Option projet informatique, 2007
routage de véhicule avec fenêtres de temps (VRPTW)

Le problème.

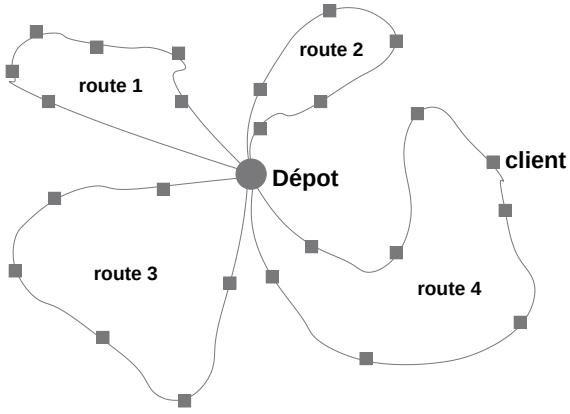
Le problème du routage de véhicules (VRP pour vehicule routing) consiste, étant donné un dépot D et un ensemble de commandes de clients $C = \{c_1, \ldots, c_n\}$, à délivrer toutes les commandes en optimisant divers paramètres. Pour çela on dispose d'une flotte de véhicules identiques, de capacité K : l'ensemble des commandes livrées lors d'une tournée par un véhicule donné doit tenir dans le véhicule (i.e. respecter sa capacité). Une commande est définie par

1. le client à qui elle est destinée,
2. le temps que dure la livraison **sur place** de la commande,
3. la capacité (on peut voir ça comme un poids) qu'elle utilise dans le véhicule.
4. la plage horaire dans laquelle s'effectue la livraison.

On dispose aussi d'une carte indiquant les liaisons et les distances entre les différents lieux géographiques, et les positions des clients.

Solution.

Une solution du problème est définie par un ensemble de tournées (ou *routes*). Une tournée correspond à une suite de commandes. Connaissant les distances à parcourir (les plus courts chemins entre les différents lieux géographiques), la vitesse du véhicule et les plages horaires, on en déduit
– les horaires de passages,
– les temps de trajets,
– les temps d'attente éventuels.



Pour que la tournée soit valide, la capacité du véhicule doit être suffisante pour contenir les commandes de la tournée, et les horaires de livraisons doivent respecter les plages horaires.

Objectifs.

L'objectif est de satisfaire tous les clients et, si c'est possible, de minimiser le nombre de véhicules utilisés. Dans un deuxième temps on s'attachera aussi à minimiser les distances parcourues par les véhicules et la durée totale d'utilisation des véhicules (cumul des durées des tournées).

Ce que l'on attends de vous.

Vous ferez le développement en JAVA. Les données du problèmes seront lues dans des fichiers texte (je vous donnerai un générateur aléatoire de problèmes).
D'un point de vue quantitatif, le projet est composé de quatre parties, et vous serez évalués sur chacune (elles apparaissent ici dans l'ordre dans lequel vous devriez les aborder) :

- **structures de données : à rendre au plus tard le 23 février 2007** (5 points dans la note finale) ; serviront à représenter les données du problème, les solutions, et les objets divers utilisés pour la recherche de solutions (tournées, horaires,...),
- **méthode glouton :** basée sur des règles de choix (faites appel au bon sens et à votre intuition), cette méthode fournira des solutions non optimales, mais aussi bonnes que possible (et dans un temps très court),
- **approche par recherche locale :** voir paragraphe suivant ; cette approche devrait donner de bons résultats,
- **interface graphique :** permettra de manipuler facilement les problèmes et de représenter graphiquement les solutions obtenues et leurs caractéristiques, voire l'évolution des solutions pendant la recherche.

Pour le développement d'un tel projet il est nécessaire d'être très rigoureux. D'un point de vue qualitatif, vous serez évalués sur

- l'originalité et l'efficacité des méthodes et des techniques,
- l'organisation du projet, la rigueur du développement, la lisibilité et la facilité de réutilisation du code.

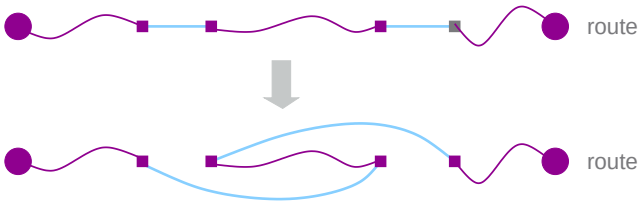
Approche par recherche locale.

Ces méthodes s'appliquent aux problèmes d'optimisation dont la taille ne permet de pas d'utiliser une méthode *complète* trop couteuse en temps (qui consiste souvent à énumérer toutes les possibilités).
Principe. Améliorer autant que possible une solution initiale (générée aléatoirement ou par une méthode glouton) en lui appliquant des modifications locales. En général on choisit la meilleure modification ou la première modification que l'on trouve qui permet d'améliorer la solution. Afin de ne pas rester confiné dans un minimum local, il est nécessaire de diversifier le choix des modifications que l'on applique, par exemple en faisant de temps à autre des modifications au hasard.
Parfois ces méthodes alternent des phases d'intensification (amélioration de la solution courante) avec des phases de diversification (changement plus importants permettant de changer de région dans l'espace de recherche).
Pour le problème du VRPTW il faudra peut-être considérer comme solutions des tournées qui ne respectent pas certaines contraintes, comme les plages horaires. En pénalisant ces solutions on permet de les parcourir momentanément pendant la recherche, sans les prendre vraiment en compte puisqu'une solution respectant les contraintes sera toujours méilleure.

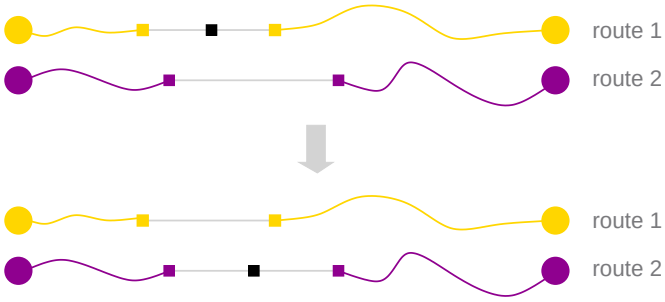
Quelques modifications élémentaires.

Nous vous suggérons d'utiliser les modifications qui suivent. La première correspond à une phase d'intensification : on cherche à améliorer chaque tournée indépendamment. Les deux autres correspondent à des chagngements plus profonds dans la structure de la solution. Les éjections se produisent souvent en cascade : un client éjecté d'une route t_1 vers une route t_2 est suivie d'une éjection de la route t_2 vers une route t_3,...

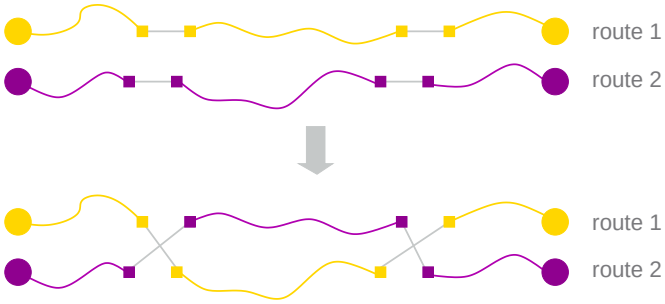
réarrangement :



éjection :



échange :



Modalités pratiques.

Une salle de machines vous est réservée trois créneaux par semaines : le jeudi matin de 8h00 à 12h00 et le vendredi de 14h00 à 16h00. Je serai présent sur place pour répondre à vos questions et suivre les projets le vendredi de 14h00 à 16h00.
Vous pouvez aussi me poser des questions par mail.

Annexe : syntaxe des fichiers de données.

Vous suivrez la syntaxe des problèmes de Solomon. Vous trouverez les instances de ces problèmes sur le web (25/50/100 clients).

```
<nom du probleme>
<nombre de vehicules> <capacite des vehicules>
<num client> <abcisse> <ordonnee> <poids> <debut plage> <fin plage> <duree livraison>
  ...
<num client> <abcisse> <ordonnee> <poids> <debut plage> <fin plage> <duree livraison>
```

Les distances représentent implicitement des durées.