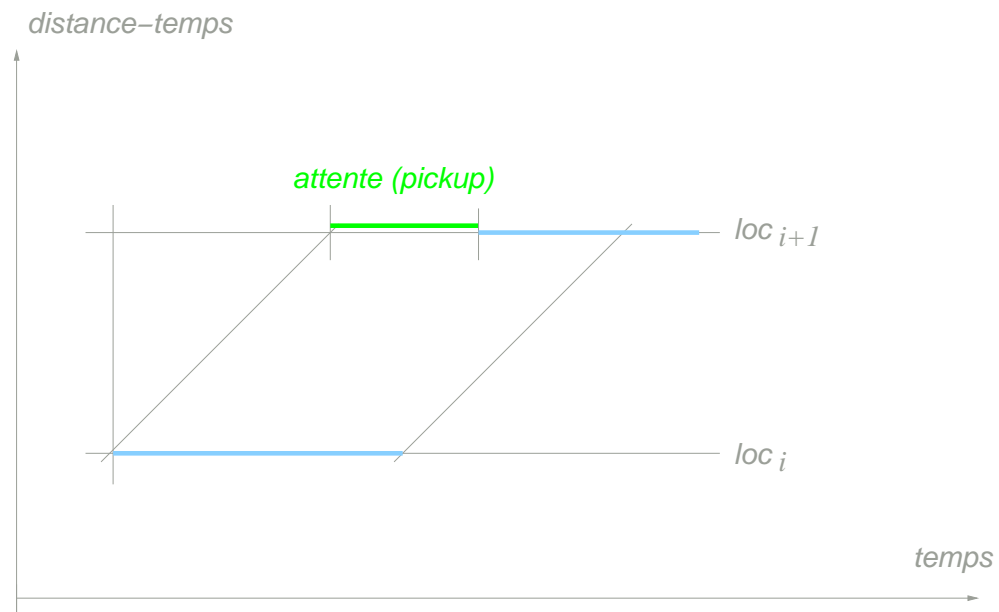






# Consistance des plages.

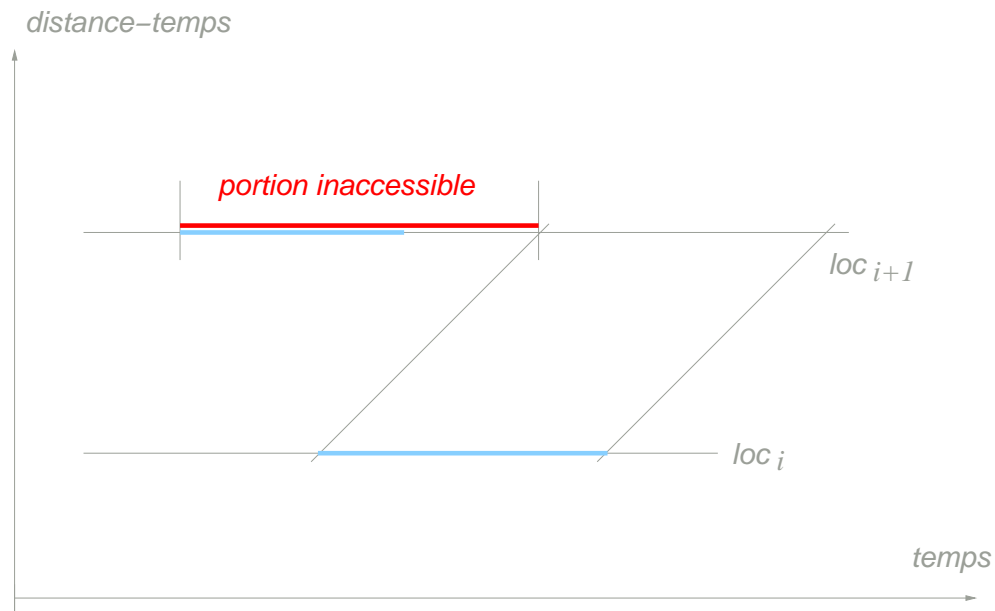
- borne minimale consistante



# Consistance des plages.



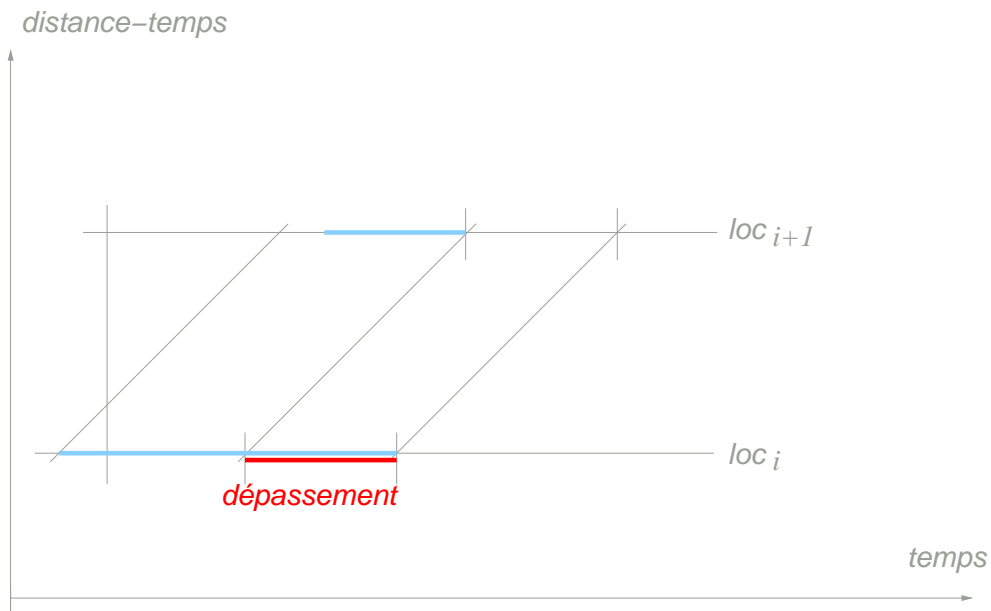
- échec : la propagation fait apparaître une inconsistance





# Consistance des plages.

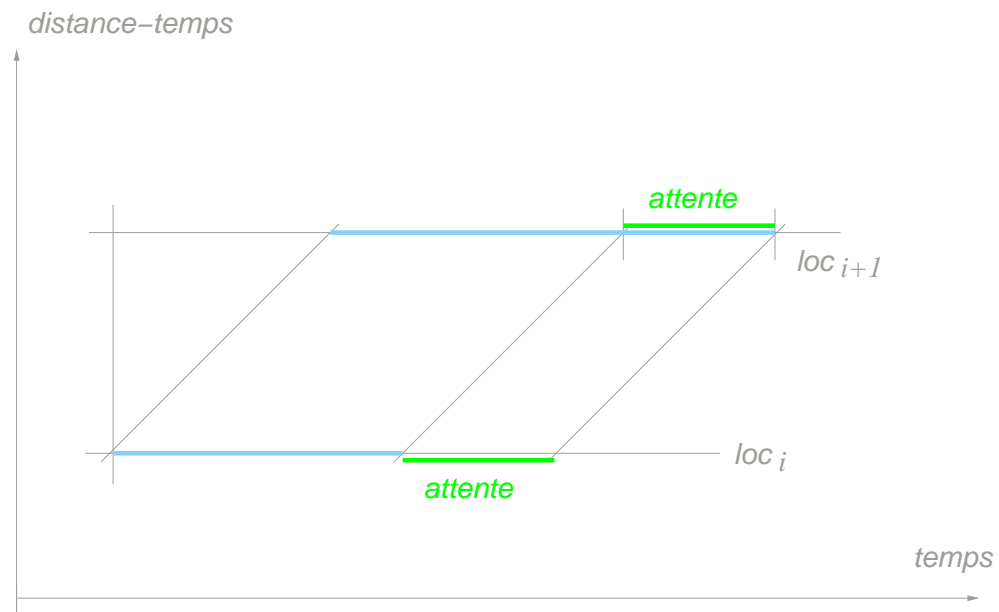
- propagation de la borne maximale vers l'arrière





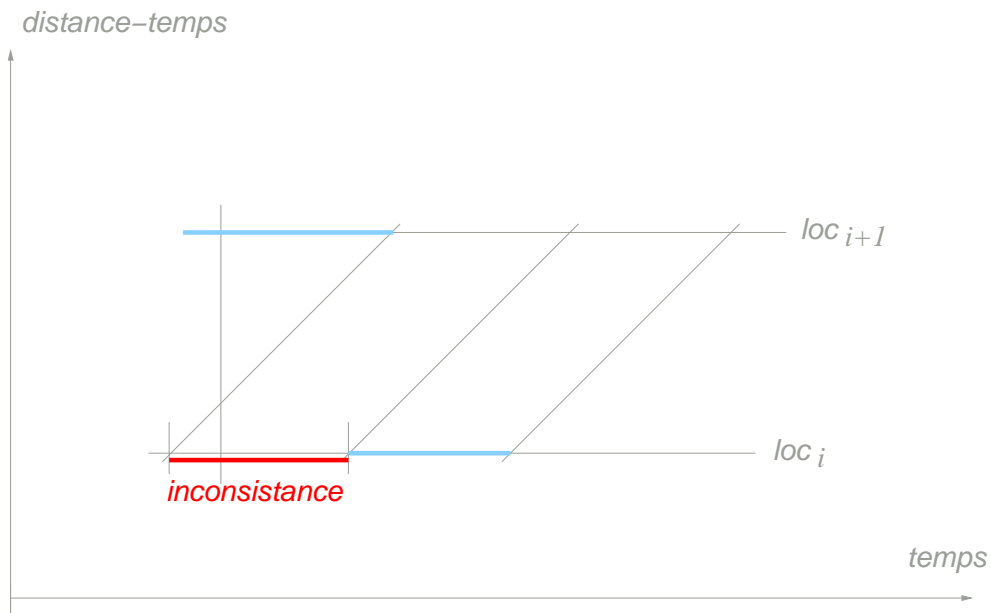
# Consistance des plages.

- borne maximale consistante



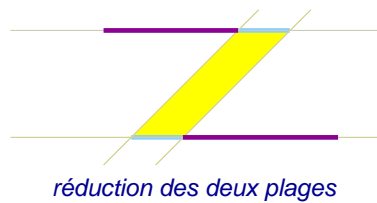
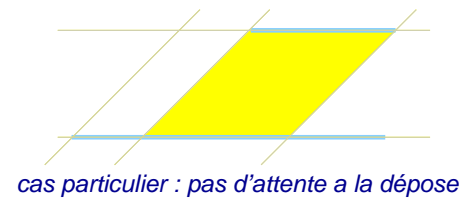
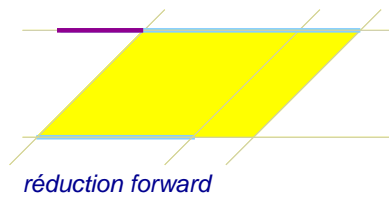
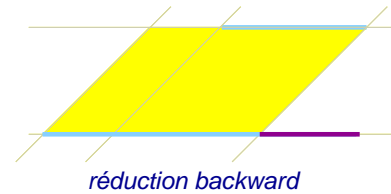
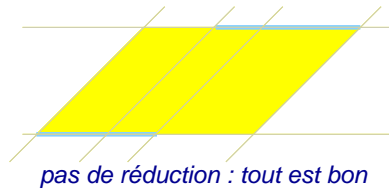
# Consistance des plages.

- échec : la propagation fait apparaître une inconsistance

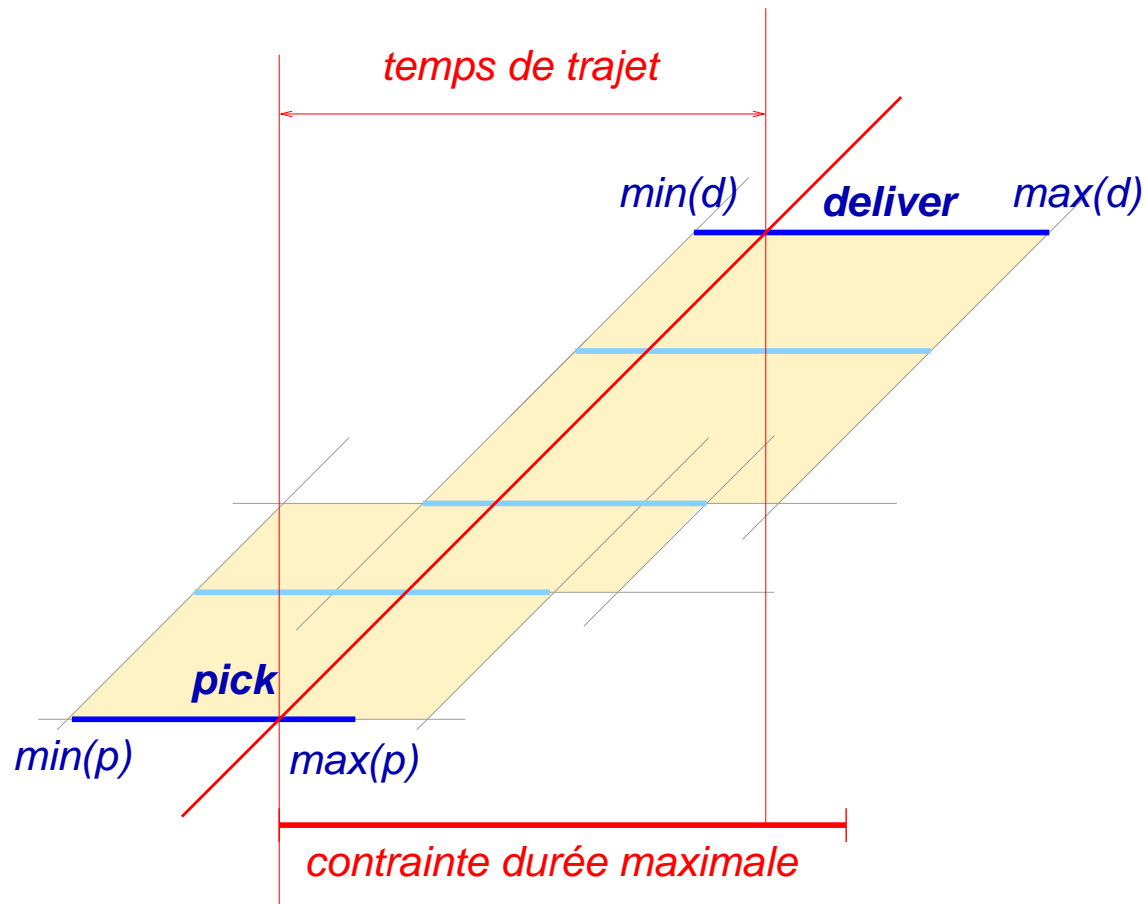




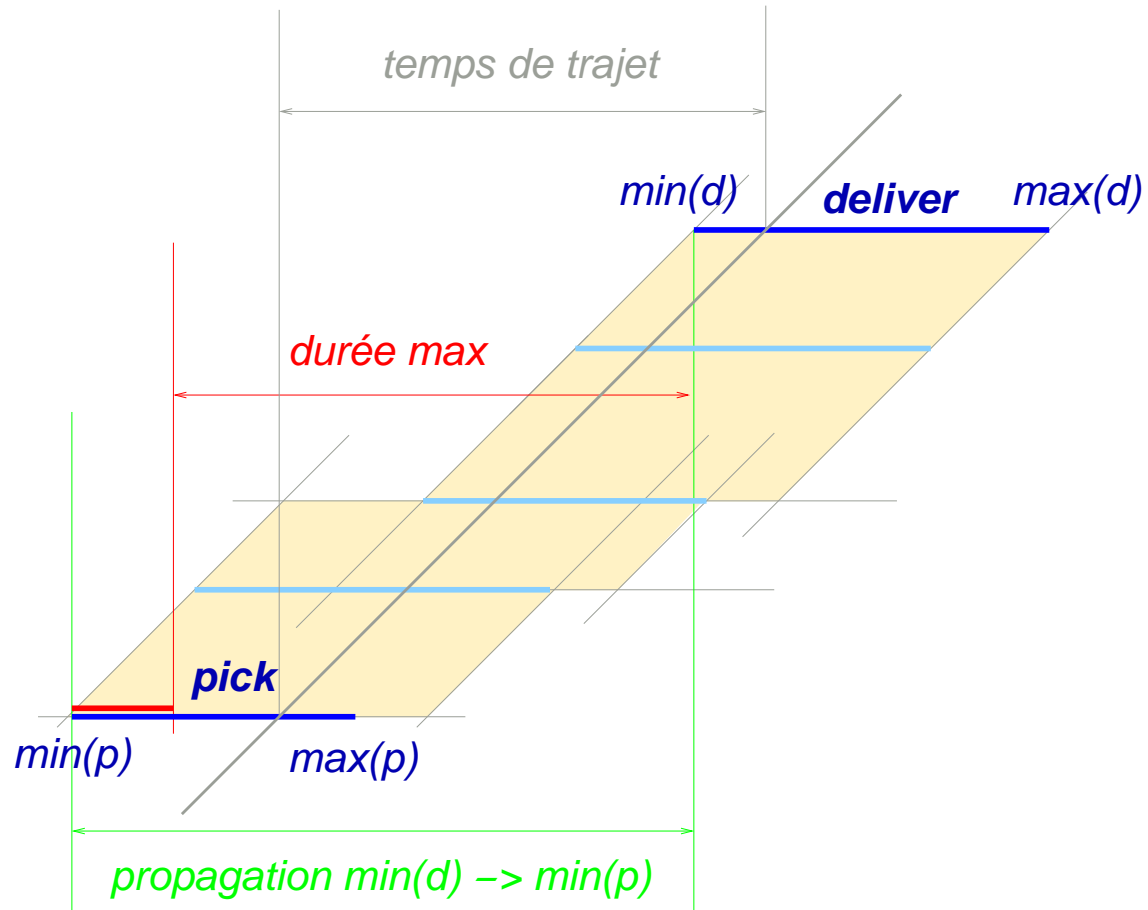
# Consistance des plages : récapitulation.



# Contrainte sur les durées.



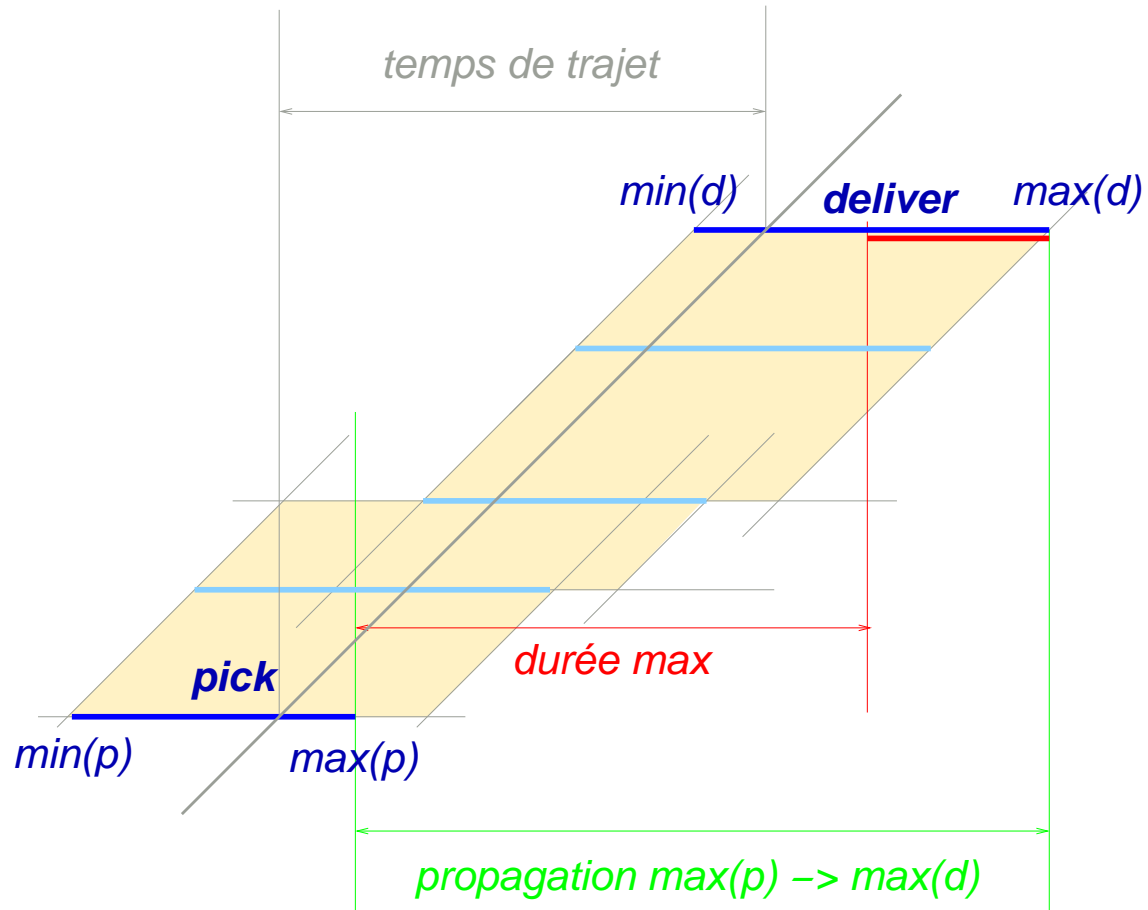
# Contrainte sur les durées.



Propagation  $min(d) \rightarrow min(p)$



# Contrainte sur les durées.



Propagation  $max(p) \rightarrow max(d)$

# Consistance : principe général.



## bornes min :

- partir du début de la suite d'opérations
- propager vers l'avant
- revenir en arrière si mise à jour durées

## bornes max :

- partir de la fin de la suite d'opérations
- propager vers l'arrière
- retourner en avant si mise à jour durées



# Recherche d'un parcours (ou route).



**phase initiale** : construction d'un parcours initial par insertion des requêtes,

**amélioration** : par des modifications élémentaires

- déplacement d'une opération dans la route (extraction puis insertion),
- échange entre opérations.



# Recherche d'un parcours (ou route).



## Prise en compte des différents paramètres :

- contraintes dures : satisfaites par une route consistante,
- contraintes molles : pénalités dans la fonction d'évaluation.

## Paramètres :

- capacité du véhicule,
- plages horaires,
- bornes sur les durées,
- longueur du parcours,
- nombre de véhicules (si il y a lieu),
- ...

