Présentation stage

Clément Legrand

June 29, 2018

Vehicle Routing Problem

Objectif

Construire un ensemble de tournées qui respectent les règles suivantes:

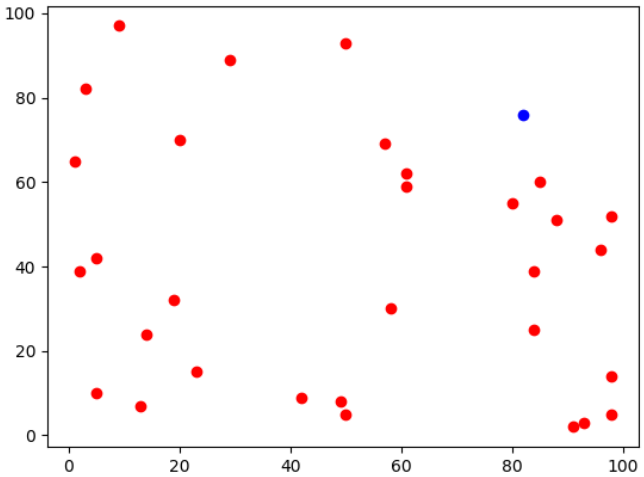
- Chaque client doit être desservi par une et une seule tournée;
- Chaque tournée doit partir et s'arrêter au dépôt;
- La longueur du réseau est minimale.

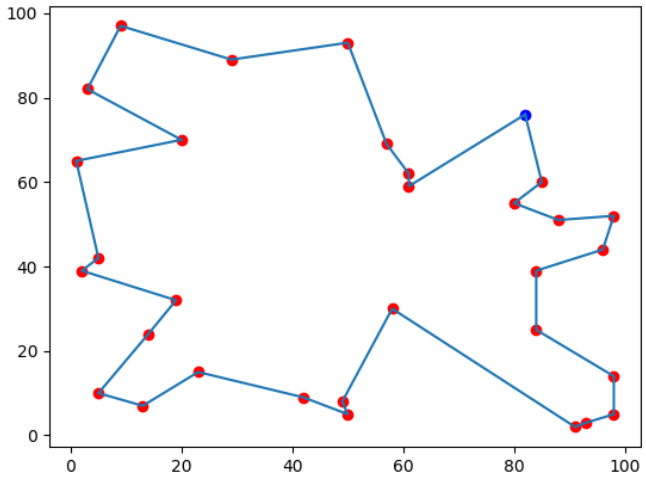
Données

Les instances utilisées sont celles de la littérature.
Pour chaque instance clients et dépôt sont définis.

Exemple

Instance A-n32-k05 de la littérature.





Ajout des demandes

Problème plus intéressant : gestion des demandes des clients et de la capacité des véhicules.

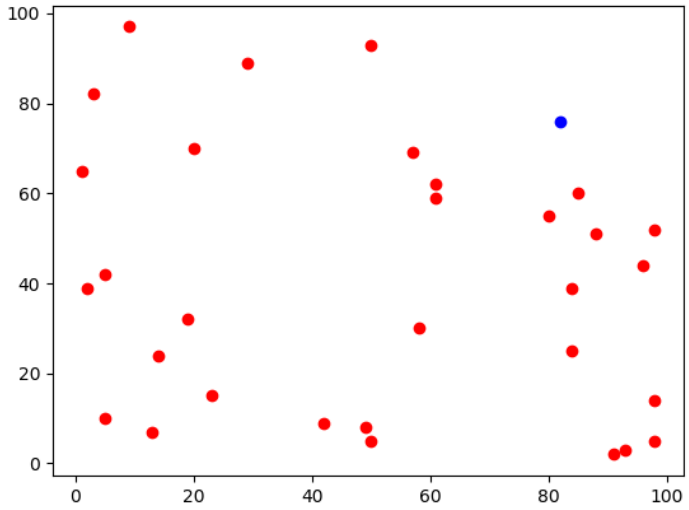
Nouvel objectif

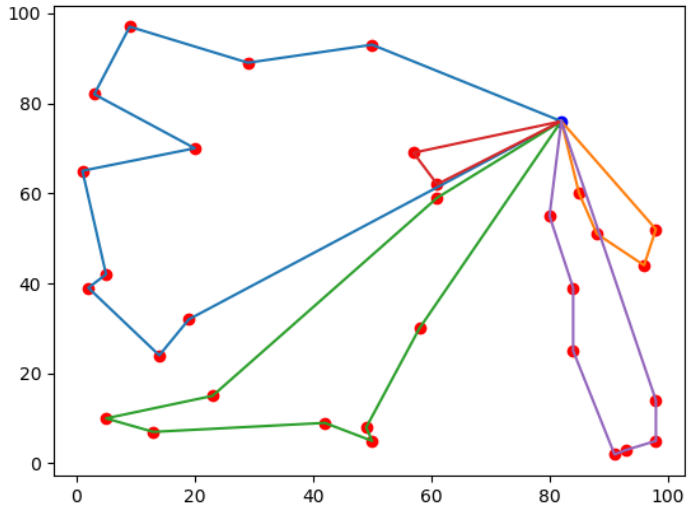
Une nouvelle règle vient s'ajouter aux règles précédentes:

- La demande totale sur une tournée ne doit pas excéder sa capacité.

Exemple

Reprenons A-n32-k05, en ajoutant les demandes des clients et une capacité limite:





Algorithme Clarke & Wright

Algorithme glouton, qui permet de trouver une solution initiale pas trop mauvaise.

Description

Pour chaque couple de clients (i, j) on calcule le saving de i et j avec:

$$s(i,j) = c_{i0} + c_{0j} - \lambda c_{ij} + \mu |c_{i0} - c_{0j}| + \nu \frac{d_i + d_j}{\overline{d}}$$

Tant que le saving maximal est positif:

- On prend (i, j) tel que $s(i, j)$ soit maximal;
- Les tournées qui contiennent i et j sont fusionnées (si possible);
- On met $s(i, j) = 0$.

Heuristique A & S

Heuristique développée par Arnold et Sörensen en 2017.
Intérêt: heuristique simple à mettre en place, et performante.

Algorithm 1: AS applique l'heuristique A& S au problème considéré

Input: L'instance considérée
Output: Une solution au problème I

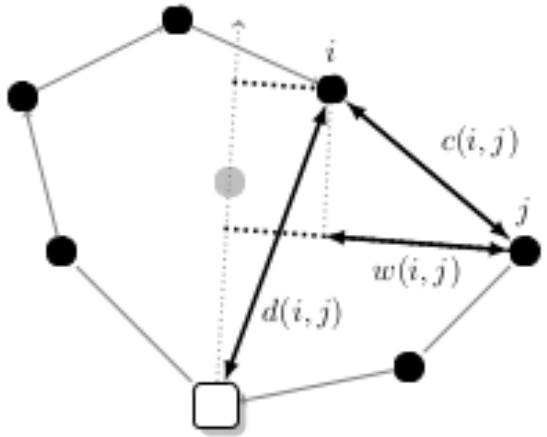
```
1 Sol ← ClarkeandWright
2 while Pas 3 minutes depuis la dernière amélioration do
3 |   Calcul de la pire arête
4 |   Application des opérateurs locaux
5 |   if Nouvelle meilleure solution then
6 |   |   Mettre à jour Sol
7 return Sol
```

Pire arête

Pire arête

La pire arête du graphe est l'arête (i,j) qui maximise la fonction:

$$b(i,j) = \frac{[\lambda_w w(i,j) + \lambda_c c(i,j)][\frac{d(i,j)}{max_{k,l} d(k,l)}]^{\frac{\lambda_d}{2}}}{1+p(i,j)}$$



Opérateurs locaux

Ejection-chain

Cet opérateur va essayer de déplacer au plus / clients sur des tournées plus adaptées.

Cross-exchange

Essaie d'échanger deux séquences de clients entre deux tournées.

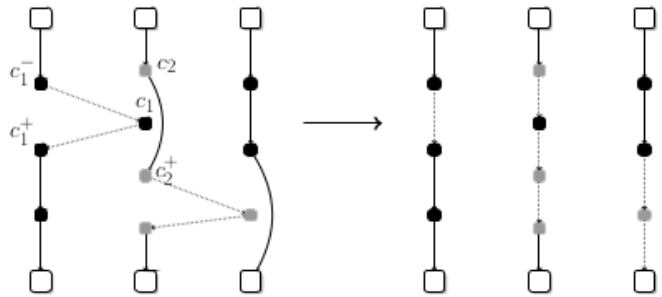


Figure 2: Illustration of the ejection chain with two relocations.

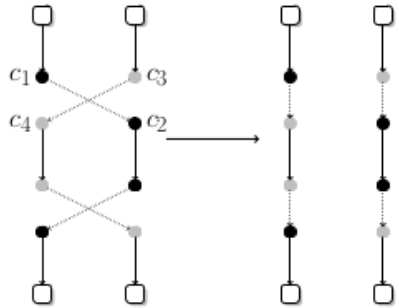
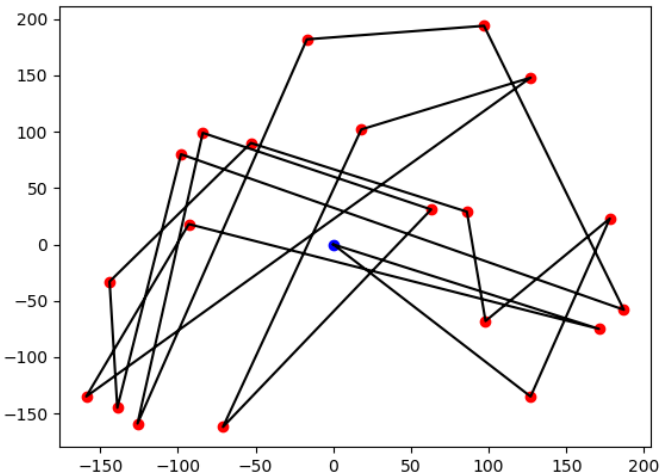


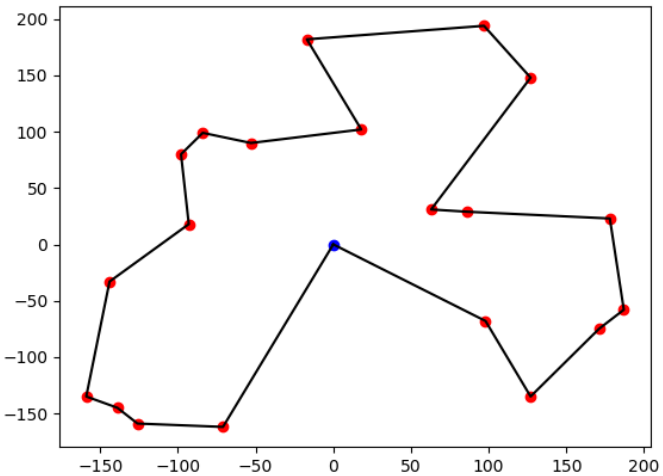
Figure 1: Illustration of the CROSS-exchange with sequences of two customers.

Opérateurs locaux

Lin-Kernighan

- Utilisé en général pour TSP;
- Optimisation intra-tournée (chaque tournée est améliorée indépendamment des autres).





Intégration de connaissances

Objectif

Améliorer les performances de l'heuristique en y intégrant de la connaissance.

Idée

Utiliser les résultats de CW pour tenter de prédire des arêtes qui appartiendront à la meilleure solution.

Description

L'algorithme utilisé s'inspire de l'heuristique A&S.

Modifications

- On applique LK à la solution initiale;
- Les opérateurs explorent les voisinages de manière aléatoire;
- On repart de la dernière meilleure solution obtenue, après N/2 itérations sans améliorations.

Résultats

Description

Création d'une base de solutions pour extraire des connaissances.

Protocole

- Génération d'une base de 100 solutions aléatoirement;
- On garde les solutions qui ont un coût inférieur à $c_{min} + (c_{max} - c_{min})\frac{10}{100}$: qualité privilégiée. On appelle cette base Qual_{10};
- Pour chaque arête (i,j) (resp (j,i) si j<i), on incrémente la valeur de MAT[i][j] (resp MAT[j][i]);
- On ne conserve que les $n/2$ premières arêtes dans la matrice.

Résultats

Description

Résultats