

Cours d'Optimisation Combinatoire C9-4

ENSTA - année 2011-2012

Projet **GoTIC** : Gestion Optimale des tournées pour les Techniciens d'Intervention Clients

T. Defaix, O. Klopfenstein, É. Gourdin, M. Diamantini

06/01/2012 (version 2.6)

1 Contexte télécom

Les TIC (Techniciens d'Intervention Clients) sont chargés de répondre aux demandes des clients d'un opérateur de télécommunication en intervenant physiquement chez le client (par exemple, installation d'une ligne téléphonique) ou sur le réseau (par exemple, connection d'une nouvelle ligne ADSL). Les demandes d'interventions sont recueillies par différents canaux (centres d'appel, agences, etc...) et sont centralisées au niveau du service qui doit planifier l'activité des TIC. Chaque intervention est caractérisée par un lieu géographique, une plage horaire où l'intervention doit avoir lieu et un type de compétence requis pour pouvoir effectuer l'intervention. Le plan de charge des TIC est élaboré toutes les semaines, en fonction des techniciens disponibles, puis revu tous les jours en fonction des aléas. L'objectif du planificateur est de satisfaire toutes les interventions dans les plages horaires prévues et ce, en minimisant la somme des longueurs des tournées réalisées par les TIC.

2 Problème

Le problème de planification des tournées des TIC s'apparente au problème académique du VRP/TW (Vehicule Routing Problem with Time Windows) dans lequel il faut planifier des tournées partant d'un dépôt central et visitant chaque client (VRP), et ce, dans une fenêtre de temps (Time Window). De plus, dans le VRP/TW, les tournées sont réalisées dans l'objectif de livrer ou de collecter des marchandises. Le problème académique considère donc que les véhicules ont des capacités et que chaque client demande une certaine quantité ou un certain volume de marchandises.

Le problème que l'on considère ici est donc différent du VRP/TW sur plusieurs aspects :

- pas de capacités ni de demandes : les véhicules ne sont pas limités par une capacité pour transporter des marchandises,
- dépôts multiples : contrairement au VRP où toutes les tournées partent du même noeud (le dépôt central), chaque TIC a son propre noeud de départ et d'arrivée (certains TIC partent directement de chez eux, d'autres partent d'une base, mais il peut y avoir plusieurs bases),
- typologie des interventions: chaque intervention nécessite des compétences spécifiques et ne peut donc être réalisée que par les TIC qui ont cette compétence.

3 Hypothèses et notations

Dans le projet GoTIC, on fait les hypothèses suivantes :

- Horizon de temps : on suppose que l'on optimise les tournées pour une journée. Toutes les informations fournies concernent donc la journée courante.
- Distances euclidiennes : la situation géographique des bases de départ des techniciens et des lieux d'intervention est décrite au moyen de coordonnées cartésiennes (x, y) dans un carré $[0, 100] \times [0, 100]$ (l'unité est le kilomètre). Les distances entre deux points dans ce carré seront les distances euclidiennes arrondies à l'entier au plus proche. Ainsi, la distance entre les points $A = (10, 48)$ et $B = (63, 25)$ est donnée par

$$\begin{aligned} \text{dist}(A, B) &= \text{arrondi}(\sqrt{(10 - 63)^2 + (48 - 25)^2}) \\ &= \text{arrondi}(57.775 \dots) = 58\text{km}. \end{aligned}$$

- Vitesse de déplacement : on suppose que les techniciens se déplacent à une vitesse moyenne notée *speed* et fournie dans le fichier d'entrée de chaque instance. **La durée des déplacements calculée à partir de la distance arrondie sera elle-même arrondie à la minute la plus proche.** Ainsi, si la vitesse moyenne est de $\text{speed} = 50$ km/h, alors un technicien parcourt la distance entre *A* et *B* en $58/50 = 1.16$ heure = 69.6 min, qu'on arrondit à 1h10min.
- Discrétisation du temps : en ce qui concerne les données d'entrée, on suppose que la journée de 24h est discrétisée par pas de temps de une minute. On utilisera un entier t compris entre 0 et 1439 pour représenter le t -ième pas de temps. Ainsi, le temps $t = 480$ représente la minute débutant à 8h00 du matin et $t = 1080$ représente 18h00.

Chaque instance du problème est définie par des paramètres généraux :

- n : le nombre de techniciens disponibles,
- m : le nombre d'interventions à réaliser dans la journée,
- p : le nombre maximum de compétences différentes,
- *speed* : la vitesse moyenne de déplacement des techniciens.

On notera $I = \{1, \dots, n\}$ l'ensemble des techniciens, $J = \{1, \dots, m\}$ l'ensemble des interventions (*J* pour *Job*) et $K = \{1, \dots, p\}$ l'ensemble des compétences.

Chaque intervention $j \in J$ sera caractérisée par les données suivantes :

- des coordonnées (x_j, y_j) du lieu où doit avoir lieu l'intervention ;
- un numéro $C_j \in K$ de la compétence requise pour mener à bien l'intervention ;
- la durée D_j de l'intervention, c'est-à-dire, le temps nécessaire pour la mener à bien. Ce temps sera exprimé en minute entière ;
- une date d_j valant, soit un entier > 0 arbitraire (représentant, par exemple, le numéro de jour dans l'année), soit 0, si l'intervention est dite « sans rendez-vous » (elle peut être réalisée n'importe quand dans la journée, sans contrainte de fenêtre de temps). Cependant cette fonctionnalité ne sera pas exploitée dans ce projet ;
- un créneau $[t_j^{\min}, t_j^{\max}]$ qui donne la fenêtre de temps dans laquelle l'intervention doit impérativement débuter (où t_j^{\min} et t_j^{\max} sont également donnés sous forme d'entiers indiquant un numéro de minute dans la journée). Si l'intervention est « sans rendez-vous », il ne faut pas tenir compte de la fenêtre de temps, et cette fenêtre de temps sera donc de la forme $[0, 1439]$;
- une pénalité P_j à payer si l'on ne parvient pas à réaliser l'intervention. Cette pénalité devra alors être ajoutée à la fonction objectif, c'est-à-dire à la somme des longueurs des tournées exprimée en km.

Chaque technicien $i \in I$ est caractérisé par les données suivantes :

- des coordonnées (x_i, y_i) du lieu d'où part et où revient le technicien à la fin sa tournée journalière,
- les heures t_i^{start} et t_i^{end} (exprimée en minutes entières) où le technicien démarre et termine sa tournée. Plus précisément, le technicien i ne peut pas avoir d'activité, que ce soit un déplacement ou une intervention, ni avant t_i^{start} ni à partir de t_i^{end} inclu,
- la liste des compétences du technicien (chaque technicien a au moins une compétence et au plus p).

Chaque instance sera intégralement décrite dans un unique fichier texte dont le format est décrit en Section 6.

4 Méthodes Exactes

L'un des objectifs du projet est d'investiguer des méthodes de résolution exactes, d'en évaluer les limites et de pousser leur performances le plus loin possible. Pour ce faire, il est demandé d'imaginer, de mettre en oeuvre et de tester deux types d'approches :

Une approche « frontale » qui consiste à modéliser le problème dans son ensemble comme un problème linéaire en nombres entiers (PLNE).

- Question 1 : proposer un modèle *linéaire* en nombres entiers valide pour le problème (bien expliquer le rôle de chaque variable ainsi que son type, et la signification de chaque contrainte).
Un prérapport noté répondant à cette question est à rendre le 26 septembre 2011.
- Question 2 : utiliser le langage OPL avec CPLEX pour mettre en oeuvre ce modèle et tenter de résoudre les instances du benchmark.
Le code OPL sera à rendre le 3 octobre avant la séance de T.P.

Une approche par « relaxation lagrangienne » qui consiste à relâcher une ou plusieurs contraintes et tenter ainsi d'aller plus loin qu'avec la méthode frontale.

- Question 3 : écrire une relaxation lagrangienne des contraintes sur les fenêtres de temps. Décrire une procédure globale pour obtenir une borne inférieure du problème qui soit meilleure que la borne donnée par le relâché continu de la formulation initiale.
- Question 4 : mettre en oeuvre quelques itérations d'un algorithme d'optimisation du lagrangien (par exemple, algorithme de sous-gradient), en changeant manuellement la valeur du multiplicateur de Lagrange. Comparer les bornes obtenues avec la borne donnée par la relaxation continue du problème.
- Question 5 : L'objectif des méthodes employées est bien sûr de construire en définitive des solutions pour GoTIC. Dans cette perspective, quel est l'intérêt de la mise en oeuvre d'une relaxation lagrangienne par rapport à l'utilisation du relâché continu ? Y a-t-il moyen de construire une solution réalisable pour GoTIC à partir de la solution d'une relaxation lagrangienne ?

5 Méthodes Approchées

L'objectif de cette seconde partie est de mettre en oeuvre concrètement les heuristiques/métaheuristiques présentées en cours. L'idée est d'implanter certains des algorithmes en les adaptant au mieux au problème : adaptation de la représentation informatique de la solution, des paramètres propres aux algorithmes mis en jeu, etc... Le travail demandé se décompose en trois volets, le premier servira de brique de base aux deux autres qui sont indépendants.

5.1 Création d'un module de résolution « exacte »

L'objectif de ce module est de calculer de manière *exacte* la tournée optimale d'un technicien pour une liste *imposée* d'interventions¹. En entrée, il devra prendre une liste d'interventions et rendre en sortie un booléen indiquant si une tournée incluant l'ensemble des interventions est possible ou non et en cas de réponse positive, la liste optimale classée des interventions avec le coût associé. La méthode implémentée est laissée libre.

Le rapport devra décrire et justifier en détail la méthode choisie.

5.2 Approche « glouton »

L'objectif de ce second volet est de mettre en oeuvre un algorithme glouton permettant de construire une solution complète au problème posé. L'algorithme glouton qui exploitera le module développé précédemment devra être d'aussi bonne qualité que possible compte tenu des limites inhérentes à ce genre de méthode.

Le rapport devra en particulier détailler la stratégie qui aura été retenue pour le choix à chaque étape. Si plusieurs approches sont testées et comparées, elles pourront également être détaillées dans le rapport.

5.3 Approche « Voisinage »

Ce dernier volet doit permettre une approche globale permettant d'éviter les limitations propres à l'approche constructive du volet précédent. L'objectif est de mettre en oeuvre un algorithme de recuit simulé qui devra impérativement se baser sur le module développé précédemment (ou une variante) à charge de choisir une structure de voisinage adaptée qui permettra de l'utiliser efficacement.

1. Ce module n'est pas nécessairement destiné à être utilisé de manière autonome, mais servira de brique élémentaire pour les méthodes glouton et recuit. Cependant, pour des raisons pratiques, vous aurez probablement intérêt à prévoir un moyen pour faciliter sa mise en oeuvre (exécutable autonome ou options particulières de votre programme principal, ...).

Le rapport devra détailler la structure de voisinage retenue, les principaux paramètres du programme de refroidissement et les modalités d'utilisation du module exact. Les choix réalisés devront être motivés et justifiés autant que possible par des résultats expérimentaux.

5.4 Travail à réaliser et évaluation

Travail demandé :

- pour le 23 janvier 2012, chaque groupe devra remettre (par courriel à l'encadrement) les sources d'un programme permettant de lire un fichier d'instance dont le nom est passé en paramètre, et qui génère une solution au format requis. Cette solution (aléatoire, peu importe sa qualité) devra être acceptée par le validateur `gotic` fourni ;
- pour le 8 février, chaque groupe devra remettre par courriel le rapport final et le programme permettant de résoudre les instances proposées ;
- pour le 13 février jour de soutenance, une annexe comportant des corrections éventuelles ou des résultats complémentaires pourra être remise.

Le rapport devra présenter en détail chacune des trois parties :

- le module de calcul exact de la tournée optimale d'un technicien,
- l'algorithme glouton,
- l'algorithme de recuit simulé.

En plus de la description propre à chaque méthode, le rapport devra fournir un tableau récapitulatif des résultats obtenus sur les scénarios de test proposés ainsi qu'une analyse critique des résultats en proposant des axes d'amélioration (autres méthodes, hybridations, ...).

L'évaluation finale du projet tiendra compte :

- du programme fourni le 23/01 ;
- du travail et des résultats obtenus sur chacune des trois parties détaillées précédemment et évaluées indépendamment ;
- de la qualité globale du rapport incluant l'analyse critique ;
- de la qualité de présentation orale du projet (clarté et esprit de synthèse).

6 instance de test et format des données

les différentes instances du problème seront fournies sous la forme de fichiers textes au format suivant, dans lequel les lignes commençant par # sont à ignorer (commentaires) et les lignes vides sont autorisées :

```
# fichier gotic_t3_j10_c5.dat

gotic_instance gotic_t3_j10_c5
nbtic 3
nbjob 10
nbcmp 5
speed 50

#      id    x    y  t_start  t_end  cmp_list
tic   tic_1  11  21    480    1080    1 3 5
tic   tic_2  26  79    480    1080    1 2 3 4 5
tic   tic_3  55  67    480    1080    1 2 3 4 5

#      id    x    y  t_min  t_max  c    d    d    p
job   job_1  58  55    990    1079    5   30   1   1000
job   job_2  30  19    750    839    1   60   1   1000
job   job_3  41  45     0    1439    4   30   0   1000
job   job_4  58  24    840    1019    3   60   1   1000
job   job_5  16   9    570    689    2   60   1   1000
job   job_6  18  32    690    809    4   30   1   1000
job   job_7  89 100     0    1439    2   60   0   1000
job   job_8  57  75     0    1439    2   60   0   1000
job   job_9  89  60    600    779    1   90   1   1000
job   job_10 89   3    570    629    2   30   1   1000

end
```

Les résultats seront donnés par un fichier dont le format est indiqué ci-dessous. La section `DETAIL_SECTION` est facultative, mais vous sera utile pour déboguer votre programme !

```
# Fichier gotic_3_5_10.sol
# generee le ... par ...

#          nom_instance  total_cost
GOTIC_SOLUTION gotic_3_5_10  439

LENGTH TIC_1  0 km
LENGTH TIC_2  221 km
LENGTH TIC_3  218 km

CIRCUIT TIC_1
CIRCUIT TIC_2 JOB_3 JOB_5 JOB_6 JOB_2 JOB_4 JOB_1
CIRCUIT TIC_3 JOB_10 JOB_9 JOB_7 JOB_8

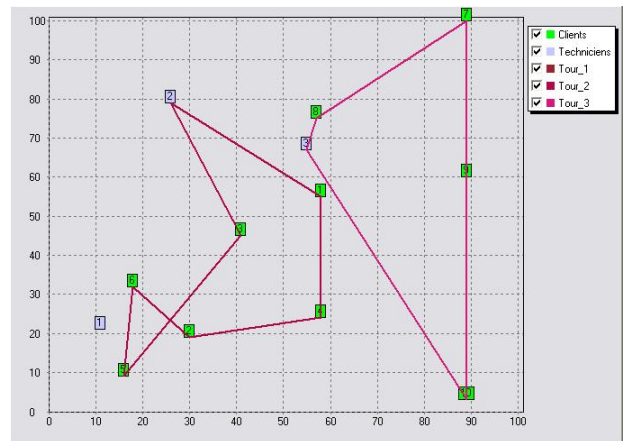
DETAIL_SECTION

TRAJET TIC_1  08:00; # depart base
TIC_1  08:00 # retour base 0km = 0mn
18:00; # fin journee, attente : 10:00

TRAJET TIC_2  08:00; # depart base
JOB_3  08:44 # arrivee sur site, 37km = 44mn
08:44 # debut intervention, pas de RdV
09:14; # fin intervention, duree = 00:30
JOB_5  10:07 # arrivee sur site, 44km = 53mn
10:07 # debut intervention, RdV respecte
11:07; # fin intervention, duree = 01:00
JOB_6  11:35 # arrivee sur site, 23km = 28mn
11:35 # debut intervention, RdV respecte
12:05; # fin intervention, duree = 00:30
JOB_2  12:27 # arrivee sur site, 18km = 22mn
12:30 # debut intervention, RdV respecte
13:30; # fin intervention, duree = 01:00
JOB_4  14:04 # arrivee sur site, 28km = 34mn
14:04 # debut intervention, RdV respecte
15:04; # fin intervention, duree = 01:00
JOB_1  15:41 # arrivee sur site, 31km = 37mn
16:30 # debut intervention, RdV respecte
17:00; # fin intervention, duree = 00:30
TIC_2  17:48 # retour base 40km = 48mn
18:00; # fin journee, attente : 00:12

TRAJET TIC_3  08:00; # depart base
JOB_10 09:26 # arrivee sur site, 72km = 86mn
09:30 # debut intervention, RdV respecte
10:00; # fin intervention, duree = 00:30
JOB_9  11:08 # arrivee sur site, 57km = 68mn
11:08 # debut intervention, RdV respecte
12:38; # fin intervention, duree = 01:30
JOB_7  13:26 # arrivee sur site, 40km = 48mn
13:26 # debut intervention, pas de RdV
14:26; # fin intervention, duree = 01:00
JOB_8  15:15 # arrivee sur site, 41km = 49mn
15:15 # debut intervention, pas de RdV
16:15; # fin intervention, duree = 01:00
TIC_3  16:25 # retour base 8km = 10mn
18:00; # fin journee, attente : 01:35

END
```



Enfin, un validateur est disponible. Son utilisation est décrite en tapant `gotic -h`. Ce programme vous permettra entre autre :

- d'extraire quelques données statistiques à partir des instances proposées,
- de convertir le format d'instance en un format opl compatible avec cplex,
- accessoirement de générer de nouvelles instances,
- mais surtout, de valider vos fichiers solutions et d'en générer la partie `DETAIL_SECTION`.

Précisons qu'une solution non acceptée par le validateur *ne sera pas prise en compte pour vos résultats*.