

---

# Contraintes TourSolver

GEOCONCEPT SAS

Contraintes des clients .....	4
Général .....	4
Localisation .....	7
Contraintes d'optimisation .....	9
Informations du contact .....	14
Satisfaction .....	16
Contraintes des visites .....	18
Nom ( <i>id</i> ) .....	18
Compatibilité avec les ressources ( <i>requiredSkills</i> ) .....	18
Exiger tous les critères pour être compatible ( <i>allSkillsRequired</i> ) .....	19
A visiter ( <i>active</i> ) .....	20
Quantité ( <i>quantity[0]</i> ) .....	20
Quantité 2, Quantité 3,...Quantité 24 ( <i>quantity[1],...quantity[23]</i> ) .....	21
Durée de chargement/déchargement unitaire ( <i>unloadingDurationPerUnit</i> ) .....	21
Type de visite ( <i>type</i> ) .....	22
Durée de la visite ( <i>fixedVisitDuration</i> ) .....	23
Heure d'ouverture 1, Heure de fermeture 1 ( <i>timeWindow[0].beginTime,timeWindow[0].endTime</i> ) .....	23
Heure d'ouverture 2, Heure de fermeture 2 ( <i>timeWindow[1].beginTime,timeWindow[1].endTime</i> ) .....	24
Heure d'ouverture 3, Heure de fermeture 3 ( <i>timeWindow[2].beginTime,timeWindow[2].endTime</i> ) .....	25
Heure d'ouverture 4, Heure de fermeture 4 ( <i>timeWindow[3].beginTime,timeWindow[3].endTime</i> ) .....	25
Jours d'ouverture 1 ( <i>possibleVisitDays[0]</i> ) .....	25
Jours d'ouverture 2 ( <i>possibleVisitDays[1]</i> ) .....	26
Jours d'ouverture 3 ( <i>possibleVisitDays[2]</i> ) .....	27
Jours d'ouverture 4 ( <i>possibleVisitDays[3]</i> ) .....	27
Terminer la visite avant l'heure de fermeture ( <i>wholeVisitInTimeWindow</i> ) .....	27
Tolérance sur le retard ( <i>punctuality</i> ) .....	28
Pénalité par heure de retard ( <i>delayPenaltyPerHour</i> ) .....	28
Fréquence de visite ( <i>frequency</i> ) .....	29
Date de dernière visite ( <i>tsOrderLastVisit</i> ) .....	30
Espacement minimal entre visites ( <i>tsOrderMinimumSpacing</i> ) .....	30
Durée d'une portion de visite découpée ( <i>minDuration</i> ) .....	31
Durée-seuil de découpe d'une visite ( <i>minPartDuration</i> ) .....	31
Exclure des ressources ( <i>excludeResources</i> ) .....	32
Affecter des ressources ( <i>assignResources</i> ) .....	32
Coût de sous-traitance ( <i>courierPenalty</i> ) .....	32
Nom de la ressource ( <i>evaluationInfos.orderOriginalResourceId</i> ) .....	33
Jour de visite ( <i>evaluationInfos.orderOriginalVisitDay</i> ) .....	33
Numéro d'ordre ( <i>evaluationInfos.orderPosition</i> ) .....	33
Autres données ( <i>customDataMap</i> ) .....	33
Contraintes des ressources .....	34

---

Nom ( <i>id</i> ) .....	34
Nom des départs et arrivées .....	34
Heures de travail ( <i>workStartTime,workEndTime</i> ) .....	34
Pause déjeuner .....	35
Heures supplémentaires .....	36
Coût horaire ( <i>workPenalty</i> ) .....	37
Coût kilométrique ( <i>travelPenalty</i> ) .....	37
Pondération de la vitesse ( <i>speedAdjustment</i> ) .....	37
Capacité du véhicule ( <i>capacity[0]</i> ) .....	38
Véhicule .....	38
Planning .....	40
Tournée .....	44
Chargement / Déchargement .....	47
Coûts .....	54
Nuitées .....	59
Législation de conduite .....	61
Contraintes des sites .....	63
Nom du site ( <i>id</i> ) .....	63
Horaires d'ouverture 1 ( <i>timeWindow[0].beginTime,timeWindow[0].endTime</i> ) .....	63
Horaires d'ouverture 2 ( <i>timeWindow[1].beginTime,timeWindow[1].endTime</i> ) .....	64
Horaires d'ouverture 3 ( <i>timeWindow[2].beginTime,timeWindow[2].endTime</i> ) .....	65
Horaires d'ouverture 4 ( <i>timeWindow[3].beginTime,timeWindow[3].endTime</i> ) .....	65
Jours d'ouverture 1 ( <i>openingDays[0]</i> ) .....	65
Jours d'ouverture 2 ( <i>openingDays[1]</i> ) .....	65
Jours d'ouverture 3 ( <i>openingDays[2]</i> ) .....	66
Jours d'ouverture 4 ( <i>openingDays[3]</i> ) .....	66
Gestion du site .....	66
Stock .....	67

## Contraintes des clients

- ❗ Dans cette documentation sont employés les mots-clés de la terminologie par défaut définie dans la page *Mon activité* de l'interface.  
Ainsi, on parlera ici de "client" mais les noms des contraintes peuvent évoluer en fonction de vos propres choix de terminologie.

### Général

#### Code client

Cette valeur permet d'identifier le client. Il s'agit d'un identifiant autogénéré par TourSolver et non éditable.

*Format : caractère*

Exemple :

1581948335603-+.124PdenmmwyADe6GVE

Elle ne peut pas être importée.

#### Référence

Cette valeur permet d'identifier le client. Le nom peut être un patronyme, un numéro, un identifiant...

*Format : caractère*

Exemple :

BL001

Cette contrainte doit être obligatoirement renseignée.

Elle peut être importée.

#### Type

Cette valeur permet segmenter la base de votre portefeuille.

Elle permet par exemple de segmenter Particulier / Entreprise, ou bien en catégorisant par activité ou chiffre d'affaire vos clients.

Cette valeur n'a pas d'impact sur l'optimisation mais peut servir à faire des filtres/Tri pour lancer des campagne de visites plus ciblé.

*Format : caractère*

Exemple :

---

## Opportunité

Elle peut être importée.

### Désignation

Il s'agit du nom complet de la société ou du particulier à servir.

*Format : caractère*

Exemple :

Geoconcept

Elle peut être importée.

### Description

Cette valeur permet d'ajouter un commentaire. Utile lorsque le client est une opportunité remonté depuis l'application mobile. Cela vous permet de remonter des informations rapidement sur la description du client, son accessibilité etc.. qui pourront être requalifiées ultérieurement.

*Format : caractère*

Exemple :

Editeur de logiciel géographique

Elle peut être importée.

### Client

Cette valeur vient en complément de la valeur Type.

Elle permet de savoir s'il s'agit d'une opportunité ou d'un client pour lequel vous avez un engagement de service à respecter. Pour les utilisateurs de l'application mobile, les opportunités que vous remontez du terrain ne sont pas considérées comme des clients (leur valeur est à "0").

Si cette valeur n'est pas importée alors la valeur par défaut est de "1".

*Format : booléen*

Exemple :

0

Elle peut être importée.

### Secteur

Nom libre ou identifiant par défaut défini pour la sectorisation.

*Format : caractère*

Exemple :

Territoire Paris

Elle peut être importée.

## Couleur

Permet de stocker la couleur d'un client code RGB/ couleur CSS.

*Format : caractère*

Exemple :

C97373

Elle peut être importée.

## Date de création

Cette valeur permet d'identifier le moment de création (par import ou manuellement) du client dans le système. Il s'agit d'un identifiant autogénéré par TourSolver et non éditable.

*Format : date et heure*

Exemple :

19/12/2019 09:08:31

Elle ne peut pas être importée.

## Date de mise à jour

Cette valeur permet d'identifier le moment de mise à jour (par import ou manuellement) du client dans le système. Il s'agit d'un identifiant autogénéré par TourSolver et non éditable.

*Format : date et heure*

Exemple :

19/12/2019 09:08:31

Elle ne peut pas être importée.

## Dernière mise à jour

Cette valeur permet d'identifier l'utilisateur ayant mis à jour (par import ou manuellement) le client dans le système. Il s'agit d'une valeur autogénérée par TourSolver et non éditable.

---

*Format : caractère*

Exemple :

Admin

Elle ne peut pas être importée.

## Localisation

### Adresse

Il s'agit du numéro, type et libellé de la voie. Ce champ est à remplir si vous ne disposez pas de coordonnées géographiques.

*Format : caractère*

Exemple :

162 avenue albert petit

Il n'est pas obligatoire mais fortement conseillé.

Elle peut être importée.

### Complément d'adresse

Ce champ permet de renseigner des compléments d'adresse. Cette information est utile dans la dernière phase de livraison pour que les utilisateurs de l'application mobile puissent trouver facilement la destination.

*Format : caractère*

Elle peut être importée.

### Ville

Il s'agit du seul champ obligatoire des champs d'adresse. Comme évoqué, nous vous conseillons de renseigner le maximum de champs adresses pour un maximum de précision.

*Format : caractère*

Exemple :

Bagneux

Cette contrainte doit être obligatoirement renseignée.

Elle peut être importée.

## Code postal

Il s'agit du code postal.

*Format : caractère*

Exemple :

92310

Elle peut être importée.

## Etat

*Format : caractère*

Elle peut être importée.

## Pays

*Format : caractère*

Exemple :

France

Elle peut être importée.

## Longitude

Si vous disposez des coordonnées géographiques de type WGS84, vous pouvez les renseigner ici.

*Format : réel*

Exemple :

2.9976290920

Elle peut être importée.

## Latitude

Si vous disposez des coordonnées géographiques de type WGS84, vous pouvez les renseigner ici.

*Format : réel*

Exemple :

48.988000772

Elle peut être importée.



---

## Type de géocodage

Ce champ est uniquement éditable par TourSolver.

*Format : caractère*

Exemple :

City

Elle ne peut pas être importée.

## Contraintes d'optimisation

### Organisation

Cette valeur permet de segmenter la base des clients en organisation, cette notion est rattaché aux droits. Ex : une ressource et/ou un planificateur faisant partie de l'organisation "Paris", ne verra pas un client indiqué "Bordeaux".

*Format : caractère*

Exemple :

Région PARIS

Elle peut être importée.

### Date de dernière visite

Cette contrainte permet de prendre en compte la date de la dernière visite effectuée qui précède la tournée à optimiser et ainsi respecter au mieux l'écart entre les visites.

*Format : date*

Exemple :

19/01/2019

Elle peut être importée.

### Date de prochaine visite

Cette valeur permet de donner l'information de la prochaine visite, si celle-ci a été partagée par le planificateur après une planification.

*Format : date*

Exemple :

30/03/2019

Elle ne peut pas être importée.

### Durée de visite

C'est le temps incompressible à passer sur la visite, c'est-à-dire un temps minimum de visite du client.

*Format : HH:MM:SS*

Exemple :

00:15:05

Elle peut être importée.

### Jours de visite possibles 1

Cette contrainte contient l'ensemble des Jours d'ouverture de visite (jusqu'à 64) sur lesquels s'applique le créneau horaire défini par les contraintes Heure d'ouverture 1 et Heure de fermeture 1.

Il est possible d'intégrer jusqu'à 4 jours d'ouverture.

*Format : numérique ou date*

Exemple :

1-5

Elle peut être importée.

### Jours de visite possibles 2

Cette contrainte contient l'ensemble des Jours d'ouverture de visite (jusqu'à 64) sur lesquels s'applique le créneau horaire défini par les contraintes Heure d'ouverture 2 et Heure de fermeture 2.

Il est possible d'intégrer jusqu'à 4 jours d'ouverture.

*Format : numérique ou date*

Exemple :

1-5

Elle peut être importée.

### Jours de visite possibles 3

Cette contrainte contient l'ensemble des Jours d'ouverture de visite (jusqu'à 64) sur lesquels s'applique le créneau horaire défini par les contraintes Heure d'ouverture 3 et Heure de fermeture 3.

Il est possible d'intégrer jusqu'à 4 jours d'ouverture.

*Format : numérique ou date*

---

Exemple :

1-5

Elle peut être importée.

### Jours de visite possibles 4

Cette contrainte contient l'ensemble des Jours d'ouverture de visite (jusqu'à 64) sur lesquels s'applique le créneau horaire défini par les contraintes Heure d'ouverture 4 et Heure de fermeture 4.

Il est possible d'intégrer jusqu'à 4 jours d'ouverture.

*Format : numérique ou date*

Exemple :

1-5

Elle peut être importée.

### Heure d'ouverture 1 et Heure de fermeture 1

Ces contraintes permettent de définir un créneau horaire dans lequel la visite peut être réalisée. Heure d'ouverture 1 correspond à l'heure à partir de laquelle la visite peut être réalisée. Heure de fermeture 1 correspond à l'heure au-delà de laquelle la visite ne peut plus être réalisée. Cette plage horaire s'appliquera à l'ensemble des jours sauf si Jours d'ouverture 1 est défini. Dans ce cas, la plage horaire est appliquée seulement à cette valeur.

Si une plage horaire n'est pas renseignée pour une visite donnée, la visite peut être réalisée n'importe quand, sur les éventuels Jours d'ouverture définis et dans la limite des horaires de travail de la ressource. On peut définir jusqu'à 4 créneaux horaires différents (et leurs jours associés) pour chaque visite.

*Format : HH:MM:SS*

Exemple Heure d'ouverture 1 :

09:50:00

Exemple Heure de fermeture 1 :

10:50:00

Elles peuvent être importées.

### Heure d'ouverture 2 et Heure de fermeture 2

Ces contraintes permettent de définir un créneau horaire dans lequel la visite peut être réalisée. Heure d'ouverture 2 correspond à l'heure à partir de laquelle la visite peut être réalisée. Heure de fermeture 2 correspond à l'heure au-delà de laquelle la visite ne peut plus être réalisée. Cette plage horaire

s'appliquera à l'ensemble des jours sauf si Jours d'ouverture 2 est défini. Dans ce cas, la plage horaire est appliquée seulement à cette valeur.

Si une plage horaire n'est pas renseignée pour une visite donnée, la visite peut être réalisée n'importe quand, sur les éventuels Jours d'ouverture définis et dans la limite des horaires de travail de la ressource. On peut définir jusqu'à 4 créneaux horaires différents (et leurs jours associés) pour chaque visite.

*Format : HH:MM:SS*

Exemple Heure d'ouverture 2 :

09:50:00

Exemple Heure de fermeture 2 :

10:50:00

Elles peuvent être importées.

### Heure d'ouverture 3 et Heure de fermeture 3

Ces contraintes permettent de définir un créneau horaire dans lequel la visite peut être réalisée. Heure d'ouverture 3 correspond à l'heure à partir de laquelle la visite peut être réalisée. Heure de fermeture 3 correspond à l'heure au-delà de laquelle la visite ne peut plus être réalisée. Cette plage horaire s'appliquera à l'ensemble des jours sauf si Jours d'ouverture 3 est défini. Dans ce cas, la plage horaire est appliquée seulement à cette valeur.

Si une plage horaire n'est pas renseignée pour une visite donnée, la visite peut être réalisée n'importe quand, sur les éventuels Jours d'ouverture définis et dans la limite des horaires de travail de la ressource. On peut définir jusqu'à 4 créneaux horaires différents (et leurs jours associés) pour chaque visite.

*Format : HH:MM:SS*

Exemple Heure d'ouverture 3 :

09:50:00

Exemple Heure de fermeture 3 :

10:50:00

Elles peuvent être importées.

### Heure d'ouverture 4 et Heure de fermeture 4

Ces contraintes permettent de définir un créneau horaire dans lequel la visite peut être réalisée. Heure d'ouverture 4 correspond à l'heure à partir de laquelle la visite peut être réalisée. Heure de fermeture 4 correspond à l'heure au-delà de laquelle la visite ne peut plus être réalisée. Cette plage horaire s'appliquera à l'ensemble des jours sauf si Jours d'ouverture 4 est défini. Dans ce cas, la plage horaire est appliquée seulement à cette valeur.

Si une plage horaire n'est pas renseignée pour une visite donnée, la visite peut être réalisée n'importe quand, sur les éventuels Jours d'ouverture définis et dans la limite des horaires de travail de la ressource. On peut définir jusqu'à 4 créneaux horaires différents (et leurs jours associés) pour chaque visite.

*Format : HH:MM:SS*

Exemple Heure d'ouverture 4 :

09:50:00

Exemple Heure de fermeture 4 :

10:50:00

Elles peuvent être importées.

## Compétences requises

Cette contrainte désigne les critères requis par le client que la ressource doit suivre pour pouvoir visiter celui-ci.

*Format : caractère - liste de mots séparés par une virgule (maximum 64 mots différents pour l'ensemble des fichiers visites et ressources traités)-*

Exemple :

robert,ana

Elle peut être importée.

## Fréquence

Cette contrainte permet de spécifier qu'une visite doit être réalisée plusieurs fois sur une période de plusieurs jours. TourSolver tente d'optimiser en équilibrant au mieux les écarts entre les visites en tenant compte également des autres contraintes. La fréquence ne garantit pas le respect exact des périodes. Un indicateur de non-respect des périodes de fréquence permet dans l'interface TourSolver de connaître le nombre de visites n'étant pas en période optimale de fréquentation.

*Format : caractère*

Exemple :

1/5

Elle peut être importée.

## Espace minimal entre visites

Cette contrainte permet de respecter un écart minimum entre 2 visites chez un même client.

*Format : entier*

Exemple :

5

Elle peut être importée.

## Espace maximal entre visites

Cette contrainte permet de respecter un écart maximum entre 2 visites chez un même client.

*Format : entier*

Exemple :

5

Elle peut être importée.

## Type de fréquence

Cette valeur est descriptive et permet uniquement de filtrer les données plus simplement dans l'interface.

*Format : caractère*

Exemple :

Hebdomadaire

Elle peut être importée.

## Informations du contact

### Prénom

Cette valeur permet de renseigner le prénom d'un contact.

*Format : caractère*

Exemple :

Alexis

Elle peut être importée.

### Nom

Cette valeur permet de renseigner le nom d'un contact.

*Format : caractère*

Exemple :

---

BERGER

Elle peut être importée.

## Occupation

Cette valeur permet de renseigner le poste d'un contact.

*Format : caractère*

Exemple :

Product Owner

Elle peut être importée.

## Adresse e-mail

Cette valeur permet de renseigner l'email un contact.

Cette information sera utilisée dans l'envoi des notifications ""Expérience client"" si la fonction est activée dans TourSolver et si le champ Recevoir les notifications pour le client est actif.

*Format : caractère*

Exemple :

[alexis.berger@geoconcept.com](mailto:alexis.berger@geoconcept.com) [mailto:alexis.berger@geoconcept.com]

Elle peut être importée.

## Recevoir les notifications

Cette valeur permet d'autoriser l'envoi de SMS ou mail à un contact.

Par défaut la valeur est à 0.

*Format : booléen*

Exemple :

0

Elle peut être importée.

## Téléphone fixe

Cette valeur est un complément d'information et n'est pas utilisée dans TourSolver.

*Format : caractère*

Exemple :

---

0172749887

Elle peut être importée.

## Téléphone mobile

Cette valeur permet d'envoyer des SMS à un contact.

Cette information sera utilisée dans l'envoi des notifications ""Expérience client"" si la fonction est activée dans TourSolver et si le champ Recevoir les notifications pour le client est actif.

*Format : caractère*

Exemple :

0639876719

Elle peut être importée.

## Satisfaction

### Dernier avis

Cette valeur contient la note du destinataire (de 1 à 5) lorsque celle-ci est recueillie depuis les "notifications expérience". Elle est non éditable.

*Format : entier*

Exemple :

5

Elle ne peut pas être importée.

### Dernier commentaire

Cette valeur contient le commentaire du client lorsque celui-ci est recueilli depuis les "notifications expérience".

*Format : caractère*

Exemple :

Le livreur m'a bien aidé !

Elle ne peut pas être importée.

### Id de dernière visite

Cette valeur contient l'identifiant de la dernière visite réalisée sur ce client.



---

*Format : entier*

Exemple :

BL00981

Elle ne peut pas être importée.

## URL photo

Cette valeur permet de stocker les photos prises depuis le terrain par les utilisateurs de l'application mobile.

*Format : URL (liste)*

Exemple :

<http://geoconcept/photo13>, <http://geoconcept/photo21>

Elle ne peut pas être importée.

## Contraintes des visites

- ⓘ Dans cette documentation sont employés les mots-clés de la terminologie par défaut définie dans la page *Mon activité* de l'interface.  
Ainsi, on parlera ici de "visite" mais les noms des contraintes peuvent évoluer en fonction de vos propres choix de terminologie.

- ⓘ Après chaque nom de contrainte, les parenthèses contiennent le nom équivalent de cette contrainte dans [TourSolver Cloud API](https://geoservices.geoconcept.com/ToursolverCloud/api-book.html) [https://geoservices.geoconcept.com/ToursolverCloud/api-book.html].

### Nom (*id*)

Cette valeur permet d'identifier la visite. Le nom peut être un patronyme, un numéro, un identifiant... Celle-ci est obligatoire.

*Format : caractère*

Exemple :

Mr Dupont / BL0012

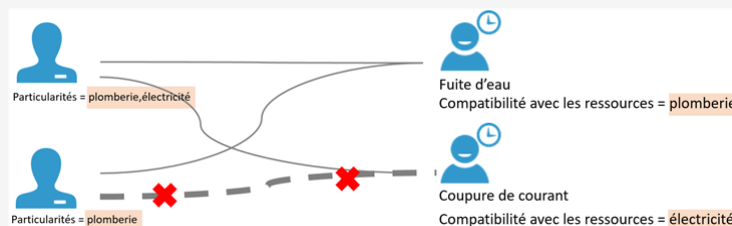
### Compatibilité avec les ressources (*requiredSkills*)

Cette contrainte désigne les critères requis par la visite que la ressource doit suivre pour pouvoir réaliser celle-ci.

*Format : liste de mots séparés par une virgule (maximum 64 mots différents pour l'ensemble des fichiers visites et ressources traités)*

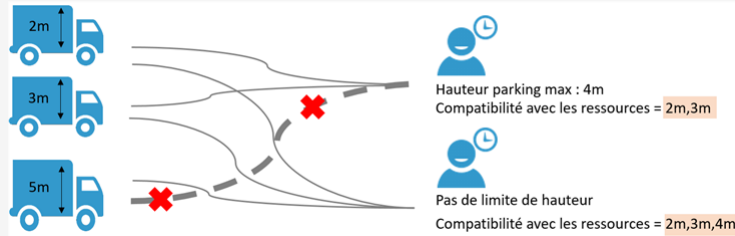
Exemple 1 :

Pour un client ayant besoin d'une intervention de maintenance concernant la plomberie, on pourra indiquer "plomberie". Il faudra qu'au moins une des ressources dispose de ce critère pour que le client soit visité.



Exemple 2 :

Pour un client dont l'accès est limité à une hauteur de 4 m, on indiquera "2m,3m" parce qu'il ne peut être livré que par un véhicule inférieur à 4 m. C'est une des ressources au critère 2 m ou 3 m qui pourra le visiter.



## Exiger tous les critères pour être compatible (*allSkillsRequired*)

Si la contrainte Compatibilité avec les ressources est utilisée (et que les valeurs sont nombreuses), cette contrainte permet d'indiquer si un seul des critères de la liste est requis ou si la liste complète des critères est requise.

*Format : binaire*

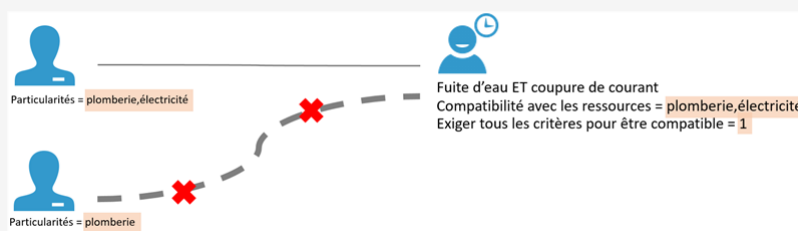
Si la valeur = 0, un seul critère est requis.

Si la valeur = 1, la liste complète de tous les critères est requise.

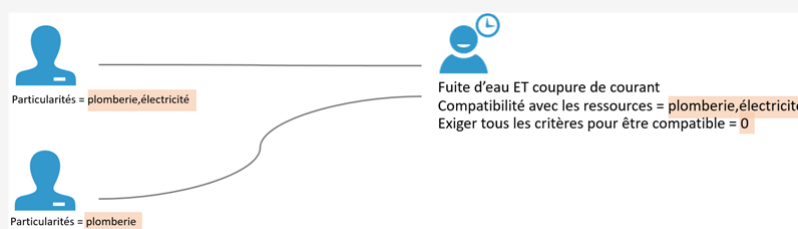
La valeur par défaut est égale 1.

Exemple 1 :

Pour un client ayant besoin d'une intervention de maintenance concernant la plomberie ET l'électricité, on pourra indiquer "plomberie,électricité". Il faudra qu'au moins une des ressources dispose de ces 2 critères pour que le client soit visité. Il faut donc mettre la valeur 1 dans la contrainte Exiger tous les critères pour être compatible.



Si la contrainte Exiger tous les critères pour être compatible = 0 alors la prise en compte des 2 critères ne sera pas demandée.



## A visiter (*active*)

Cette contrainte permet de prendre en considération les visites à réaliser sans avoir à ajouter ou enlever des données dans le jeu de données complet.

*Format : binaire*

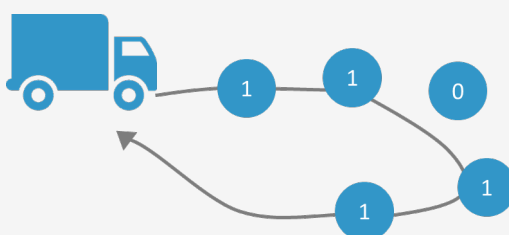
Si la valeur = 0, la visite n'est pas intégrée dans la tournée.

Si la valeur = 1, la visite est réalisée.

La valeur par défaut est égale à 1.

Exemple :

Sur 5 visites, 1 seule ne sera pas intégrée à cette tournée.



## Quantité (*quantity[0]*)

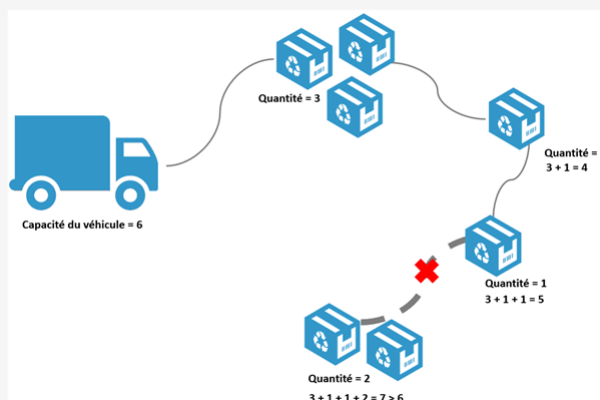
Cette valeur représente la quantité à livrer ou à collecter. L'utilisateur détermine l'unité de la quantité (kilos, m3, litres...) et assure la cohérence avec la contrainte Capacité du véhicule de la ressource.

*Format : numérique*

Les valeurs possibles pour cette contrainte vont de 0 à 2147483. TourSolver ne reconnaît pas plus de 3 chiffres significatifs après la virgule.

Exemple :

Dans le cas de livraison de colis, on indique le nombre de colis à livrer (3 pour 3 colis), dans le cas de matière liquide on indique le nombre de litres (3000 pour 3000 litres), dans le cas de gros conditionnements on indique le volume (2 pour 2m3).



- ! Une visite ne peut être desservie que par une seule ressource unique. Une visite ne peut pas être réalisée par plusieurs ressources dont les capacités respectives additionnées pourraient satisfaire la quantité demandée. Ainsi, une visite dont une contrainte Quantité dépasse la contrainte correspondante Capacité du véhicule maximale des ressources est considérée comme impossible et n'est pas prise en compte dans la résolution du problème. Pour contourner le problème, l'utilisateur peut décomposer la visite en plusieurs visites.

## Quantité 2, Quantité 3,...Quantité 24 (*quantity[1],...quantity[23]*)

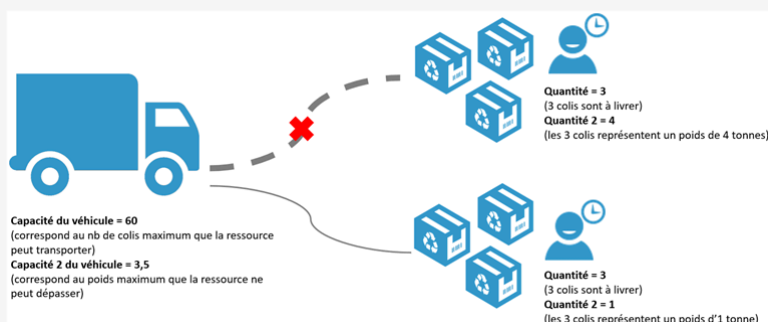
En plus du paramètre Quantité, TourSolver propose 24 autres paramètres de dimensionnement à livrer ou à collecter. Chacun d'entre eux est à mettre en correspondance avec les paramètres Capacité du véhicule à Capacité du véhicule 24 des ressources et permettent de rendre compte de différentes situations.

*Format : numérique*

Les valeurs possibles pour cette contrainte vont de 0 à 2147483. TourSolver ne reconnaît pas plus de 3 chiffres significatifs après la virgule.

Exemple :

Livraisons ou collecte d'objets lourds et volumineux. On utilise Quantité et Capacité du véhicule pour le poids, Quantité 2 et Capacité du véhicule 2 pour le volume. Le chargement des ressources et la livraison ou collecte des clients seront optimisés en fonction de la double contrainte de poids et de volume.



## Durée de chargement/déchargement unitaire (*unloadingDurationPerUnit*)

Cette valeur correspond au temps nécessaire pour livrer une unité. Cette valeur est multipliée par la valeur saisie dans la contrainte Quantité. Cette contrainte s'applique uniquement sur la contrainte Quantité de la visite et non sur les 23 autres quantités possibles.

*Format : HH:MM:SS ou en seconde (20 pour 20 secondes)*

Exemple :

dans le cas d'une livraison de 10 colis chez un client : le temps de déchargement estimé pour un seul colis est de 3 minutes. On saisira la valeur 00:03:00 (ou 180 secondes) pour la contrainte Durée de

chargement/déchargement unitaire et la valeur 10 dans la contrainte Quantité, ainsi la ressource mettra 30 minutes pour la livraison (3min x 10 colis).

### Type de visite (*type*)

Cette contrainte permet d'indiquer si la visite concerne une livraison/dépose ou une reprise/collecte/ramasse.

*Format : binaire*

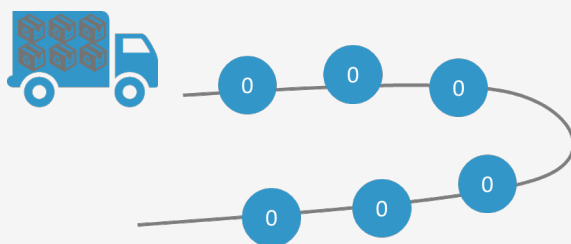
Si la valeur = 0, la tournée est en mode livraison ;

Si la valeur = 1, la tournée est en mode collecte ;

Si la valeur = 2, la tournée est en mode livraison à effectuer avant toute collecte.

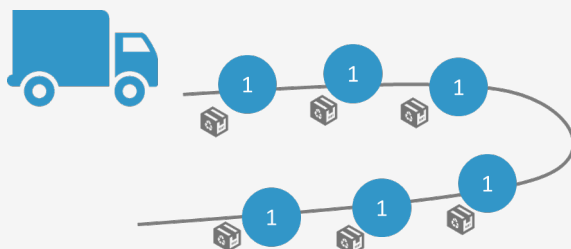
Exemple 1 :

une tournée de livraison (Type de visite = 0)



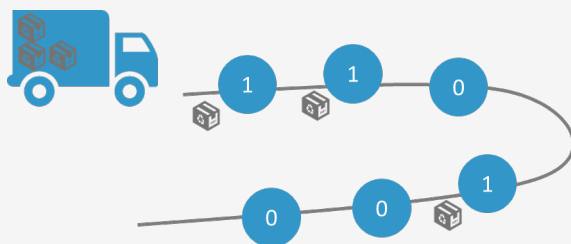
Exemple 2 :

une tournée de collecte (Type de visite = 1)



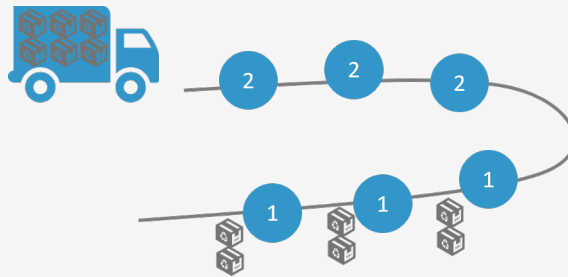
Exemple 3 :

une tournée de livraison (Type de visite = 0) / collecte (Type de visite = 1)



Exemple 4 :

une tournée ordonnée de livraison (Type de visite = 2) / collecte (Type de visite = 1)



Pour une même visite, on pourra effectuer une livraison et une collecte. Il est alors indispensable de dupliquer la ligne visite, modifier le Type de visite et si nécessaire la quantité (à livrer ou à collecter).

Nom	Type	Quantité
Client1	0	20
Client1	1	10

## Durée de la visite (*fixedVisitDuration*)

C'est le temps incompressible à passer sur la visite, c'est-à-dire un temps minimum de visite du client. A ce temps, peut venir s'ajouter le temps saisi dans la contrainte Durée de chargement/déchargement unitaire.

Format : HH:MM:SS

Exemple :

pour une livraison, la Durée de la visite peut correspondre au temps de stationnement, de faire signer un bon de livraison, d'actionner le hayon... à ajouter au temps propre du déchargement renseigné dans la contrainte Durée de chargement/déchargement unitaire.

## Heure d'ouverture 1, Heure de fermeture 1 (*timeWindow[0].beginTime,timeWindow[0].endTime*)

Ces contraintes permettent de définir un créneau horaire dans lequel la visite peut être réalisée. Heure d'ouverture 1 correspond à l'heure à partir de laquelle la visite peut être réalisée. Heure de fermeture 1 correspond à l'heure au-delà de laquelle la visite ne peut plus être réalisée. Cette plage horaire s'appliquera à l'ensemble des jours sauf si Jours d'ouverture 1 est défini. Dans ce cas, la plage horaire est appliquée seulement à cette valeur.

Si une plage horaire n'est pas renseignée pour une visite donnée, la visite peut être réalisée n'importe quand, sur les éventuels Jours d'ouverture définis et dans la limite des horaires de travail de la ressource. On peut définir jusqu'à 4 créneaux horaires différents (et leurs jours associés) pour chaque visite.

*Format : HH:MM:SS*

Exemple 1 :

Heure d'ouverture 1 = 07:30:00 et Heure de fermeture 1 = 11:30:00 définissent un créneau horaire de 4 heures pendant lesquels la ressource peut visiter le client.

Exemple 2 :

Le client doit être livré entre 23:00:00 et 03:00:00 du matin. Dans ce cas, il faut décaler l'ensemble des horaires (dans un sens ou l'autre) afin qu'ils soient tous contenus dans une journée et récupérer l'ordonnancement des visites à faire.

Heure d'ouverture 1 = 19:00:00 et Heure de fermeture 1 = 23:00:00 définissent un créneau horaire de 4 heures pendant lesquels la ressource peut visiter le client.

Post optimisation, il faudra décaler à nouveau l'ensemble des horaires des visites pour retrouver la plage horaire d'origine.

Ce client a été planifié par TourSolver à 21:00:00, dans la réalité il sera visité à 01:00:00 du matin (=23:00:00 + 2 heures).

- ❗ Les créneaux définis à cheval sur plusieurs jours ne sont pas gérés. L'utilisateur ne peut pas utiliser de créneau démarrant à 23:00:00 et finissant à 03:00:00 le lendemain sans décaler l'ensemble des horaires.

- 💡 Une visite ne peut démarrer avant l'heure de début. Par défaut, une visite démarrée dans le créneau horaire peut se poursuivre au-delà de l'heure de fin. Si l'on souhaite qu'une visite se termine avant Heure de fermeture 1, il faut activer Terminer la visite avant l'heure de fermeture.

## Heure d'ouverture 2, Heure de fermeture 2

***(timeWindow[1].beginTime,timeWindow[1].endTime)***

Ces contraintes permettent de définir un deuxième créneau horaire qui s'applique aux jours de visite définis dans Jours d'ouverture 2. Heure d'ouverture 2 correspond à l'heure à partir de laquelle la visite peut être réalisée. Heure de fermeture 2 correspond à l'heure au-delà de laquelle la visite ne peut plus être réalisée.

*Format : HH:MM:SS*

Exemple 1 :

Le client peut être livré le matin entre 09:00:00 et 12:00:00 et l'après-midi entre 14:00:00 et 17:00:00.

Pour cela, on utilisera la contrainte :

Heure d'ouverture 1 = 09:00:00 ;

Heure de fermeture 1 = 12:00:00 ;

Heure d'ouverture 2 = 14:00:00 ;



Heure de fermeture 2 = 17:00:00.

Exemple 2 :

Le client peut être livré le lundi entre 07:00:00 et 17:00:00 et le mercredi entre 09:00:00 et 18:00:00, dans ce cas :

Heure d'ouverture 1 = 07:00:00 ;

Heure de fermeture 1 = 17:00:00 ;

Jours d'ouverture 1 = 1 ;

Heure d'ouverture 2 = 09:00:00 ;

Heure de fermeture 2 = 18:00:00 ;

Jours d'ouverture 2 = 3.

### Heure d'ouverture 3, Heure de fermeture 3 (*timeWindow[2].beginTime,timeWindow[2].endTime*)

Ces contraintes permettent de définir un créneau horaire qui s'applique aux jours de visite définis dans Jours d'ouverture 3. Heure d'ouverture 3 correspond à l'heure à partir de laquelle la visite peut être réalisée. Heure de fermeture 3 correspond à l'heure au-delà de laquelle la visite ne peut plus être réalisée.

*Format : HH:MM:SS*

### Heure d'ouverture 4, Heure de fermeture 4 (*timeWindow[3].beginTime,timeWindow[3].endTime*)

Ces contraintes permettent de définir un créneau horaire qui s'applique aux jours de visite définis dans Jours d'ouverture 4. Heure d'ouverture 4 correspond à l'heure à partir de laquelle la visite peut être réalisée. Heure de fermeture 4 correspond à l'heure au-delà de laquelle la visite ne peut plus être réalisée.

*Format : HH:MM:SS*

### Jours d'ouverture 1 (*possibleVisitDays[0]*)

Cette contrainte contient l'ensemble des Jours d'ouverture de visite (jusqu'à 64) sur lesquels s'applique le créneau horaire défini par les contraintes Heure d'ouverture 1 et Heure de fermeture 1.

*Format : numérique ou date*

Ces jours peuvent être définis un à un (1,4) ou par créneau (1-4), en nombre entier (1,2,...64) ou au format date (14/05/2016 => 17/07/2016).

Exemple 1 :

Jours d'ouverture 1 = 1,2,5 (ou 14/05/2016,15/05/2016,18/05/2016 ) signifie que la visite peut s'effectuer les jours 1, 2 et 5 du planning (ou les 14/05/2016, 15/05/2016 et 18/05/2016) aux horaires définis par les contraintes Heure d'ouverture 1 et Heure de fermeture 1.

Exemple 2 :

Jours d'ouverture 1 = 1-5 (ou 14/05/2016 => 18/05/2016) signifie que la visite peut s'effectuer tous les jours 1, 2, 3, 4 et 5 du planning (ou entre le 14/05/2016 et le 18/05/2016) aux horaires définis par les contraintes Heure d'ouverture 1 et Heure de fermeture 1.



Si l'on veut indiquer qu'une visite doit être réalisée le jour 2 ET le jour 4 de la semaine, il faut créer deux lignes Visites en indiquant 2 comme Jours d'ouverture 1 sur la première ligne et 4 sur la seconde ligne...

Clients	Jours possibles 1
Client A	2
Client A	4

...ou utiliser la contrainte Fréquence de visite.

Clients	Jours possibles 1	Fréquence
Client A	2,4	2/5



Attention, cette contrainte doit être mise en relation avec la contrainte Jours tournée des ressources et utiliser le même format (date ou nombre). Aucune date de Jours d'ouverture 1 ne peut être antérieure à la plus ancienne date de Jours tournée.

## Jours d'ouverture 2 (*possibleVisitDays[1]*)

Cette contrainte contient l'ensemble des Jours d'ouverture de visite (jusqu'à 64) sur lesquels s'applique le créneau horaire défini par les contraintes Heure d'ouverture 2 et Heure de fermeture 2. Ces jours peuvent être définis un à un ou par créneau et suivent la même syntaxe que Jours d'ouverture 1.

Exemple 1 :

Le client peut être livré le lundi entre 07:00:00 et 17:00:00 et le mercredi entre 09:00:00 et 18:00:00 :  
 Heure d'ouverture 1 = 07:00:00, Heure de fermeture 1 = 17:00:00, Jours d'ouverture 1 = 1 et Heure d'ouverture 2 = 09:00:00, Heure de fermeture 2 = 18:00:00, Jours d'ouverture 2 = 3

Exemple 2 :

Le client peut être livré du lundi au vendredi entre 07:00:00 et 17:00:00 et le samedi entre 09:00:00 et 12:00:00 :

Heure d'ouverture 1 = 07:00:00, Heure de fermeture 1 = 17:00:00, Jours d'ouverture 1 = 1-5 et Heure d'ouverture 2 = 09:00:00, Heure de fermeture 2 = 12:00:00, Jours d'ouverture 2 = 6

## Jours d'ouverture 3 (*possibleVisitDays[2]*)

Cette contrainte contient l'ensemble des Jours d'ouverture de visite (jusqu'à 64) sur lesquels s'applique le créneau horaire défini par les contraintes Heure d'ouverture 3 et Heure de fermeture 3. Ces jours peuvent être définis un à un ou par créneau et suivent la même syntaxe que Jours d'ouverture 1.

## Jours d'ouverture 4 (*possibleVisitDays[3]*)

Cette contrainte contient l'ensemble des Jours d'ouverture de visite (jusqu'à 64) sur lesquels s'applique le créneau horaire défini par les contraintes Heure d'ouverture 4 et Heure de fermeture 4. Ces jours peuvent être définis un à un ou par créneau et suivent la même syntaxe que Jours d'ouverture 1.

## Terminer la visite avant l'heure de fermeture (*wholeVisitInTimeWindow*)

Cette contrainte permet de déterminer si les visites peuvent se terminer après la fin de la plage horaire indiquée. Par défaut, une visite démarrée dans le créneau horaire peut se poursuivre au-delà de l'heure de fin. Si l'on souhaite qu'une visite se termine avant l'heure de fermeture 1, il faut activer cette contrainte.

*Format : binaire*

Exemple :



Dans le cas où, Terminer la visite avant l'heure de fermeture = 0



Dans le cas où, Terminer la visite avant l'heure de fermeture = 1



## Tolérance sur le retard (*punctuality*)

C'est une valeur qui permet de donner une importance au retard de visite. Par défaut, TourSolver fait en sorte d'éviter les retards. Rien ne sert de mettre une intolérance sur les retards à toutes les visites, il est préférable de jouer sur les écarts de tolérance.

*Format : numérique (valeur comprise entre 1 et 5.)*

Une tolérance à 1 correspond à une Pénalité par heure de retard nulle, c'est à dire que la visite peut être réalisée avec du retard. En revanche, une tolérance de 2 ou plus impose à TourSolver de ne pas réaliser la visite en retard, dans la mesure du possible. Les visites ayant une tolérance à 5 auront moins de risque d'être réalisées en retard que les visites ayant une tolérance à 2.

La tolérance est respectée par une ressource quand celle-ci arrive à la visite dans le créneau horaire éventuellement défini, et non pas s'il termine la visite dans cet intervalle de temps (sauf si la contrainte Terminer la visite avant l'heure de fermeture est active).

Exemple :

On peut imaginer cette contrainte par un curseur de tolérance.



## Pénalité par heure de retard (*delayPenaltyPerHour*)

C'est un coût supplémentaire appliqué par heure de retard. Cette contrainte permet un contrôle plus fin du respect des plages horaires de visite. Attention, l'utilisation des contraintes liées à la tolérance vont fortement bloquer l'optimisation et faire du retard sa principale « priorité ».

*Format : numérique (nombre entier ou réel)*

Exemple :

J'accepte de faire quelques détours et donc d'avantage de kilomètres pour mieux respecter les horaires de livraison.

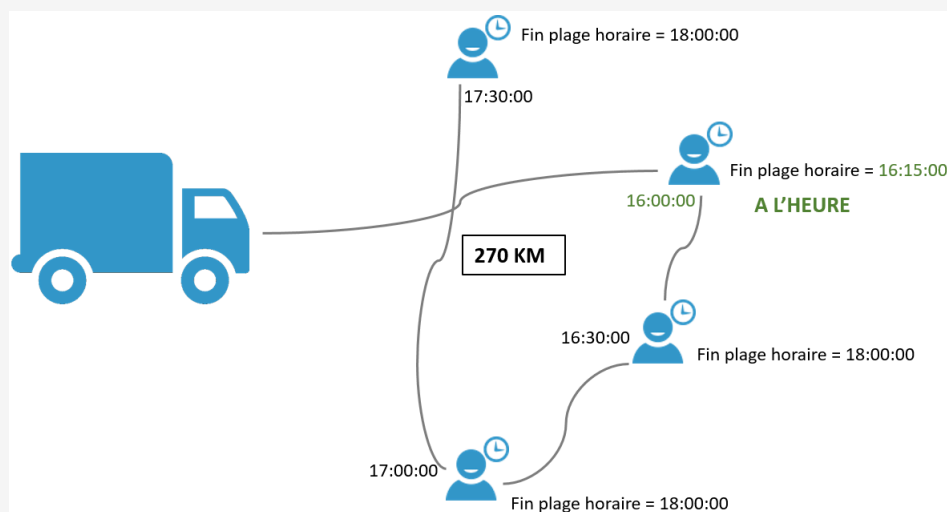
J'accepte de faire 70 km de plus pour éviter 5 minutes de retard chez un client.

Alors  $0,5 \times 70 \times 12 (=60/5\text{min}) = 420$

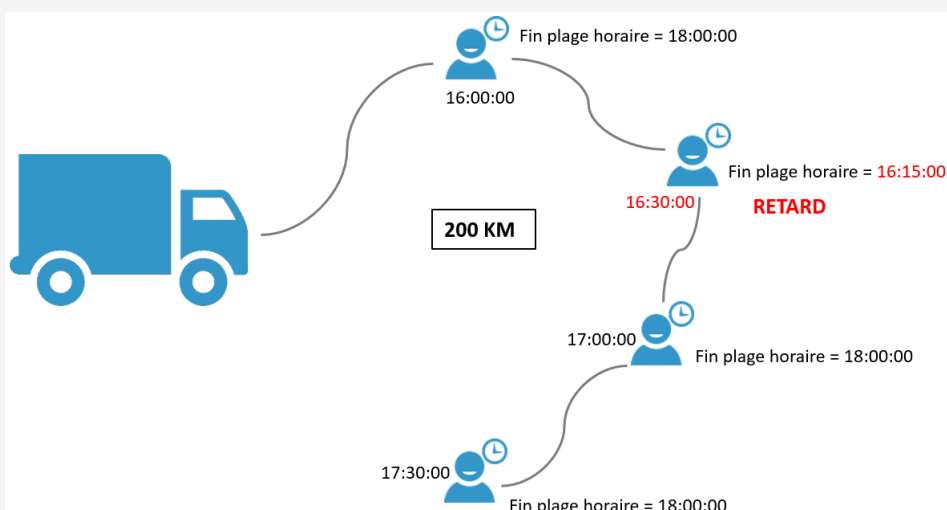
(0,5 exprimant la contrainte des ressources Coût kilométrique)

La valeur à inscrire dans la contrainte Pénalité par heure de retard est 420.

## Schéma de tournée en utilisant la contrainte Pénalité par heure de retard



## Schéma de tournée en n'utilisant pas la contrainte Pénalité par heure de retard

Fréquence de visite (*frequency*)

Cette contrainte permet de spécifier qu'une visite doit être réalisée plusieurs fois sur une période de plusieurs jours. TourSolver tente d'optimiser en équilibrant au mieux les écarts entre les visites en tenant compte également des autres contraintes. La fréquence ne garantit pas le respect exact des périodes. Un indicateur de non-respect des périodes de fréquence permet dans l'interface TourSolver de connaître le nombre de visites n'étant pas en période optimale de fréquentation.

Format : caractère (1/5)

- ❗ Si cette contrainte est utilisée, la contrainte Jours travaillés des ressources doit être renseignée.

Exemple 1 :

Fréquence de visite = 3/5 signifie qu'un client doit être visité 3 fois en 5 jours. TourSolver cherche à respecter un écart de 2 jours entre chaque visite.

Exemple 2 :

Fréquence de visite = 3/5\* signifie qu'un client doit être visité 3 fois en 5 jours. TourSolver cherche à réaliser les 3 visites d'un même client par une ressource unique. Dans ce cas, le respect de l'écart entre 2 visites devient secondaire.

Exemple 3 :

Fréquence de visite = 1/5 signifie qu'un client doit être visité 1 fois en 5 jours. Si la contrainte des ressources Jours travaillés = 1-15, le client sera visité une fois entre le jour 1 à 5, une deuxième fois entre le jour 6 à 10 et une troisième fois entre le jour 11 à 15. La contrainte Espacement minimal entre visites permet de respecter un écart strict entre les visites.

### Date de dernière visite (*tsOrderLastVisit*)

Cette contrainte permet de prendre en compte la date de la dernière visite effectuée qui précède la tournée à optimiser et ainsi respecter au mieux l'écart entre les visites.

*Format : entier négatif*

- ❗ Si le nombre de récurrences de la fréquence est supérieur à un, la contrainte n'est pas prise en compte.

Exemple 1 :

Fréquence de visites = 1/5

Espacement minimal entre visites = 2

Date de dernière visite = -1

Le client sera visité qu'à partir du jour 2, car si la visite a lieu le jour 1, l'écart entre les visites ne respecte pas l'espacement minimal de 2 jours.

### Espacement minimal entre visites (*tsOrderMinimumSpacing*)

Cette contrainte permet de respecter un écart minimum entre 2 visites chez un même client.

*Format : entier*

- ❗ Si le nombre de récurrences de la fréquence est supérieur à un, la contrainte n'est pas prise en compte.

Exemple 1 :

Fréquence de visites = 1/5

Espacement minimal entre visites = 3

Jours travaillés = 1-15

Pour respecter un écart de 2 jours entre 2 visites, le client ne pourra pas être visité les jours 5,8,11 mais plutôt les jours 5,9,12.

## Durée d'une portion de visite découpée (*minDuration*)

Cette contrainte correspond à la durée minimum autorisant la découpe d'une visite en plusieurs passages. Le découpage peut être fait d'un jour au jour suivant ou au moment de la pause déjeuner. Cette valeur est intimement liée à la contrainte Durée de la visite.

Format : HH:MM:SS

Exemple :

Soit une visite A dont la contrainte Durée de la visite est de 4 heures et une visite B égale à 30 minutes. Durée d'une portion de visite découpée est égale à 01:00:00 alors tous les clients dont la contrainte Durée de la visite est supérieur à 01:00:00 pourra être découpée en plusieurs visites. La visite A pourra être découpée si besoin mais pas la visite B. Le temps minimum de découpe de chacune de ces visites est défini par la contrainte Durée-seuil de découpe d'une visite.

! Une visite ne peut être découpée en plus de 16 blocs.

## Durée-seuil de découpe d'une visite (*minPartDuration*)

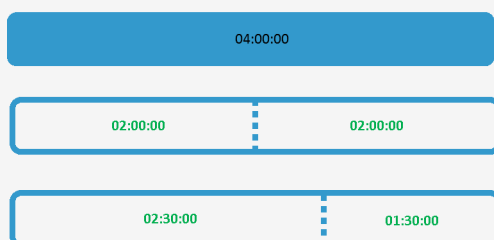
Cette valeur correspond à la durée minimum en dessous de laquelle une visite ne peut plus être découpée.

Format : HH:MM:SS

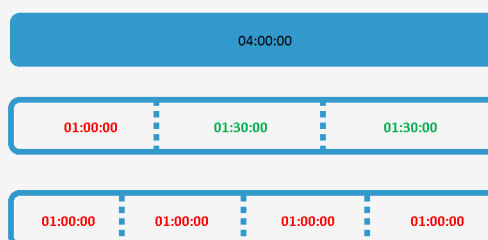
Exemple :

Pour un client dont la Durée de la visite est égale à 04:00:00. La contrainte Durée d'une portion de visite découpée est égale à 01:00:00, ce qui signifie que la visite peut être découpée. Durée-seuil de découpe d'une visite est égal à 01:30:00, ce qui signifie qu'après découpage, aucune visite ne pourra faire moins de 1h30.

TourSolver peut proposer le découpage suivant :



TourSolver ne pourra pas proposer le découpage suivant :

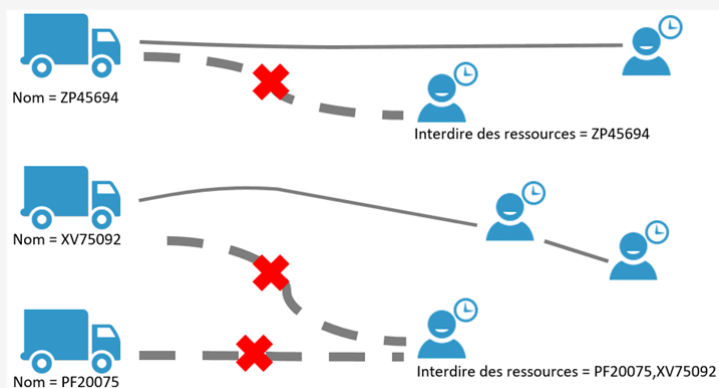


## Exclure des ressources (*excludeResources*)

Cette contrainte permet à l'utilisateur de spécifier, pour une visite, la liste des ressources qui ne peuvent pas la réaliser. La valeur à inscrire dans cette contrainte doit être égale à la valeur renseignée dans la contrainte Nom des ressources.

*Format : caractère, sous forme de liste séparée par une virgule si les ressources sont multiples.*

Exemple :

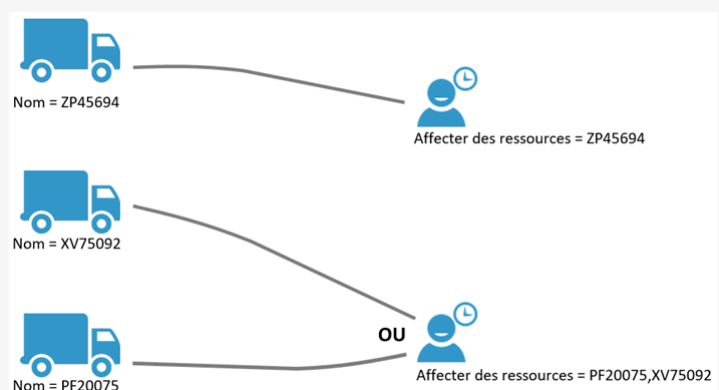


## Affecter des ressources (*assignResources*)

Cette contrainte permet à l'utilisateur de spécifier, pour une visite, la liste des ressources qui peuvent la réaliser. La valeur à inscrire dans cette contrainte doit être égale à la valeur renseignée dans la contrainte Nom des ressources.

*Format : caractère, sous forme de liste séparée par une virgule si les ressources sont multiples.*

Exemple :



## Coût de sous-traitance (*courierPenalty*)

Cette contrainte permet à l'utilisateur de spécifier qu'une commande peut être livrée par un service de messagerie externe. Le moteur d'optimisation calcule s'il est plus intéressant d'envoyer la commande



par ce service ou de l'intégrer à une tournée. Ces visites n'apparaîtront nulle part dans l'optimisation (tournées, planning...). Le coût de sous-traitance est intégré au coût total de l'optimisation.

*Format : numérique (nombre entier ou réel)*

Exemple :

1000 client A et 100 client B

## Nom de la ressource (*evaluationInfos.orderOriginalResourceId*)

Cette valeur s'utilise en mode évaluation. Elle permet d'affecter une visite à une tournée particulière. Elle est particulièrement utile pour évaluer des tournées existantes.

*Format : caractère*

## Jour de visite (*evaluationInfos.orderOriginalVisitDay*)

Cette valeur s'utilise en mode évaluation. Lors de tournées multi-jours, cette contrainte permet de préciser sur la visite un jour de tournée particulier. La valeur de cette contrainte doit être identique à l'une des valeurs de la contrainte Jours tournée définis pour la ressource spécifiée dans Nom de la ressource et également, si défini, à l'un des Jours d'ouverture pour cette visite.

*Format : caractère*

## Numéro d'ordre (*evaluationInfos.orderPosition*)

Cette valeur s'utilise en mode évaluation. Cette contrainte permet de préciser le rang de la visite dans cette tournée. Toutes les visites doivent être renseignées pour que cette contrainte s'applique.

*Format : numérique*

## Autres données (*customDataMap*)

Cette contrainte ne joue aucun rôle dans l'optimisation, elle permet de sélectionner d'autres données descriptives sur les visites, qui seront consultables dans les résultats (commentaires, numéro de téléphone, ...). Il suffit de sélectionner les colonnes et elles apparaîtront dans les différents résultats générés par TourSolver (planning, feuilles de routes, etc.). Un maximum de 50 colonnes peut être conservé.

## Contraintes des ressources

- ❗ Dans cette documentation sont employés les mots-clés de la terminologie par défaut définie dans la page *Mon activité* de l'interface.  
Ainsi, on parlera ici de "ressource" mais les noms des contraintes peuvent évoluer en fonction de vos propres choix de terminologie.

- ❗ Après chaque nom de contrainte, les parenthèses contiennent le nom équivalent de cette contrainte dans [TourSolver Cloud API](https://geoservices.geoconcept.com/ToursolverCloud/api-book.html) [https://geoservices.geoconcept.com/ToursolverCloud/api-book.html].

### Nom (*id*)

Cette valeur permet d'identifier la ressource. Le nom peut être un patronyme, un numéro, un identifiant, une immatriculation...

*Format : caractère (maximum 128 caractères)*

Exemple :

Mr Dupont / ZP65294 / Peugeot607

Cette contrainte doit être obligatoirement renseignée.

Cette valeur doit être unique.

### Nom des départs et arrivées

Cette valeur permet d'identifier les départs et arrivées. Le nom peut être un patronyme, un numéro, un identifiant, une ville...

*Format : caractère*

Les valeurs par défaut dans les résultats sont : *Départ* et *Retour à la base*.

Exemple :

Départ Bagneux / Agence Grenoble

### Heures de travail (*workStartTime,workEndTime*)

Ces contraintes permettent de définir un créneau horaire dans lequel la ressource peut travailler. L'heure à laquelle la ressource commence et finit de travailler.

Cette plage horaire s'appliquera à l'ensemble des jours sauf si Jours travaillés est défini. Dans ce cas, la plage horaire est appliquée seulement à cette valeur.

On peut définir jusqu'à 4 créneaux horaires différents (et leurs jours associés) pour chaque ressource.

Format : HH:MM

Exemple 1 :

07:30 à 11:30 signifie que la ressource commence à travailler à 07:30 et rentrera au plus tard à 11:30.

Exemple 2 :

La ressource travaille entre 23:00 et 03:00 du matin. Dans ce cas, il faut décaler l'ensemble des horaires (dans un sens ou l'autre) afin qu'ils soient tous contenus dans une journée et récupérer l'ordonnancement des visites à faire. 19:00 à 23:00 définissent un créneau horaire de 4 heures pendant lesquels la ressource peut travailler. Post optimisation, il faudra décaler à nouveau l'ensemble des horaires des visites pour retrouver la plage horaire d'origine. Si le client a été planifié par TourSolver à 21:00, dans la réalité il sera visité à 01:00 du matin (=23:00 + 2 heures).

- ❗ Les créneaux définis à cheval sur plusieurs jours ne sont pas gérés. L'utilisateur ne peut pas utiliser de créneau démarrant à 23:00 et finissant à 03:00 le lendemain sans décaler l'ensemble des horaires.

## Pause déjeuner

Ces contraintes permettent de définir une pause déjeuner.

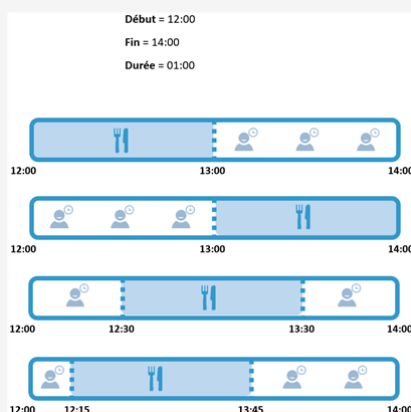
Format : HH:MM

Début (*lunch.start*) : Cette contrainte indique l'heure à partir de laquelle la pause déjeuner de la ressource peut commencer.

Fin (*lunch.end*) : Cette contrainte indique l'heure maximale à laquelle la pause déjeuner de la ressource peut être terminée. Si cette contrainte est utilisée, la pause déjeuner définie dans Durée doit ainsi être terminée pendant le créneau horaire défini par Début et Fin.

Durée (*lunch.duration*) : Cette contrainte indique la durée consacrée au temps de déjeuner de la ressource. Cette durée n'est pas considérée par TourSolver comme du temps de travail mais comme un temps de pause. Aussi est-il important de noter qu'elle est déduite de l'amplitude de la journée de travail.

Exemple 1 : cas possibles trouvés par TourSolver





En fonction des coûts, TourSolver calculera s'il est moins coûteux de faire travailler une autre ressource ou de consommer des heures supplémentaires pour la même ressource.

## Coût horaire (*workPenalty*)

Cette contrainte permet d'indiquer le coût d'une heure de travail assurée par une ressource. Le Coût horaire ET le Coût kilométrique ne doivent pas être simultanément nuls.

*Format : numérique*

Exemple :

Pour une ressource dont les contraintes sont les suivantes :

Coût horaire = 20

Heures de travail = 08:00 et 18:00

La tournée sortie par TourSolver fait terminer la ressource à 16:00 soit 8h de travail (16-8), le coût total du travail s'élève à 160 (20\*8).

## Coût kilométrique (*travelPenalty*)

Cette contrainte permet d'indiquer le coût par unité de trajet (km, miles...) parcourue par la ressource.

*Format : numérique*

Exemple :

Soit le coût d'1km parcouru coûte 0,7, alors Coût kilométrique = 0.7

## Pondération de la vitesse (*speedAdjustment*)

Si les temps de parcours observés vous paraissent trop faibles ou trop élevés, l'utilisateur peut les ajuster en spécifiant, pour chaque ressource, à quel pourcentage de la vitesse définie dans les statistiques routières elle circule réellement. Ainsi l'utilisateur peut adapter les données de circulation aux contraintes des ressources tout en gardant les informations de ralentissement de la vitesse en fonction de l'heure et du sens de circulation.

*Format : numérique (exprimé en %)*

100% étant la valeur par défaut et de référence de la vitesse définie dans les profils de véhicule par les statistiques routières.

Exemple 1 :

Les statistiques indiquent que la vitesse sur l'Avenue Aristide Briand à Bagneux est de 50km/h. L'utilisateur juge ce temps trop rapide, il renseigne la contrainte Pondération de la vitesse = 80, la ressource roulera 20% moins vite, soit à 40km/h au lieu de 50km/h.

Exemple 2 :

Les statistiques indiquent que la vitesse sur l'Avenue Aristide Briand à Bagneux est de 50km/h. L'utilisateur juge ce temps trop faible, il renseigne la contrainte Pondération de la vitesse = 120, la ressource roulera 20% plus vite, soit à 60km/h au lieu de 50km/h.

### Capacité du véhicule (*capacity[0]*)

Cette valeur représente la capacité maximum qu'une ressource peut livrer ou collecter. L'utilisateur détermine l'unité de la quantité (kilos, m3, litres...) et assure la cohérence avec la contrainte Quantité des visites.

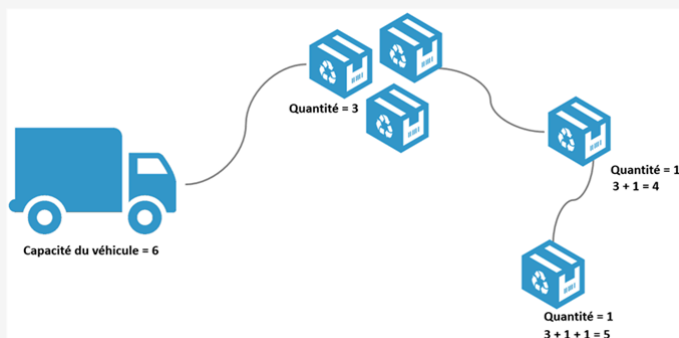
*Format : numérique*

Exemple :

Dans le cas de livraison de colis, on indique le nombre maximum de colis que la ressource peut contenir (6 pour 6 colis).

Dans le cas de matière liquide, on indique le nombre de litres (3000 pour 3000 litres).

Dans le cas de gros conditionnements, on indique le volume (2 pour 2m3).



## Véhicule

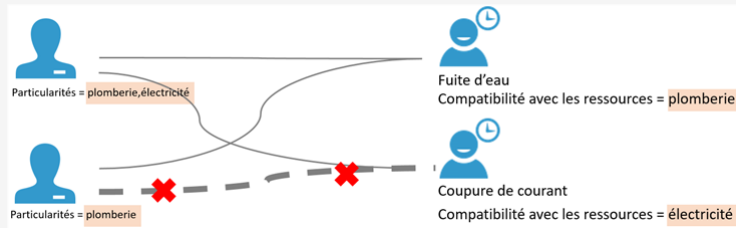
### Particularités (*providedSkills*)

Cette contrainte désigne les critères requis par la visite que la ressource doit suivre pour pouvoir réaliser celle-ci.

*Format : caractère (exprimés sous la forme d'une liste de mots séparés par une virgule).*

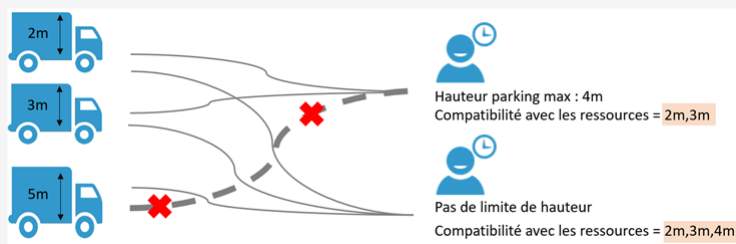
Exemple 1 :

Pour un client ayant besoin d'une intervention de maintenance concernant la plomberie, on pourra indiquer "plomberie". Il faudra qu'au moins une des ressources dispose de cette particularité pour que le client soit visité.



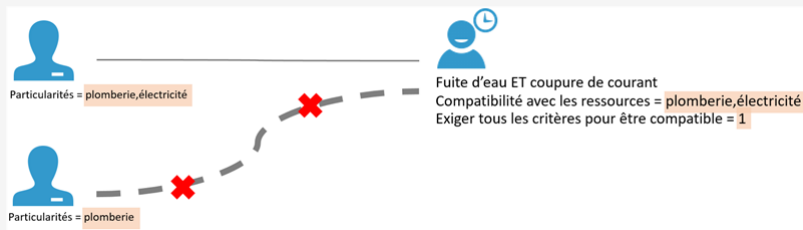
Exemple 2 :

Pour un client dont l'accès est limité à une hauteur de 4 m, on indiquera "2m,3m" parce qu'il ne peut être livré que par un véhicule inférieur à 4m . C'est une des ressources avec la particularité 2m ou 3m qui pourra le visiter.

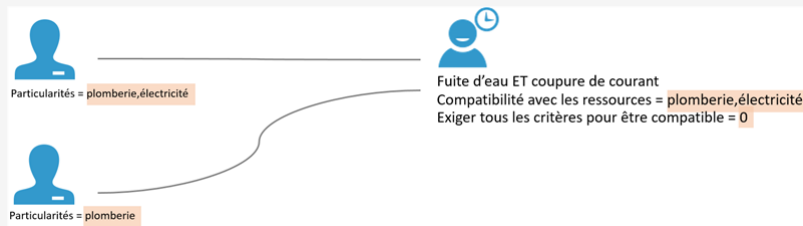


Exemple 3 :

Pour un client ayant besoin d'une intervention de maintenance concernant la plomberie ET l'électricité, on pourra indiquer "plomberie,électricité". Il faudra qu'au moins une des ressources dispose de ces 2 particularités pour que le client soit visité. Il faut donc utiliser la contrainte Exiger tous les critères pour être compatible = 1.



Si la contrainte Exiger tous les critères pour être compatible = 0 alors la prise en compte des 2 particularités ne sera pas assurée.



## Planning

### Disponibilité de la ressource (*available*)

Cette contrainte permet de prendre en considération les ressources mises à disposition sans avoir à ajouter ou enlever des données dans le jeu de données complet.

Si le bouton est sur la valeur Disponible, la ressource peut effectuer sa tournée.

Si le bouton est sur la valeur Indisponible, la ressource ne peut pas effectuer sa tournée.

Par défaut la ressource est disponible.

### Horaires journaliers 2 à 4 (*otherWorkStartTime[0]...[2],otherWorkEndTime[0]...[2]*)

Ces contraintes permettent de définir d'autres créneaux horaires qui s'appliquent aux jours de travail définis dans Jours travaillés 2 à 4.

*Format : HH:MM*

Exemple 1 :

La ressource peut travailler du lundi au jeudi de 09:00 à 17:00 et le vendredi de 09:00 à 12:00. Pour cela, on utilisera la contrainte :

Horaires journaliers 1 = 09:00 et 17:00 ;

Jours travaillés 1 = 1-4 ;

Horaires journaliers 2 = 09:00 et 12:00 ;

Jours travaillés 2 = 5.

Exemple 2 :

La ressource peut travailler le lundi entre 07:00 et 17:00 et le mercredi entre 09:00 et 18:00, dans ce cas :

Horaires journaliers 1 = 07:00 et 17:00 ;

Jours travaillés 1 = 1 ;

Horaires journaliers 2 = 09:00 et 18:00 ;

Jours travaillés 2 = 3.



Pour un jour donné, seul un créneau horaire peut être défini. Il n'est donc pas possible de paramétrer cet exemple : "la ressource peut travailler le matin entre 09:00 et 12:00 et l'après-midi entre 14:00 et 17:00 sans pause déjeuner".

### Jours travaillés 1 à 4 (*workingDays,otherWorkDays[0]...[2]*)

Cette contrainte contient l'ensemble des Jours de travail (jusqu'à 64 jours) sur lesquels s'applique le créneau horaire défini par les contraintes Horaires journaliers 1 à 4.

*Format : caractère*



Ces jours peuvent être définis un à un (1,4) ou par créneau (1-4), en nombre entier (1,2,...64) ou au format date (14/05/2016 => 17/07/2016).

Exemple 1 :

Jours travaillés 1 = 1,2,5 (ou 14/05/2016,15/05/2016,18/05/2016 ) signifie que la ressource peut faire des visites les jours 1, 2 et 5 du planning (ou les 14/05/2016, 15/05/2016 et 18/05/2016) aux horaires définis par les contraintes Horaires journaliers 1.

Jours travaillés 1 = 1-5 (ou 14/05/2016 => 18/05/2016) signifie que la ressource peut faire des visites tous les jours 1, 2, 3, 4 et 5 du planning (ou entre le 14/05/2016 et le 18/05/2016) aux horaires définis par les contraintes Horaires journaliers 1.

Exemple 2 :

La ressource peut travailler le lundi entre 07:00 et 17:00 et le mercredi entre 09:00 et 18:00 :

Horaires journaliers 1 = 07:00 et 17:00, Jours travaillés 1 = 1 et

Horaires journaliers 2 = 09:00 et 18:00, Jours travaillés 2 = 3

Exemple 3 :

La ressource peut travailler du lundi au vendredi entre 07:00 et 17:00 et le samedi entre 09:00 et 12:00 :

Horaires journaliers 1 = 07:00 et 17:00, Jours travaillés 1 = 1-5 et

Horaires journaliers 2 = 09:00 et 12:00, Jours travaillés 2 = 6

- ⓘ Attention, cette contrainte doit être mise en relation avec les contraintes Jours d'ouverture des visites et utiliser le même format (date ou nombre). Aucune date de Jours d'ouverture ne peut être antérieure à la plus ancienne date de Jours travaillés.

## Heure de départ automatique (*optimumStartTime*)

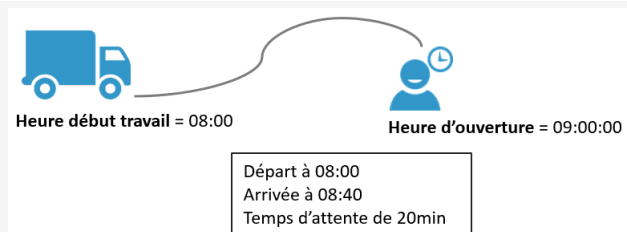
Cette contrainte permet d'ajuster l'heure de départ de la ressource en tenant compte des horaires des visites.

Si le bouton est sur la valeur Désactivé, la ressource démarre sa tournée à l'heure indiquée par sa contrainte Horaires journaliers quelle que soit l'heure d'ouverture de sa première visite (attente possible);  
Si le bouton est sur la valeur Activé, la ressource adapte son heure de départ pour optimiser la tournée et réduire les éventuels temps d'attente s'il y en a en fonction de l'heure d'ouverture de sa première visite.

Dans tous les cas, il ne peut pas partir avant l'heure indiquée par sa contrainte Horaires journaliers. L'heure de fin de travail est décalée d'autant, sauf si la contrainte Durée quotidienne de travail est également utilisée.

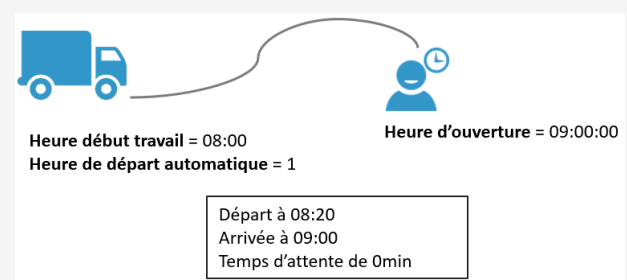
Exemple 1 :

Si l'heure de départ automatique est sur désactivé



Exemple 2 :

Si l'heure de départ automatique est sur activé



## Durée quotidienne de travail (*dailyWorkTime*)

Cette contrainte indique la durée maximum d'heures de travail que la ressource peut effectuer chaque jour, hors éventuelles heures supplémentaires si celles-ci sont utilisées.

Format : *HH:MM*

## Durée hebdomadaire de travail (*weeklyWorkTime*)

Cette contrainte indique la durée maximum d'heures de travail que la ressource peut effectuer chaque semaine. Ce quota hebdomadaire doit alors s'inscrire au sein des plages horaires de travail de la ressource : aucune heure supplémentaire par rapport à cette durée n'est possible.

Format : *HH:MM*

Il est à noter que la notion de semaine utilisée est : en cas d'utilisation d'un format de dates dans la contrainte Jours travaillés, celle d'une semaine calendaire ; à défaut, celle définie par le planificateur (intervalle de 7 jours).

## Heures supplémentaires 2

Nombre (*overtimeDuration[1]*)

Cette contrainte indique la durée d'heures supplémentaires autorisées. Elle indique la durée de la seconde tranche d'heures supplémentaires quotidiennes. Lorsque cette valeur est indiquée, la ressource ne peut travailler au-delà de la somme du temps de travail et du temps d'heures supplémentaires.

Format : *HH:MM*

Exemple :

Dans le cas d'une ressource dont Horaires journaliers = 09:00 et 16:00 et Nombre d'heures supplémentaires 1 = 02:00, si Nombre d'heures supplémentaires 2 = 01:00 alors l'amplitude de travail de la ressource est de 10h ( $=16-9+2+1$ ).



Surcoût (*overtimePenalty[1]*)

Cette contrainte permet d'indiquer le surcoût des heures travaillées pendant la seconde tranche d'heures supplémentaires quotidiennes renseignées dans la contrainte Nombre.

Format : numérique

Exemple : Si le coût horaire est 20 et le coût de l'heure supplémentaire est 30, alors Surcoût d'heures supplémentaires 2 = 10 ( $30-20$ ).

Pour une ressource dont les contraintes sont les suivantes :

Coût horaire = 20

Horaires journaliers = 08:00 et 18:00

Nombre d'heures supplémentaires 1 = 01:00

Surcoût d'heures supplémentaires 1 = 5

Nombre d'heures supplémentaires 2 = 01:00

Surcoût d'heures supplémentaires 2 = 10

TourSolver peut faire travailler la ressource de 08:00 à 20:00 puisque deux heures supplémentaire sont autorisées, et calculer un coût du travail à 255 ( $20*10+25*1+30*1$ ).

En fonction des coûts, TourSolver calculera s'il est moins coûteux de faire travailler une autre ressource ou de consommer des heures supplémentaires pour la même ressource.

## Durée de briefing (*briefingDuration*)

Cette contrainte indique une durée fixe à ajouter avant le départ de la ressource pour sa tournée. Ce temps s'applique au lieu de départ de la ressource.

Format : HH:MM:SS

Exemple :

Avant de partir en tournée, un chauffeur doit laver son véhicule, un technicien doit se changer au vestiaire, un livreur doit récupérer les bons de livraison associés à sa tournée...

## Durée de débriefing (*debriefingDuration*)

Cette contrainte indique une durée fixe à ajouter après le retour de la ressource de sa tournée. Ce temps s'applique au lieu d'arrivée de la ressource.

Format : HH:MM:SS

Exemple :

En rentrant de tournée, un chauffeur doit laver son véhicule, un technicien doit se changer au vestiaire, un livreur doit déposer les bons de livraison signés...

## Tournée

### Départ à la première visite (*openStart*)

Cette contrainte permet d'indiquer si la ressource commence sa tournée à la première visite ou à l'adresse indiquée de sa localisation.

Si le bouton est sur la valeur Désactivé, le coût de la tournée est calculé à partir de la localisation de la ressource.

Si le bouton est sur la valeur Activé, le coût de la tournée est calculé à partir de la première visite.

La valeur par défaut est Désactivé.

Dans le cas où la valeur de la contrainte Première étape au dépôt est Activé, le coût calculé démarre au site où s'effectue le chargement.

### Arrivée à la dernière visite (*openStop*)

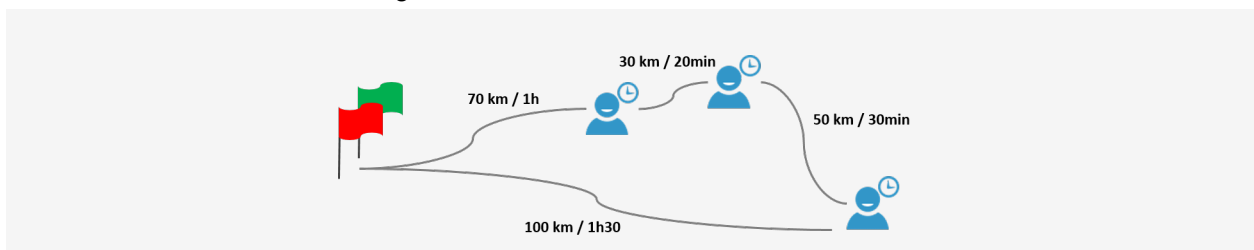
Cette contrainte permet d'indiquer si la ressource termine sa tournée à la dernière visite ou à l'adresse indiquée de sa localisation.

Si le bouton est sur la valeur Désactivé, le coût de la tournée est calculé jusqu'au retour à la localisation de la ressource.

Si le bouton est sur la valeur Activé, le coût de la tournée est calculé jusqu'au retour à la dernière visite.

La valeur par défaut est Désactivé.

Dans le cas où la valeur de la contrainte Dernière étape au dépôt est Activé, le coût calculé s'arrête au site où s'effectue le dernier chargement.



Exemple 1 : Départ à la première visite = Désactivé et Arrivée à la dernière visite = Désactivé

Cette tournée aura un temps de trajet de 3h20 (1h+20min+30min+1h30) et une distance de 250km (70+30+50+100).

Exemple 2 : Départ à la première visite = Activé et Arrivée à la dernière visite = Activé

Cette tournée aura un temps de trajet de 50 minutes (20min+30min) et une distance de 80km (30+50).

Il est possible de dissocier les contraintes départ et arrivée.

## Mesurer la distance à partir de la première visite

Cette contrainte permet de prendre en compte ou non dans l'optimisation de la tournée la distance entre le début de la tournée et la première visite (localisation de la ressource ou point de départ s'il est renseigné).

*Format : binaire*

Si le bouton est sur la valeur Désactivé, le coût kilométrique de la tournée est calculé à partir de la localisation de la ressource.

Si le bouton est sur la valeur Activé, le coût kilométrique de la tournée est calculé à partir de la première visite.

La valeur par défaut est Désactivé.

- ❗ Dans le cas où la valeur de la contrainte Première étape au dépôt est Activé, le coût kilométrique calculé démarre au site où s'effectue le chargement.  
Dans le cas où la contrainte Durée de briefing est renseignée, le coût kilométrique commence à l'arrêt qui suit le briefing.  
Dans le cas où la valeur de la contrainte Départ à la première visite est Activé, cette contrainte est désactivée.

## Arrêter la distance à la dernière visite

Cette contrainte permet de prendre en compte ou non dans l'optimisation de la tournée la distance entre la dernière visite et la fin de la tournée (localisation de la ressource ou point d'arrivée s'il est renseigné).

*Format : binaire*

Si le bouton est sur la valeur Désactivé, le coût kilométrique de la tournée est calculé jusqu'au dernier point d'arrêt de la ressource.

Si le bouton est sur la valeur Activé, le coût kilométrique de la tournée est calculé jusqu'à la dernière visite.

La valeur par défaut est Désactivé.

- ❗ Dans le cas où la valeur de la contrainte Dernière étape au dépôt est Activé, le coût kilométrique calculé s'arrête au site où s'effectue le rechargement.

Dans le cas où la contrainte Durée de débriefing est renseignée, le coût kilométrique s'arrête à l'arrêt précédent le débriefing.

Dans le cas où la valeur de la contrainte Arrivée à la dernière visite est Activé, cette contrainte est désactivée.

## Mesurer le temps à partir de la première visite

Cette contrainte permet de prendre en compte ou non dans l'optimisation de la tournée le temps entre le début de la tournée et la première visite (localisation de la ressource ou point de départ s'il est renseigné).

*Format : binaire*

Si le bouton est sur la valeur Désactivé, le coût horaire de la tournée est calculé à partir de la localisation de la ressource.

Si le bouton est sur la valeur Activé, le coût horaire de la tournée est calculé à partir de la première visite. La valeur par défaut est Désactivé.

- ❗ Dans le cas où la valeur de la contrainte Première étape au dépôt est Activé, le coût horaire calculé démarre au site où s'effectue le chargement.  
Dans le cas où la contrainte Durée de briefing est renseignée, le coût horaire commence à l'arrêt qui suit le briefing.  
Dans le cas où la valeur de la contrainte Départ à la première visite est Activé, cette contrainte est désactivée.

## Arrêter le temps à la dernière visite

Cette contrainte permet de prendre en compte ou non dans l'optimisation de la tournée le temps entre la dernière visite et la fin de la tournée (localisation de la ressource ou point d'arrivée s'il est renseigné).

*Format : binaire*

Si le bouton est sur la valeur Désactivé, le coût horaire de la tournée est calculé jusqu'au dernier point d'arrêt de la ressource.

Si le bouton est sur la valeur Activé, le coût horaire de la tournée est calculé jusqu'à la dernière visite. La valeur par défaut est Désactivé.

- ❗ Dans le cas où la valeur de la contrainte Dernière étape au dépôt est Activé, le coût horaire calculé s'arrête au site où s'effectue le rechargement.  
Dans le cas où la contrainte Durée de débriefing est renseignée, le coût horaire s'arrête à l'arrêt précédent le débriefing.  
Dans le cas où la valeur de la contrainte Arrivée à la dernière visite est Activé, cette contrainte est désactivée.



Exemple 1 : Début et fin tournée temps = Désactivé et Début et fin tournée distance = Activé

Cette tournée aura un temps de trajet de 3h20 (1h+20min+30min+1h30) et une distance de 80km (30+50).

Exemple 2 : Début et fin tournée temps = Activé et Début et fin tournée distance = Désactivé

Cette tournée aura un temps de trajet de 50 minutes (20min+30min) et une distance de 250km (70+30+50+100).

### Distance quotidienne maximale (*maximumDistance*)

Cette contrainte permet de définir la distance maximale qu'une ressource peut parcourir dans une journée de travail.

*Format : numérique*

### Maximum de visites par jour (*maximumVisits*)

Cette contrainte désigne le nombre maximum de visites qu'une ressource peut réaliser dans une journée de travail.

*Format : numérique*

## Chargement / Déchargement

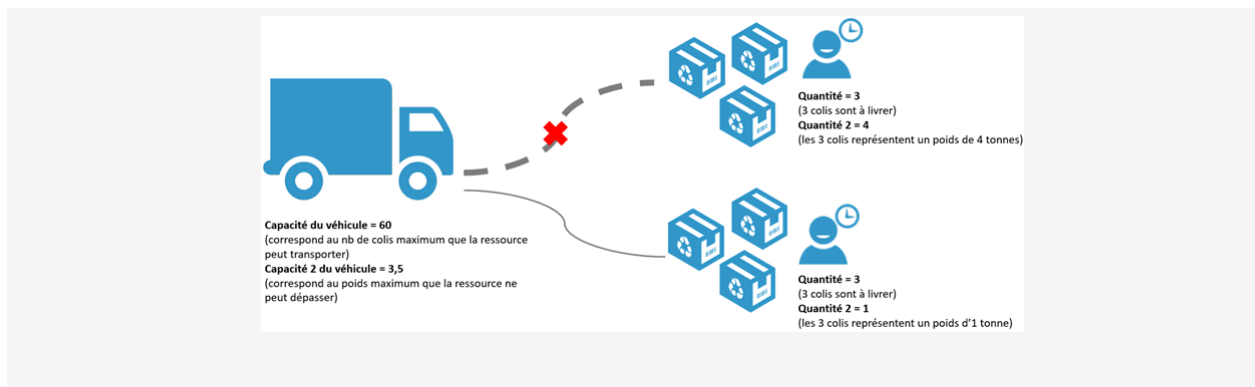
### Capacité du véhicule 2 à 24 (*capacity[1],...capacity[23]*)

En plus de la contrainte Capacité du véhicule, TourSolver propose 24 autres contraintes de dimensionnement à livrer ou à collecter. Chacune d'entre elles est à mettre en correspondance avec les contraintes Visites Quantité, Quantité 2, ..., Quantité 24 et permettent de tenir compte des différentes unités de livraison.

*Format : numérique*

Exemple :

Livraisons ou collecte d'objets lourds et volumineux. On utilise Quantité et Capacité du véhicule pour le poids, Quantité 2 et Capacité du véhicule 2 pour le volume. Le chargement des ressources et la livraison ou collecte des clients seront optimisés en fonction de la double contrainte de poids et de volume.



## Capacité globale (*globalCapacity*)

Cette valeur représente la capacité totale qu'une ressource peut livrer ou collecter lorsque plusieurs quantités sont utilisées sur les visites et sites.

*Format : numérique*

Exemple :

Dans le cas de livraison de légumes, on indique pour la ressource :

- Capacité 1 du véhicule = 150 (kg de carottes)
- Capacité 2 du véhicule = 50 (kg de citrouilles)
- Capacité 3 du véhicule = 100 (kg de pommes de terre)

Soit un total de 300 kg de légumes.

et Capacité globale = 200

Dans ce cas, la ressource peut transporter un maximum de 200 kg tous légumes confondus.

## Utiliser toutes les capacités (*useAllCapacities*)

Si le bouton est sur la valeur Désactivé, seule la Capacité globale est prise en compte.

Si le bouton est sur la valeur Activé, toutes les capacités ainsi que la Capacité globale seront prises en compte.

*Format : binaire*

Exemple :

Dans le cas de livraison de légumes, on indique pour la ressource :

- Capacité 1 du véhicule = 150 (kg de carottes)
- Capacité 2 du véhicule = 50 (kg de citrouilles)
- Capacité 3 du véhicule = 100 (kg de pommes de terre)



- Capacité globale = 200 (kg)

Si Utiliser toutes les capacités est sur la valeur Désactivé, seule la valeur 200 de la Capacité globale ne doit pas être dépassée et les autres capacités ne sont pas prises en compte. La ressource pourra transporter 30 kg de carottes, 60 kg de citrouilles et 110 kg de pommes de terre.

Si Utiliser toutes les capacités est sur la valeur Activé, toutes les capacités devront être respectées. La ressource pourra transporter 50 kg de carottes, 50 kg de citrouilles et 100 kg de pommes de terre.

### Quantité minimale de livraison (*minimumQuantity*)

Cette contrainte indique la quantité minimum que la ressource doit livrer. Si cette quantité est non nulle, alors la ressource peut réaliser les visites seulement si leur quantité à livrer ou ramasser est supérieure à la valeur renseignée dans cette contrainte.

*Format : numérique*

Exemple :

Une ressource dont la Quantité minimale de livraison est de 2, et un client A dont la Quantité est de 4, un client B dont la Quantité est de 1 et un client C dont la Quantité est de 2, seul le client A pourra être visité car  $4 > 2$ .

### Durée de chargement/déchargement au site (*fixedLoadingDuration*)

C'est le temps incompressible spécifique à la ressource, qu'elle doit passer au site lors d'un chargement ou d'un déchargement quel que soit le site. À ce temps, peuvent venir s'ajouter le temps saisi dans Durée de chargement/déchargement unitaire au site (ressources), Durée de chargement/déchargement fixe (sites) et Durée de chargement/déchargement unitaire (sites).

*Format : HH:MM:SS*

Exemple :

Pour une collecte, la ressource restera le temps indiqué dans Durée de chargement/déchargement au site, indépendamment de la quantité à collecter. Ce temps peut correspondre par exemple au temps d'actionner un hayon... à ajouter au temps propre à passer chez le client renseigné dans la contrainte Durée de la visite.

### Durée de chargement/déchargement unitaire au site (*loadingDurationPerUnit*)

Cette valeur correspond au temps nécessaire spécifique à la ressource pour charger/décharger une unité de produit au site. Cette valeur est multipliée par la quantité à charger ou à décharger selon le type de tournée. Cette contrainte s'applique uniquement sur la contrainte Quantité 1 des visites et non sur les 23 autres Quantités possibles.

*Format : HH:MM:SS*

Exemple :

Dans le cas d'une collecte de 10 bouteilles de gaz : le temps de chargement estimé pour une seule bouteille de gaz est de 3 minutes. On saisira la valeur 00:03:00 pour la contrainte Durée de chargement/déchargement unitaire au site. La ressource mettra ainsi 30 minutes pour la collecte (3min x 10 bouteilles).

### Durée de chargement/déchargement à chaque visite (***fixedUnloadingDuration***)

C'est le temps incompressible spécifique à la ressource, qu'elle doit passer en visite lors d'un chargement ou d'un déchargement quel que soit le client, c'est-à-dire un temps minimum de visite chez les clients. À ce temps, peuvent venir s'ajouter le temps saisi dans Durée de chargement/déchargement unitaire à chaque visite (ressources), Durée de visite (visites) et Durée de chargement/déchargement unitaire (visites).

*Format : HH:MM:SS*

Exemple :

Pour une livraison, la ressource restera le temps indiqué dans Durée de chargement/déchargement à chaque visite, indépendamment de la quantité à collecter. Ce temps peut correspondre par exemple au temps d'actionner un hayon... à ajouter au temps propre à passer chez le client renseigné dans la contrainte Durée de la visite.

### Durée de chargement/déchargement unitaire à chaque visite (***unloadingDurationPerUnit***)

Cette valeur correspond au temps nécessaire spécifique à la ressource pour livrer ou ramasser une unité de produit lors d'une visite. Cette valeur est multipliée par la valeur saisie dans la contrainte Quantité 1 de la visite. Cette contrainte s'applique uniquement sur la contrainte Quantité 1 des visites et non sur les 23 autres Quantités possibles.

*Format : HH:MM:SS*

Exemple :

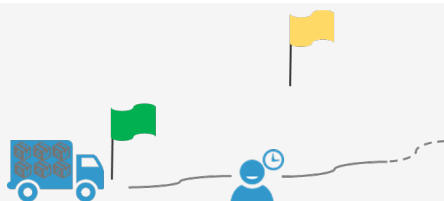
Dans le cas d'une livraison de 10 colis : le temps de déchargement estimé pour un seul colis est de 3 minutes. On saisira la valeur 00:03:00 pour la contrainte Durée de chargement/déchargement unitaire à chaque visite. La ressource mettra ainsi 30 minutes pour la livraison (3min x 10 colis).

### Première étape au dépôt (***loadBeforeDeparture***)

Cette contrainte permet d'indiquer si la ressource part chargée ou si elle doit s'arrêter charger à l'adresse d'un site ou à défaut à l'adresse où elle se situe avant la première visite.

Exemple :

Si le bouton est sur la valeur Désactivé, la ressource part chargée effectuer sa première visite.



Si le bouton est sur la valeur Activé, la ressource doit charger avant d'effectuer sa première visite.

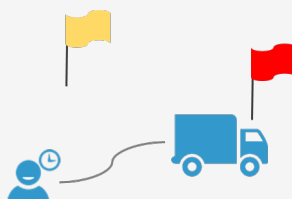


### Dernière étape au dépôt (*loadOnReturn*)

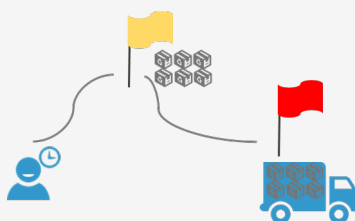
Cette contrainte permet d'indiquer si la ressource termine déchargée (vide) ou si elle doit s'arrêter recharger à l'adresse d'un site ou à défaut à l'adresse où elle se situe après la dernière visite.

Exemple :

Si le bouton est sur la valeur Désactivé, la ressource ne recharge pas après sa dernière visite.

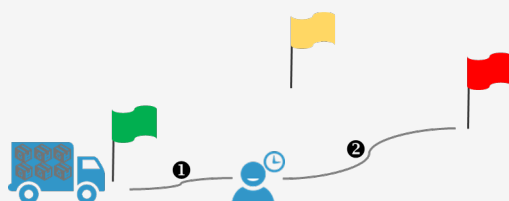


Si le bouton est sur la valeur Activé, la ressource doit recharger après sa dernière visite.

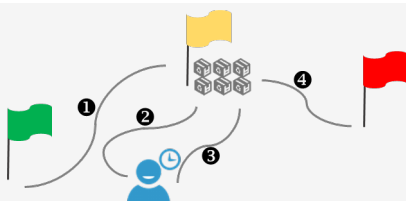


Exemple d'utilisation des contraintes Première étape au dépôt et Dernière étape au dépôt

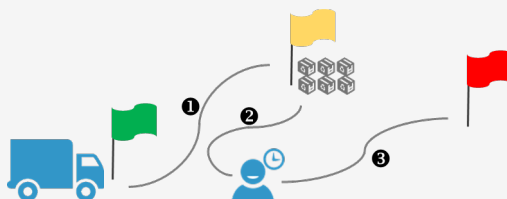
Si Première étape au dépôt et Dernière étape au dépôt sont sur désactivé



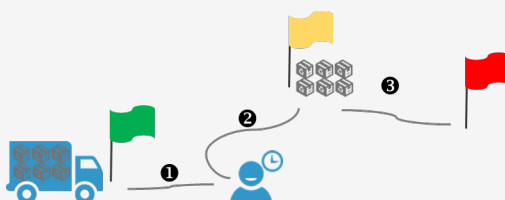
Si Première étape au dépôt et Dernière étape au dépôt sont sur activé



Si Première étape au dépôt est sur activé et Dernière étape au dépôt est sur désactivé



Si Première étape au dépôt est sur désactivé et Dernière étape au dépôt est sur activé



### Rechargement/Déchargement (*noReload*)

Cette contrainte permet d'interdire à une ressource de recharger au site durant une tournée. Si les contraintes Première étape au dépôt et Dernière étape au dépôt sont utilisées, la ressource ne rechargera jamais au site durant une tournée. Dans ce cas, la ressource chargera une fois au début ou à la fin, ou deux fois, au début et à la fin.

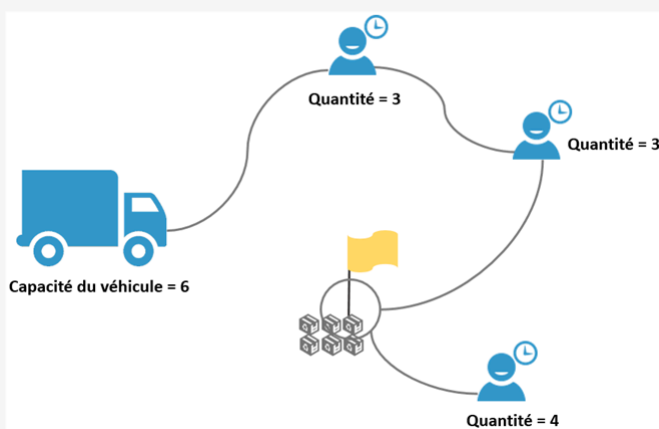
Si le bouton est sur la valeur Interdit, la ressource ne peut pas recharger.

Si le bouton est sur la valeur Autorisé, la ressource peut recharger.

La valeur par défaut est Autorisé.

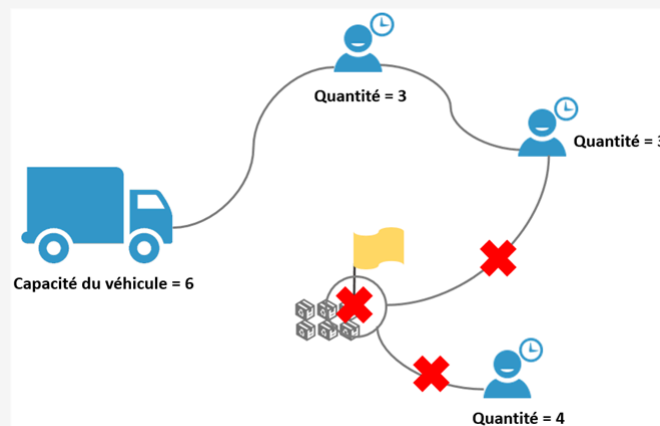
Exemple :

Si Rechargement/Déchargement est sur autorisé



Tous les clients pourront être visités même si la quantité totale à livrer dépasse la capacité initiale de la ressource, elle recharge au dépôt en cours de tournée.

Si Rechargement/Déchargement est sur interdit



Dans ce cas un client ne pourra pas être visité par cette même ressource car la capacité maximale de la ressource est atteinte et elle ne peut recharger au dépôt.

### Nombre maximum de rechargements/déchargements (*maximumReloads*)

Cette contrainte permet de limiter le nombre de rechargements quand celui-ci est autorisé dans la contrainte Rechargement/Déchargement. C'est un nombre maximum de rechargements qu'une ressource peut faire par jour.

*Format : numérique*

### Pénalité de rechargements/déchargements dépassés (*maximumReloadsPenalty*)

Cette contrainte permet de donner un coût supplémentaire appliqué par rechargement supérieur à la valeur exprimée dans la contrainte Nombre maximum de rechargements/déchargements.

*Format : numérique*

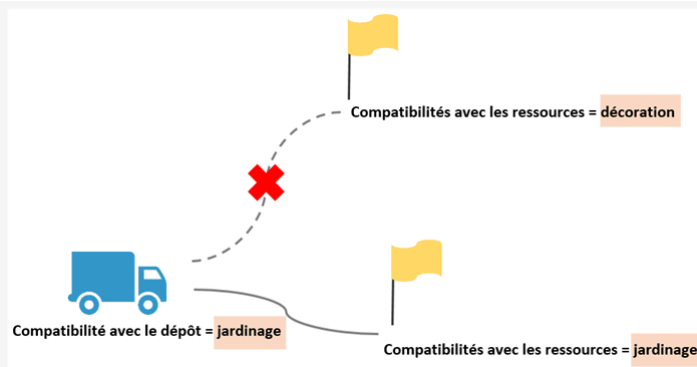
### Compatibilité avec le dépôt (*providedProducts*)

Cette contrainte désigne les critères requis par la ressource pour qu'elle puisse se rendre à un site.

*Format : caractère, exprimées sous la forme d'une liste de mots séparés par une virgule.*

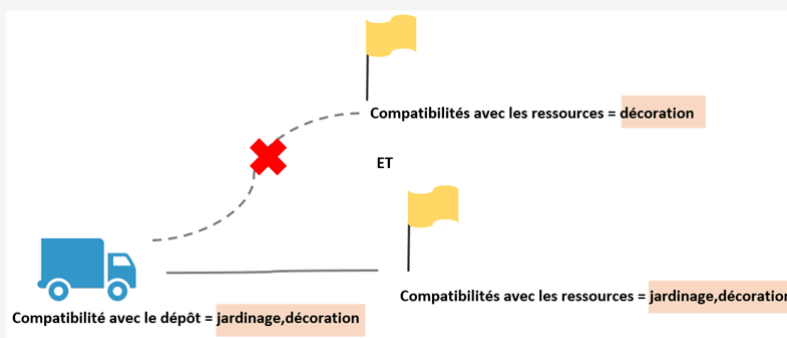
Exemple 1 :

Pour une ressource ayant besoin de passer à un dépôt pour retirer une commande concernant des outils de jardinage, on pourra indiquer "jardinage". Il faudra qu'au moins un dépôt dispose de ce critère pour que la ressource puisse s'y rendre.

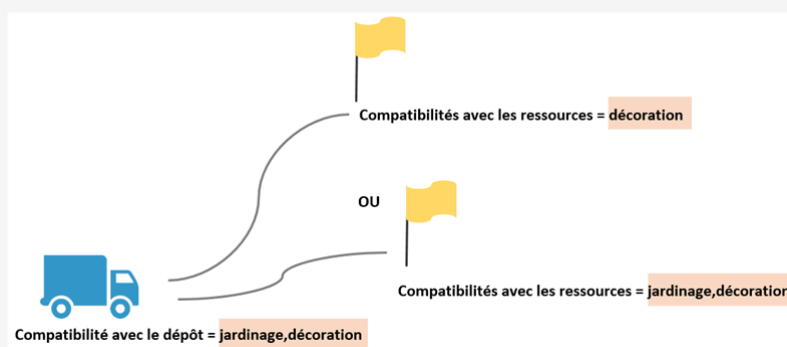


Exemple 2 :

Pour une ressource ayant besoin de passer à un dépôt pour retirer une commande concernant des outils de jardinage ET une autre commande concernant de la décoration d'intérieure, on pourra indiquer "jardinage,décoration". Il faudra qu'au moins un dépôt dispose de ces 2 compatibilités pour que la ressource puisse s'y rendre. Il faut donc mettre la valeur 1 dans la contrainte Toutes les compatibilités.



Si la contrainte Toutes les compatibilités = 0 alors la prise en compte des 2 compatibilités ne sera pas demandée.



## Coûts

Cette partie concerne les contraintes de coût. Le moteur de TourSolver s'appuie sur ces contraintes pour optimiser vos tournées. Son but premier est de vous proposer la tournée avec un coût total le plus

bas, tout en respectant les autres contraintes. C'est pourquoi il est important de bien considérer ces contraintes, lorsqu'on parle de coût, on parle surtout de poids. TourSolver étant un logiciel, il n'a pas de notion monétaire mais bien un poids.

## Forfait journée (*payWholeDay*)

Cette contrainte permet d'indiquer si toutes les heures de la journée sont comptabilisées ou seulement celles utilisées.

Si le bouton est sur la valeur Désactivé, seules les heures travaillées sont comptabilisées.

Si le bouton est sur la valeur Activé, toutes les heures de la journée sont comptabilisées.

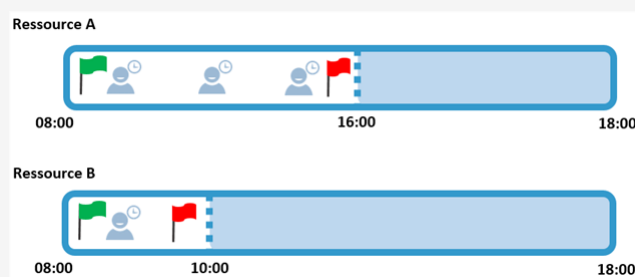
Cette contrainte a une influence sur le remplissage des plannings des ressources.

Exemple :

Soit 2 ressources A et B : Coût horaire = 20 Horaires journaliers = 08:00 et 18:00

Si Forfait journée est sur désactivé

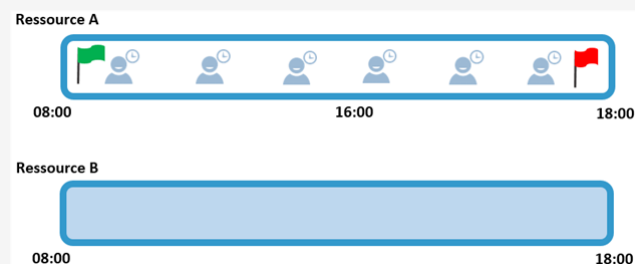
TourSolver peut proposer ce planning :



Dans ce cas, la ressource A travaille donc 8h et la ressource B 2h. Soit un coût total de 200 ( $8 \times 20 + 2 \times 20$ ) calculé par TourSolver.

Si Forfait journée est sur activé

TourSolver peut proposer ce planning :



La ressource A travaille donc 10h et la ressource B 0h. Soit un coût total de 200 ( $10 \times 20$ ) calculé par TourSolver et une seule ressource utilisée.

## Coût fixe de la ressource utilisée (*useInPlanningPenalty*)

Cette contrainte permet d'indiquer un coût fixe d'emploi journalier de la ressource, qui sera appliqué chaque jour où celle-ci est utilisée dans une tournée.

*Format : numérique*

Exemple :

La flotte de la société de livraison est constituée d'une ressource A propre et d'une ressource B affrétée. Dans ce cas, la société préfère utiliser en priorité sa propre ressource avec les contraintes Coût horaire et Coût kilométrique renseignées et dans un second temps la ressource affrétée qui a un coût d'utilisation global à la journée.

Soit Ressource A :

Coût horaire = 20

Coût kilométrique = 1

Coût fixe de la ressource utilisée = 0

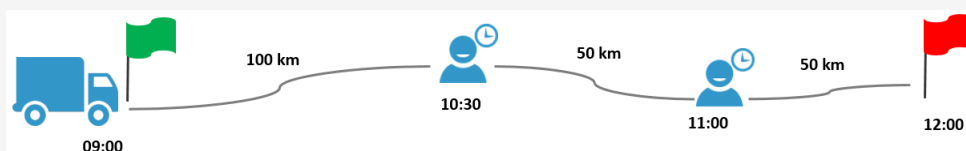
Soit Ressource B :

Coût horaire = 0

Coût kilométrique = 0

Coût fixe de la ressource utilisée = 600

Tournée 1/2



Cette tournée coûte 3h de travail (12-9) et 200 km (100+50+50). Pour livrer ces 2 clients, TourSolver calculera quelle ressource entre A et B sera la moins coûteuse :

Ressource A =  $240 (20 \cdot 2h + 1 \cdot 200km + 0)$

Ressource B =  $600 (0 \cdot 2h + 0 \cdot 200km + 600)$

Au vu du coût total de la tournée, TourSolver choisira d'utiliser la Ressource A pour effectuer cette tournée.

Tournée 2/2



Cette tournée coûte 6h de travail (15-9) et 550 km (250+150+150). Pour livrer ces 2 clients, TourSolver calculera quelle ressource entre A et B sera la moins coûteuse :



Ressource A = 670 ( $20 \times 6h + 1 \times 550km + 0$ )

Ressource B = 600 ( $0 \times 6h + 0 \times 550km + 600$ )

Au vu du coût total de la tournée, TourSolver choisira d'utiliser la Ressource B pour effectuer cette tournée.

### Coût quotidien de la ressource non utilisée (*nonUsePenalty*)

Cette contrainte permet d'indiquer un coût fixe de non utilisation journalier de la ressource, qui sera appliqué chaque jour où celle-ci ne sera pas utilisée dans une tournée. Cette contrainte peut être utilisée pour intégrer dans le calcul d'optimisation, les coûts encourus par une société même si la ressource est immobilisée (amortissement, parking, ...).

*Format : numérique*

Exemple :

Soit Ressource A :

Coût quotidien de la ressource non utilisée = 0

Et Ressource B :

Coût quotidien de la ressource non utilisée = 600

1) Si TourSolver choisit de faire travailler la Ressource A, alors cette tournée coûtera 600 ( $0+600$ ) car la ressource B non utilisée coûtera tout de même 600.

2) Si TourSolver choisit de faire travailler la Ressource B, alors cette tournée coûtera 0 ( $0+0$ ) car la ressource A non utilisée ne coûte rien contrairement à la ressource B.

TourSolver retiendra donc la deuxième solution.

### Coût de sollicitation de la ressource (*usePenalty*)

Cette contrainte permet d'indiquer un coût fixe d'emploi de la ressource sur la totalité de sa tournée, qui sera appliqué dès sa mise en service. A la différence du Coût fixe de la ressource utilisée, celui-ci ne s'applique pas à chaque jour travaillé par la ressource mais bien dès le premier jour travaillé.

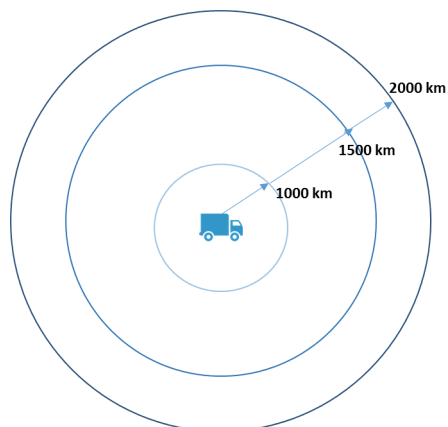
*Format : numérique*

### Coût km 2 à 4

*Format : numérique*

Seuil [distance\_2...4] : Cette contrainte permet d'indiquer la distance à partir de laquelle un coût différent s'appliquera.

Coût [penalty\_2...4] : Cette contrainte permet d'indiquer le coût par unité de trajet (km, miles...) parcourue par la ressource à partir d'une certaine distance renseignée dans la contrainte Seuil.



Exemple :

Soit une société dont la flotte est composée de ressources affrétées parcourant de nombreux km.

Coût km = 1

Seuil coût km 2 = 1000

Coût km 2 = 1,5

Seuil coût km 3 = 1500

Coût km 3 = 2

Si ce transporteur effectue 2000 km dans la semaine, alors le coût kilométrique total = 4000 (2000 x 2). C'est le Coût km 3 = 2 indiqué pour le Seuil coût km 3 qui s'applique car 2000 > 1500.

### Coût fixe de la visite (*penaltyPerVisit*)

Cette contrainte permet de définir pour chaque ressource un coût fixe à ajouter à chaque visite assurée par une ressource. Ainsi, TourSolver viendra ajouter ce coût au coût global d'une ressource.

Format : numérique

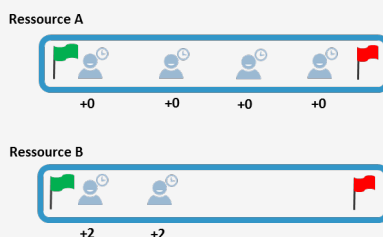
Exemple :

Pour la ressource A, Coût fixe de la visite = 0,

Pour la ressource B, Coût fixe de la visite = 2.

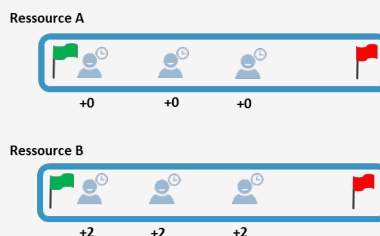
TourSolver cherchera à remplir le planning de la ressource A d'abord car le coût de la tournée est moins important que la tournée de la ressource B, qui voit son coût global augmenté en fonction du nombre de visites.

Exemple de tournée retenue par TourSolver :

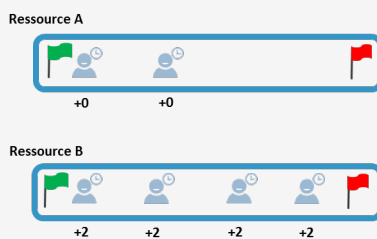


Le coût global de cette tournée s'élève à 4 (2+2+0+0+0+0).

Exemple de tournées non retenues par TourSolver :



Le coût global de cette tournée s'élève à 6 (2+2+2+0+0+0).



Le coût global de cette tournée s'élève à 8 (2+2+2+2+0+0).

## Nuitées

### Maximum de nuitées consécutives (*maxNightsOutPerJourney*)

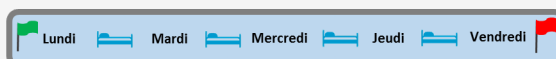
Cette contrainte permet d'indiquer le nombre de nuitées maximum consécutives sans retour à la base.

*Format : numérique*

Exemple :

Un commercial travaille en tournée du lundi jusqu'au vendredi sans retourner à la base.

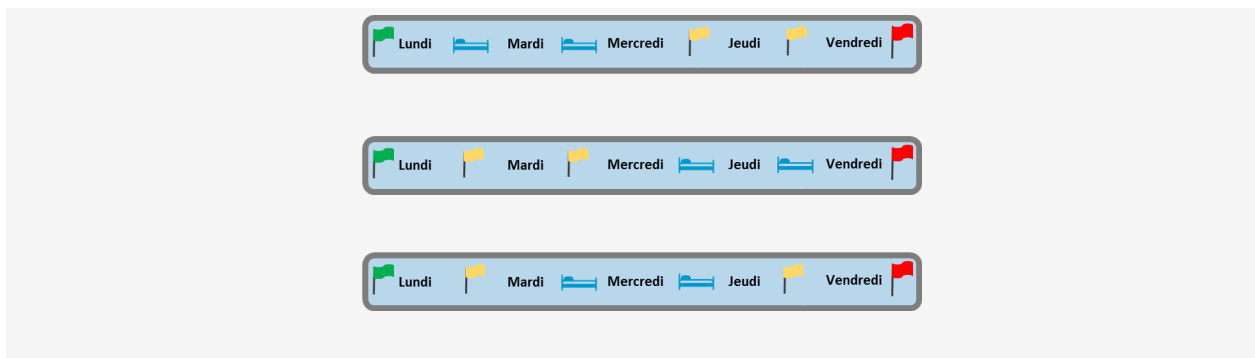
Jours travaillés = 1-5 et Maximum de nuitées consécutives = 4



Un commercial travaille en tournée du lundi jusqu'au vendredi mais limité à 2 nuits sans retourner à la base.

Jours travaillés = 1-5 et Maximum de nuitées consécutives = 2

TourSolver proposera les tournées suivantes :



### Coût d'une nuitée (*nightPenalty*)

Cette contrainte permet de renseigner un coût global d'une nuitée. TourSolver prendra en compte ce coût pour savoir s'il est préférable de faire découcher la ressource ou de la faire revenir à la base. Plus le coût est faible, plus la nuitée est permise.

Format : numérique

### Temps de retour max. avant nuitée (*overnightMinDriving*)

Cette contrainte permet de définir le temps de conduite maximum permettant à la ressource un retour à la base plutôt qu'une nuitée.

Format : HH:MM

Exemple 1 :

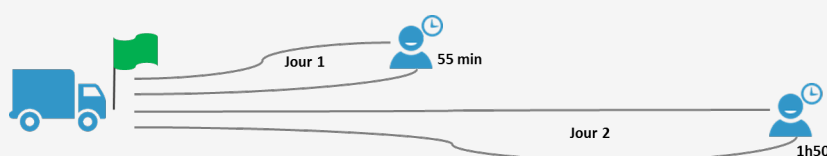
Temps de retour max. avant nuitée = 00:30 indique que si la ressource est à plus de 30 minutes de la base, sa nuitée aura lieu à l'extérieur. Au contraire, si le temps la séparant de la base est inférieur à 30 minutes, alors elle rentrera à la base.



Dans ce cas, la ressource découche car elle est à plus de 30 minutes de la base.

Exemple 2 :

Temps de retour max. avant nuitée = 02:00 indique que si la ressource est à plus de 2 heures de la base, sa nuitée aura lieu à l'extérieur. Au contraire, si le temps la séparant de la base est inférieur à 2 heures, alors elle rentrera à la base.



---

Dans ce cas, la ressource ne découche pas car elle est à moins de 2h de la base.

## Législation de conduite

### Temps de conduite sans pause [*legalMaxDriveDuration*]

Cette contrainte permet d'indiquer la durée maximum de conduite que la ressource peut effectuer sans pause.

*Format : HH:MM*

Exemple :

Une ressource ne peut pas conduire plus de 4h30 consécutives sans faire de pause, alors Temps de conduite sans pause = 04:30

### Durée de la pause conduite (*legalDriveRestDuration*)

Cette contrainte permet d'indiquer la durée de la pause à effectuer lorsqu'une ressource a atteint le nombre maximum d'heures de conduite continue avant de pouvoir conduire à nouveau.

*Format : HH:MM*

Exemple :

Une ressource ne peut pas conduire à nouveau si elle n'a pas fait de pause de 45 minutes, alors Durée de la pause conduite = 00:45.

### Durée min. de la pause conduite (*legalMinRestDuration*)

Cette contrainte permet d'indiquer la durée minimum d'une pause pour qu'elle soit comptabilisée comme temps de repos pendant une période de conduite.

*Format : HH:MM*

Le temps de pause conduite n'est pas considéré par TourSolver comme du temps de travail. La durée d'une journée de travail en est allongée d'autant.

Exemple :

Une pause est considérée comme du repos à partir de 15 minutes, alors Durée min. de la pause conduite = 00:15

Exemple : illustration des contraintes Temps de conduite sans pause + Durée de la pause conduite + Durée min. de la pause conduite

Si Temps de conduite sans pause = 04:30 ;

Si Durée de la pause conduite = 00:45 ;

Si Durée min. de la pause conduite = 00:15 ;

Alors la ressource peut conduire 4h30 s'il a fait 45 minutes de pause. Cette pause peut se faire en 3 pauses de 15 minutes ou une pause de 30 minutes puis une autre de 15 minutes par exemple. En revanche s'il a fait 2 pauses de 20 minutes et une de 10 minutes, il sera obligé de refaire une pause d'au moins 15 minutes pour pouvoir conduire 4h30 car celle de 10 minutes n'est pas compté comme une pause puisque la durée est inférieure à 15 minutes.

### Pause conduite en visite (*driveRestAtCustomer*)

Cette contrainte permet d'indiquer si le temps passé à une visite est considéré comme un temps de repos.

Si le bouton est sur la valeur Désactivé, le temps n'est pas compté comme du repos ;

Si le bouton est sur la valeur Activé, le temps est compté comme du repos.

### Pause conduite au dépôt (*driveRestAtDepot*)

Cette contrainte permet d'indiquer si le temps passé au site lors d'un rechargement est considéré comme un temps de repos.

Si le bouton est sur la valeur Désactivé, le temps n'est pas compté comme du repos ;

Si le bouton est sur la valeur Activé, le temps est compté comme du repos.

### Durée quotidienne de conduite (*legalDailyDriveDuration*)

Cette contrainte permet d'indiquer la durée maximum de conduite que la ressource peut effectuer sans nuitée sur une journée. Cette contrainte permet d'indiquer la durée maximum de conduite cumulée que la ressource peut effectuer sans pause nuitée dans une journée. A l'issue de ce temps de conduite, la ressource doit obligatoirement prendre une pause de la durée spécifiée dans Durée de la nuitée.

*Format : HH:MM*

Exemple :

Une ressource ne peut pas conduire plus de 9h sans faire de pause nuitée, alors Durée quotidienne de conduite = 09:00

### Durée de la nuitée (*legalDailyRestDuration*)

Cette contrainte permet d'indiquer la durée de la pause nuitée effectuée par une ressource dès que la Durée quotidienne de conduite est dépassée.

*Format : HH:MM*

Exemple :

Une ressource ne peut pas conduire à nouveau si elle n'a pas fait une nuit de 4h, alors Durée de la nuitée = 04:00.

## Contraintes des sites

- ! Dans cette documentation sont employés les mots-clés de la terminologie par défaut définie dans la page *Mon activité* de l'interface.  
Ainsi, on parlera ici de "site" mais les noms des contraintes peuvent évoluer en fonction de vos propres choix de terminologie.

- ! Après chaque nom de contrainte, les parenthèses contiennent le nom équivalent de cette contrainte dans [TourSolver Cloud API](https://geoservices.geoconcept.com/ToursolverCloud/api-book.html) [https://geoservices.geoconcept.com/ToursolverCloud/api-book.html].

### Nom du site (*id*)

Cette valeur permet d'identifier le site. Le nom peut être un patronyme, le nom de la ville, un identifiant...

*Format : caractère*

Exemple :

Dépôt A, BAGNEUX, Agence Nantes3

Cette contrainte doit être obligatoirement renseignée.

Il est fortement recommandé que cette valeur soit unique.

### Horaires d'ouverture 1

**(*timeWindow[0].beginTime,timeWindow[0].endTime*)**

Ces contraintes permettent de définir un créneau horaire dans lequel les ressources pourront se rendre au site. Heure d'ouverture 1 correspond à l'heure à partir de laquelle la ressource peut effectuer le (re)chargement. Heure de fermeture 1 correspond à l'heure au-delà de laquelle la ressource ne peut plus effectuer le (re)chargement. Cette plage horaire s'applique à l'ensemble des jours sauf si Jours d'ouverture 1 est défini. Dans ce cas, la plage horaire est appliquée seulement à cette valeur.

*Format : HH:MM*

Un (re)chargement ne peut démarrer avant l'heure de début. Un (re)chargement démarré dans le créneau horaire peut se poursuivre au-delà de l'heure de fin. Si l'on souhaite qu'un (re)chargement se termine avant une certaine heure il faut retrancher de cette heure le temps du (re)chargement (si ce temps est fixe) et renseigner cette nouvelle heure dans la contrainte Heure de fermeture 1.

Si une plage horaire n'est pas renseignée pour un site, la ressource peut effectuer le (re)chargement n'importe quand sur les éventuels Jours d'ouverture définis et dans la limite des horaires de travail de la

ressource. On peut définir jusqu'à 4 créneaux horaires différents (et leurs jours associés) pour chaque site.

- ❗ Les créneaux définis à cheval sur plusieurs jours ne sont pas gérés. L'utilisateur ne peut pas utiliser de créneau démarrant à 23:00 et finissant à 03:00 le lendemain matin mais une solution de contournement existe.

Exemple 1 :

Heure d'ouverture 1 = 07:30 et Heure de fermeture 1 = 11:30 définissent un créneau horaire de 4 heures pendant lesquels la ressource peut effectuer le (re)chargement.

Exemple 2 :

La ressource peut effectuer le (re)chargement au dépôt entre 23:00 et 03:00 du matin. Dans ce cas, il faut décaler l'ensemble des horaires (dans un sens ou l'autre) afin qu'ils soient tous contenus dans une journée et récupérer l'ordonnancement de la tournée. Heure d'ouverture 1 = 19:00 et Heure de fermeture 1 = 23:00 définissent un créneau horaire de 4 heures pendant lesquels la ressource peut effectuer le (re)chargement.

Post optimisation, il faudra décaler à nouveau l'ensemble des horaires des visites et (re)chargement pour retrouver la plage horaire d'origine. Ce (re)chargement a été planifié par TourSolver à 21:00, dans la réalité il sera effectué à 01:00 du matin (=23:00+2h). Il faudra donc ajouter la constante de 2h à toutes les données horaires calculées.

## Horaires d'ouverture 2

***(timeWindow[1].beginTime,timeWindow[1].endTime)***

Ces contraintes permettent de définir un deuxième créneau horaire qui s'applique aux jours de visite définis dans Jours d'ouverture 2. Heure d'ouverture 2 correspond à l'heure à partir de laquelle la ressource peut effectuer le (re)chargement. Heure de fermeture 2 correspond à l'heure au-delà de laquelle la ressource ne peut plus effectuer le (re)chargement.

*Format : HH:MM*

Exemple 1 :

La ressource peut effectuer le (re)chargement au dépôt entre 09:00 et 12:00 et l'après-midi entre 14:00 et 17:00. Pour cela, on utilisera la contrainte Heure d'ouverture 1 = 09:00, Heure de fermeture 1 = 12:00 et Heure d'ouverture 2 = 14:00, Heure de fermeture 2 = 17:00

Exemple 2 :

Le dépôt est ouvert le lundi entre 07:00 et 17:00 et le mercredi entre 09:00 et 18:00, dans ce cas, Heure d'ouverture 1 = 07:00, Heure de fermeture 1 = 17:00, Jours d'ouverture 1 = 1 et Heure d'ouverture 2 = 09:00, Heure de fermeture 2 = 18:00, Jours d'ouverture 2 = 3



## Horaires d'ouverture 3

(*timeWindow[2].beginTime,timeWindow[2].endTime*)

Ces contraintes permettent de définir un deuxième créneau horaire qui s'applique aux jours de visite définis dans Jours d'ouverture 3. Heure d'ouverture 3 correspond à l'heure à partir de laquelle la ressource peut effectuer le (re)chargement. Heure de fermeture 3 correspond à l'heure au-delà de laquelle la ressource ne peut plus effectuer le (re)chargement.

## Horaires d'ouverture 4

(*timeWindow[3].beginTime,timeWindow[3].endTime*)

Ces contraintes permettent de définir un deuxième créneau horaire qui s'applique aux jours de visite définis dans Jours d'ouverture 4. Heure d'ouverture 4 correspond à l'heure à partir de laquelle la ressource peut effectuer le (re)chargement. Heure de fermeture 4 correspond à l'heure au-delà de laquelle la ressource ne peut plus effectuer le (re)chargement.

## Jours d'ouverture 1 (*openingDays[0]*)

Cette contrainte contient l'ensemble des jours d'ouverture de visite au site (jusqu'à 64) sur lesquels s'applique le créneau horaire défini par les contraintes Heure d'ouverture 1 et Heure de fermeture 1.

*Format : caractère*

Ces jours peuvent être définis un à un (1,4) ou par créneau (1-4), en nombre entier (1,2,...64) ou au format date (14/05/2016 => 17/07/2016).

Exemple :

Jours d'ouverture 1 = 1,2,5 (ou 14/05/2016,15/05/2016,18/05/2016 ) signifie que le (re)chargement peut s'effectuer les jours 1, 2 et 5 du planning (ou les 14/05/2016, 15/05/2016 et 18/05/2016) aux horaires définis par les contraintes Heure d'ouverture 1 et Heure de fermeture 1. Jours d'ouverture 1 = 1-5 (ou 14/05/2016 => 18/05/2016) signifie que le (re)chargement peut s'effectuer tous les jours 1, 2, 3, 4 et 5 du planning (ou entre le 14/05/2016 et le 18/05/2016) aux horaires définis par les contraintes Heure d'ouverture 1 et Heure de fermeture 1.

- ⓘ Attention, cette contrainte doit être mise en relation avec la contrainte Jours tournée des Ressources et utiliser le même format (date ou nombre). Aucune date de Jours d'ouverture 1 ne peut être antérieure à la plus ancienne date de Jours tournée.

## Jours d'ouverture 2 (*openingDays[1]*)

Cette contrainte contient l'ensemble des jours d'ouverture de visite au site (jusqu'à 64) sur lesquels s'applique le créneau horaire défini par les contraintes Heure d'ouverture 2 et Heure de fermeture 2.

*Format : caractère*

Exemple 1 :

Le dépôt est ouvert le lundi entre 07:00 et 17:00 et le mercredi entre 09:00 et 18:00 : Heure d'ouverture 1 = 07:00, Heure de fermeture 1 = 17:00, Jours d'ouverture 1 = 1 et Heure d'ouverture 2 = 09:00, Heure de fermeture 2 = 18:00, Jours d'ouverture 2 = 3

Exemple 2 :

Le dépôt est ouvert du lundi au vendredi entre 07:00 et 17:00 et le samedi entre 09:00 et 12:00 : Heure d'ouverture 1 = 07:00, Heure de fermeture 1 = 17:00, Jours d'ouverture 1 = 1-5 et Heure d'ouverture 2 = 09:00, Heure de fermeture 2 = 12:00, Jours d'ouverture 2 = 6

### Jours d'ouverture 3 (*openingDays[2]*)

Cette contrainte contient l'ensemble des jours d'ouverture de visite au site (jusqu'à 64) sur lesquels s'applique le créneau horaire défini par les contraintes Heure d'ouverture 3 et Heure de fermeture 3.

### Jours d'ouverture 4 (*openingDays[3]*)

Cette contrainte contient l'ensemble des jours d'ouverture de visite au site (jusqu'à 64) sur lesquels s'applique le créneau horaire défini par les contraintes Heure d'ouverture 4 et Heure de fermeture 4.

## Gestion du site

### Affecter des ressources (*resourceNames*)

Cette contrainte permet à l'utilisateur de spécifier, pour un site, la liste des ressources qui peuvent le visiter. La valeur à inscrire dans cette contrainte doit être égale à la valeur renseignée dans la contrainte Nom des ressources.

*Format : caractère, sous forme de liste séparée par une virgule si les ressources sont multiples.*

### Exclure des ressources (*excludeResources*)

Cette contrainte permet à l'utilisateur de spécifier, pour un site, la liste des ressources qui ne peuvent pas le visiter. La valeur à inscrire dans cette contrainte doit être égale à la valeur renseignée dans la contrainte Nom des ressources.

*Format : caractère, sous forme de liste séparée par une virgule si les ressources sont multiples.*

### Disponibilité (*availability*)

Cette contrainte permet de prendre en considération les sites mis à disposition sans avoir à ajouter ou enlever des données dans le jeu de données complet.

*Format : binaire*

Si le bouton est sur la valeur Oui, le site est pris en compte dans la tournée.

Si le bouton est sur la valeur Non, le site n'est pas pris en compte dans la tournée.

Par défaut, le site est disponible.

### Priorité d'un site (*priority*)

Cette contrainte permet de rendre un site prioritaire par rapport aux autres. Si deux sites sont à proximité, TourSolver choisira en priorité celui qui a la valeur la plus grande. Si deux sites ne sont pas à proximité, TourSolver regardera celui qui a la valeur la plus grande tout en tenant compte de la distance séparant la ressource et les sites.

*Format : numérique (entier)*

## Stock

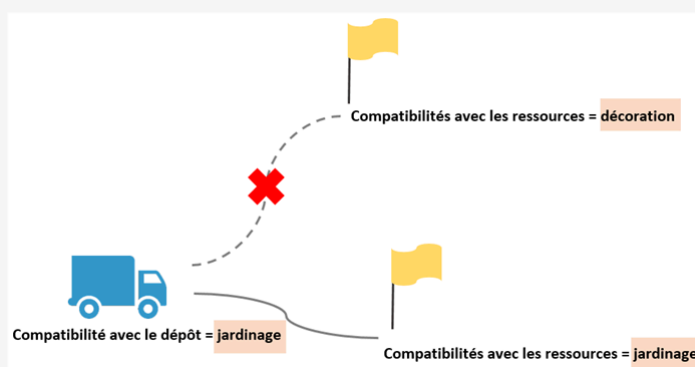
### Compatibilités avec les ressources (*requiredProducts*)

Cette contrainte désigne les critères requis par le site pour que la ressource puisse s'y rendre.

*Format : caractères (exprimé sous la forme d'une liste de mots séparés par une virgule.)*

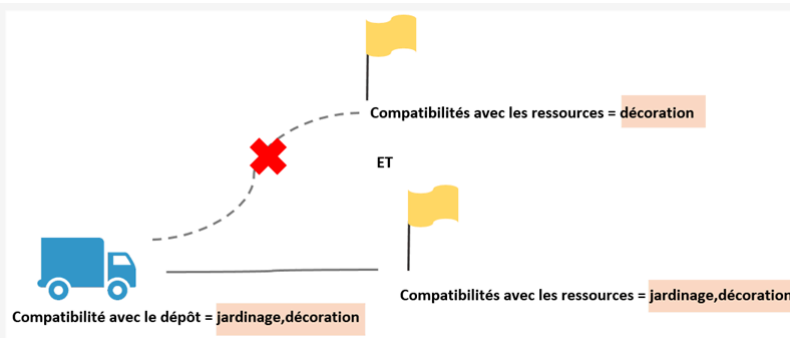
Exemple 1 :

Pour une ressource ayant besoin de passer à un dépôt pour retirer une commande concernant des outils de jardinage, on pourra indiquer "jardinage". Il faudra qu'au moins un dépôt dispose de ce type de produit pour que la ressource puisse s'y rendre.



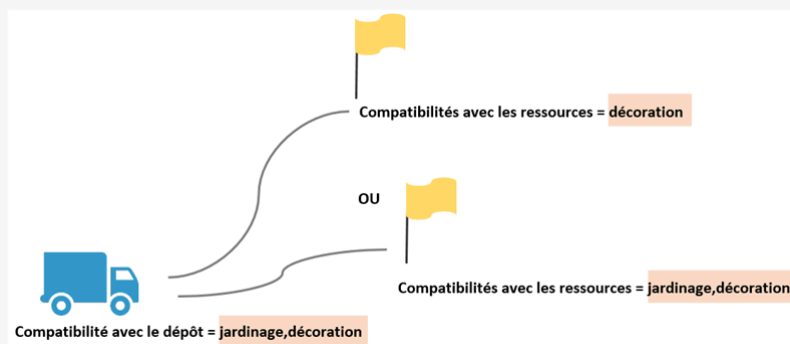
Exemple 2 :

Pour une ressource ayant besoin de passer à un dépôt pour retirer une commande concernant des outils de jardinage ET une autre commande concernant de la décoration d'intérieur, on pourra indiquer "jardinage,décoration". Il faudra qu'au moins un dépôt dispose de ces 2 compatibilités pour que la ressource puisse s'y rendre. Il faut en complément, utiliser la contrainte Toutes les compatibilités avec la valeur 1.



Exemple 3 :

Par défaut, sans l'utilisation de la contrainte Toutes les compatibilités (ou si celle-ci est utilisée avec la valeur 0), une seule des 2 compatibilités est nécessaire.



### Toutes les compatibilités (*allProductsRequired*)

Si la contrainte Compatibilités avec les ressources est utilisée (et que les valeurs sont nombreuses), cette contrainte permet d'indiquer si un seul des critères de la liste est requis ou si la liste complète des critères est requise.

*Format : binaire*

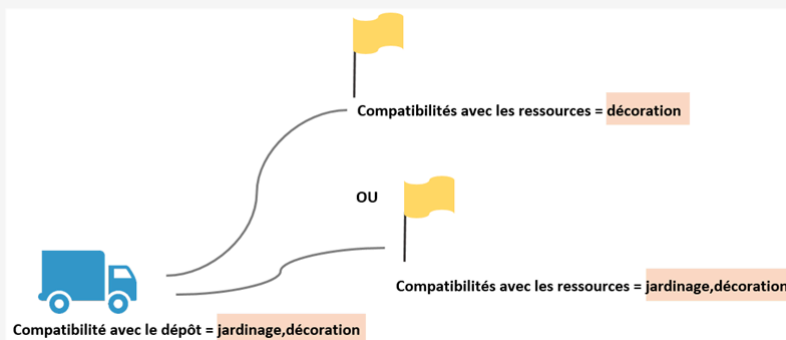
Si le bouton est sur la valeur Non, un seul des critères est requis.

Si le bouton est sur la valeur Oui, la liste complète est requise.

La valeur par défaut est égale à Non.

Exemple :

Si la contrainte Toutes les compatibilités est sur la valeur Non alors la prise en compte des 2 critères ne sera pas demandée. Seulement un des 2 critères est suffisant.



## Durée de chargement/déchargement fixe (*fixedLoadingDuration*)

C'est le temps incompressible à passer au site lors d'un chargement ou d'un déchargement, c'est-à-dire un temps minimum de passage au site. A ce temps, peut venir s'ajouter le temps saisi dans la contrainte Durée de chargement/déchargement unitaire.

*Format : HH:MM:SS*

Exemple :

Durée de chargement/déchargement fixe = 00:30:00 signifie que 30 minutes sont nécessaires pour charger/décharger au site. Ce temps peut correspondre par exemple au temps de préparation pour charger ou décharger.

## Durée de chargement/déchargement unitaire (*loadingDurationPerUnit*)

Cette valeur correspond au temps propre au site pour charger ou décharger une unité de produit. Cette valeur est multipliée par la quantité à charger ou à décharger selon le type de tournée. Cette contrainte s'applique uniquement sur la contrainte Quantité 1 des visites et non sur les 23 autres Quantités possibles.

*Format : HH:MM:SS*

Exemple :

Durée de chargement/déchargement fixe = 00:30:00 et Durée de chargement/déchargement unitaire = 120 signifie que 120 sec sont nécessaires pour charger une unité de produit. Si la quantité à charger est de 8, alors 16 minutes (120\*8) seront nécessaires pour charger ou décharger la quantité totale au site. La durée totale passée au site pour ce chargement ou déchargement est alors de 46 minutes (30 + 16).

## Quantité initiale (*deliveryQuantity[0]*)

Cette contrainte correspond à la quantité disponible du stock au site qu'on peut charger.

*Format : numérique*

Exemple 1 :

La ressource peut aller charger jusqu'à 6 quantités à ce dépôt.



Exemple 2 :

La ressource ne peut pas aller charger à ce dépôt car les quantités n'y sont pas disponibles.



### Quantité initiale 2,...quantité 24 (*deliveryQuantity[1],...deliveryQuantity[23]*)

Cette contrainte correspond à la quantité 2,...24, disponible en stock au site pour un produit à charger. Ces contraintes permettent de distinguer plusieurs produits ou unité de mesure.

*Format : numérique*

### Espace disponible (*pickupQuantity[0]*)

Cette contrainte correspond à la quantité disponible du stock qu'on peut décharger.

*Format : numérique*

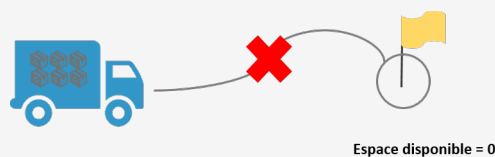
Exemple 1 :

La ressource peut aller décharger jusqu'à 6 espaces (quantités) à ce dépôt.



Exemple 2 :

La ressource ne peut pas aller décharger à ce dépôt car l'espace n'y est pas suffisant.



---

Espace disponible 2,...,espace disponible 24 (***pickupQuantity[1],...  
pickupQuantity[23]***)

Cette contrainte correspond à la quantité 2,...24, disponible en stock au site pour un produit à télécharger.  
Ces contraintes permettent de distinguer plusieurs produits ou unité de mesure.

*Format : numérique*

