

# KIRO - 2ème édition Edition

Y. VANLAER      J. DARLAY      A. PARMENTIER

29 novembre 2018

Le sujet de ce Hackathon vous est proposé par LocalSolver. LocalSolver est un éditeur de logiciels dans le domaine de l'optimisation et de l'aide à la décision. Lancée sur le marché en 2012, la société est issue du Bouygues e-lab, département de R&D du Groupe Bouygues créé en 1994. L'équipe développe LocalSolver, un solveur d'optimisation mathématique innovant. En outre, elle développe des solutions sur-mesure d'optimisation aux interfaces ergonomiques pour des métiers variés. En forte croissance et rentable, la société compte parmi ses clients les plus grandes entreprises françaises et étrangères. LocalSolver vous propose un sujet issue d'un projet dans l'industrie des télécoms.

## 1 Problématique métier

Les opérateurs télécoms déploient des réseaux de fibres optiques pour raccorder les antennes de leur activité mobile à leur cœur de réseau. Ce réseau a pour but d'absorber l'augmentation de trafic liées aux technologies 4G et d'anticiper les besoins en 5G en remplaçant les vieux liens par des liens optiques plus efficaces. On parle alors d'un déploiement de réseau FTTA (Fiber To The Antenna). Les fibres sont tirées dans un réseau de fourreaux installé par l'opérateur historique. L'objectif de ce challenge est de proposer une architecture FTTA pour trois grandes agglomérations françaises : Grenoble, Nice et Paris.

La Figure 1 présente une agglomération à couvrir. Le cœur de réseau (ou DN pour Distribution Node) est représenté par le carré bleu, les antennes (Terminals) par les triangles roses et les fourreaux par les arcs bleus.

Afin de garantir la robustesse à la rupture d'un lien les architectures déployées sont idéalement sous forme de boucles. Cette architecture a un coût élevé et les opérateurs cherchent un compromis entre une architecture robuste à base de boucles et une architecture économe à base d'arbres. L'architecture cible pour ce challenge est à deux niveaux, des boucles structurantes sur laquelle il est possible d'ancrer des chaînes de collecte. Une boucle de distribution doit partir d'un point de distribution puis passe par un maximum de 30 antennes avant de revenir au point de distribution de départ. Une chaîne de collecte va partir d'une antenne appartenant à une boucle de distribution et va passer par un maximum de 5 autres antennes. La Figure 2 donne un exemple d'architecture avec une boucle structurante noire et 4 chaînes de collecte rouge.



FIGURE 1 – Exemple d'agglomération à couvrir

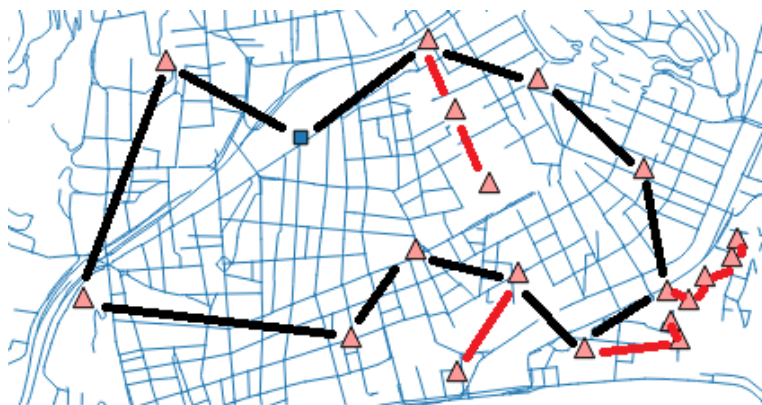


FIGURE 2 – Exemple d'architecture

Le coût d'une architecture est proportionnel à la longueur des fibres optiques déployées, on cherchera donc à minimiser la longueur totale de fibre optique utilisée pour déployer les trois villes en collectant toutes les antennes.

## 2 Description mathématique

Soit  $V^p$  l'ensemble des points de distributions,  $V^a$  l'ensemble des antennes, et  $V = V^p \cup V^a$ . Soit  $D = (V, A)$  le graphe orienté complet sur  $V$  :

$$A = \{(u, v) : u \in V, v \in V, u \neq v\}.$$

Soit  $\ell_a \in \mathbb{Z}_+$  la longueur de l'arête  $a$ . Les fourreaux n'étant pas symétriques, il est possible que  $\ell_{(v,u)} \neq \ell_{(u,v)}$ .

Une *boucle structurante*  $C$  est un circuit  $C$  dans  $D$  contenant au moins un sommet dans  $V^p$  et au plus  $k^{mb} = 30$  sommets dans  $V^a$ . En notant  $v_1, v_2, \dots, v_k, v_1$  la suite de sommets dans  $C$ , le coût de  $C$  est

$$c_C = \sum_{a \in C} \ell_a = \ell_{(v_k, v_1)} + \sum_{i=2}^k \ell_{(v_{i-1}, v_i)}.$$

Une *chaîne de collecte* pour une boucle structurante  $C$  est un chemin  $P$  dans  $D$  qui commence par un sommet de  $C$  et contient au plus  $k^{mc} = 5$  sommets qui ne sont pas dans  $C$ . En notant  $v_1, \dots, v_k$  les sommets de  $P$  (où  $v_1 \in C$  et  $k \leq k^{mc} + 1$ ), le coût de la chaîne de collecte  $P$  est

$$c_P = \sum_{a \in P} \ell_a = \sum_{i=2}^k \ell_{(v_{i-1}, v_i)}.$$

Un *réseau*  $r$  est composé d'une boucle structurante  $C^r$  et d'une collection  $\mathcal{P}^r$  de chaînes de collecte pour  $C^r$ . Un réseau  $r$  couvre une antenne  $v \in V^a$  si  $v$  est un sommet de  $C^r$  ou un sommet d'une chaîne  $P$  de  $\mathcal{P}^r$ . Plusieurs chaînes de collecte d'un même réseau peuvent se terminer sur le même sommet de la boucle structurante. Le nombre de chaînes de collectes dans un réseau n'est pas limité. Le coût d'un réseau est

$$c_r = c_{C^r} + \sum_{P \in \mathcal{P}^r} c_P.$$

Une *architecture* est une collection  $\mathcal{R}$  de réseaux  $r$  telle que chaque antenne de  $V^a$  est couverte par exactement un réseau  $r$  de  $\mathcal{R}$ . Le coût d'une architecture  $\mathcal{R}$  est

$$c_{\mathcal{R}} = \sum_{r \in \mathcal{R}} c_r.$$

Le problème consiste à trouver une architecture  $\mathcal{R}$  de coût minimum.

Les coordonnées géographiques de chaque sommet de  $V$  vous sont fournies. Nous insistons sur le fait que la distance  $\ell_{(u,v)}$  n'est pas la distance euclidienne entre  $u$  et  $v$ . Toutefois, les coordonnées des sommets peuvent fournir une information intéressantes car ces distances sont corrélées.

### 3 Format des instances et des solutions

Les données sont dans des dossiers correspondant à une des trois villes. Chaque dossier contient un fichier `nodes.csv` décrivant les sommets de l'instance (points de distribution et antennes) et un fichier `distances.csv` contenant la matrice des distances entre chaque paire de sommets.

Le fichier `nodes.csv` contient une ligne par sommet. L'index du sommet sur la première ligne est 0. Les deux premières colonnes donnent les coordonnées du sommet et la troisième son type : `distribution` (point de distribution) ou `terminal` (antennes). Voici un exemple de fichier `nodes.csv`.

```
5.76718673776955;45.1868105403962;distribution
5.81532994477435;45.2506180935299;distribution
5.72943382836614;45.1898920801397;terminal
5.75068308330930;45.2016577552734;terminal
```

Le fichier `distances.csv` contient une ligne par paire de sommets contenant la distance depuis le premier sommet vers le deuxième en utilisant les fourreaux. La première ligne contient la distance entre le premier sommet et lui-même, la seconde contient la distance entre le premier sommet et le second et ainsi de suite. Dans une instances avec  $n$  sommets, la distance  $\ell_{(u,v)}$  entre le  $i$ -ème sommet  $u$  et le  $j$ -ème sommet  $v$  est sur la  $(1 + n \times i + j)$ -ème ligne de `distances.csv`. Par exemple, voici le début du fichier `distances.csv` pour la même instance que le fichier `nodes.csv` ci-dessus.

```
0
33
15
24
32
0
17
etc .
```

Par exemple  $\ell_{(0,1)} = 33$ ,  $\ell_{(1,0)} = 32$  et  $\ell_{(1,2)} = 17$ .

Le format de sortie attendue est un fichier `.txt` par ville. Chaque ligne contient la liste ordonnée des sommets visités par une boucle structurante ou une chaîne de collecte. Une ligne correspondant à une *boucle structurante* commence par `b`. Une ligne correspondant à une *chaîne de collecte* commence par `c`. Nous rappelons que les sommets sont décrits par leur position dans le fichier `nodes.csv` en commençant à l'indice 0. La sortie attendue pour la solution simple présentée en Figure 3 s'écrit de la manière suivante.

```
b 0 1 2 5
c 2 3 4
```

Une autre solution à deux réseaux est

```
b 0 1 2
```

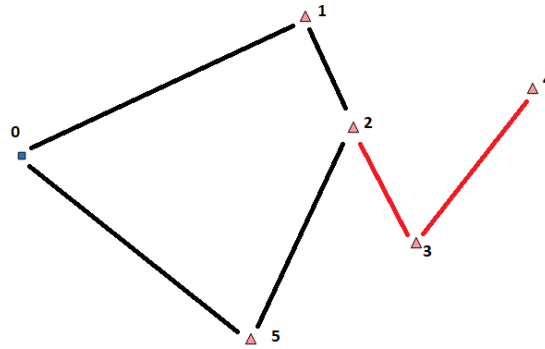


FIGURE 3 – Exemple de solution avec un seul réseau

c	2	4
b	0	3 5

## 4 Comment sera établi le classement

Le score d'une équipe est la somme des coûts des solutions proposées pour chaque instance. L'équipe avec le meilleur score, c'est à dire le score le plus petit, gagne.

## 5 Quelques conseils

Six heures, c'est très court. L'équipe qui gagnera sera celle qui arrivera à produire le plus vite des solutions décentes.

1. Partagez vous les tâches
2. Commencez immédiatement à coder les outils pour parser un fichier d'entrée et créer un fichier de sortie
3. Faites simple : il n'est pas difficile de construire une solution admissible. Commencez par coder des méthodes simples vous donnant des solutions plutôt bonnes, et évitez de faire un algorithme très puissant que vous n'arriverez pas à coder en 6h.