

Un minimum local



Génération des candidats

Soit un ensemble de groupes de visites, une tournée candidate est une tournée existante à laquelle on peut :

- ne rien changer
- enlever un groupe
- ajouter un groupe
- enlever un groupe et ajouter un groupe

On obtient un problème de *set packing* classique, que l'on peut facilement résoudre avec un MILP.

Génération des groupes

Classification Ascendante Hiérarchique :

- Distance : intervalle de temps
- *Linkage* : *single*



Génération des groupes

Classification Ascendante Hiérarchique :

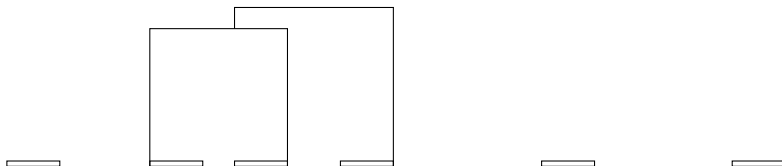
- Distance : intervalle de temps
- *Linkage* : *single*



Génération des groupes

Classification Ascendante Hiérarchique :

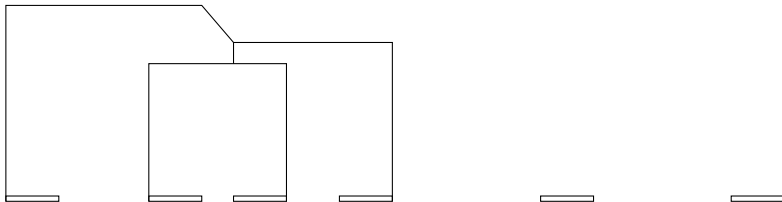
- Distance : intervalle de temps
- *Linkage* : *single*



Génération des groupes

Classification Ascendante Hiérarchique :

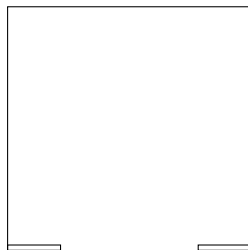
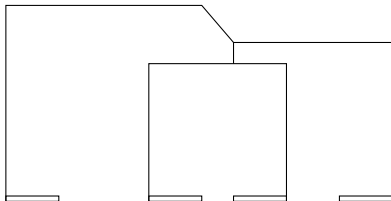
- Distance : intervalle de temps
- *Linkage* : *single*



Génération des groupes

Classification Ascendante Hiérarchique :

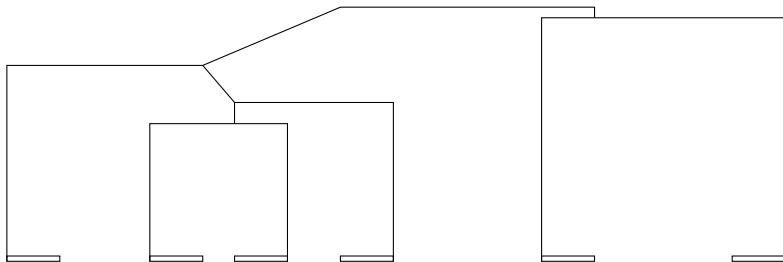
- Distance : intervalle de temps
- *Linkage* : *single*



Génération des groupes

Classification Ascendante Hiérarchique :

- Distance : intervalle de temps
- *Linkage* : *single*



Quelques mouvements inclus

Grâce aux groupes à une visite :

- échange de 2 visites entre 2 tours
- rotations de n visites entre n tours
- déplacement en chaîne de n visites sur $n + 1$ tours

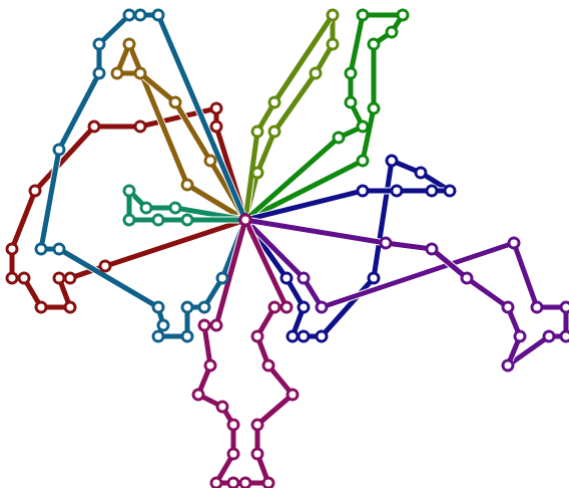
Grâce aux groupes contenant toutes les visites du tour :

- fusion de 2 tours en un seul

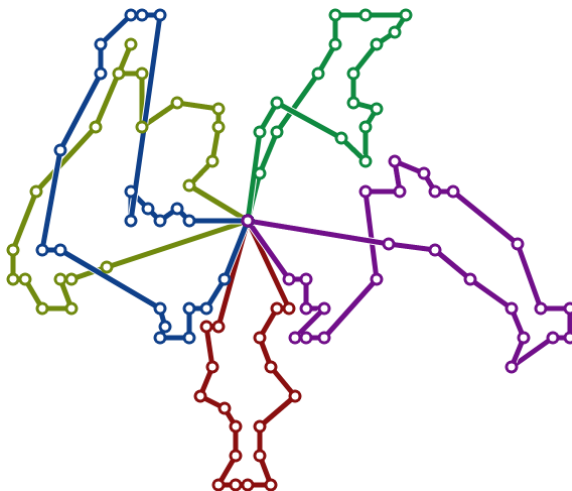
Grâce à la hiérarchie des groupes :

- répartitions des visites d'un tour sur plusieurs tours
- répartitions d'un grand nombre de visites d'un tour sur plusieurs tours
- des combinaisons de tous ça

VRPTW C203 : minimum local



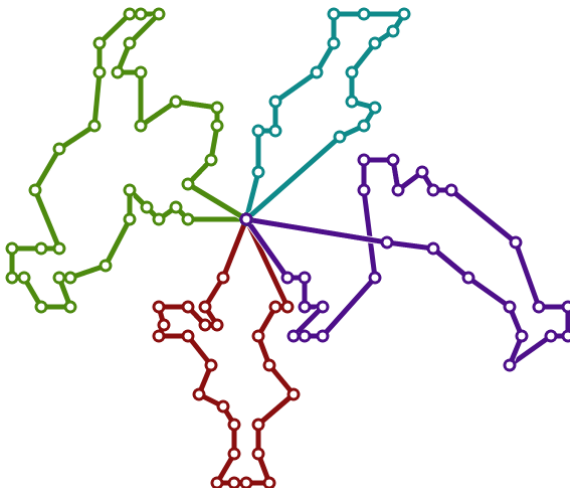
VRPTW C203 : itération 1



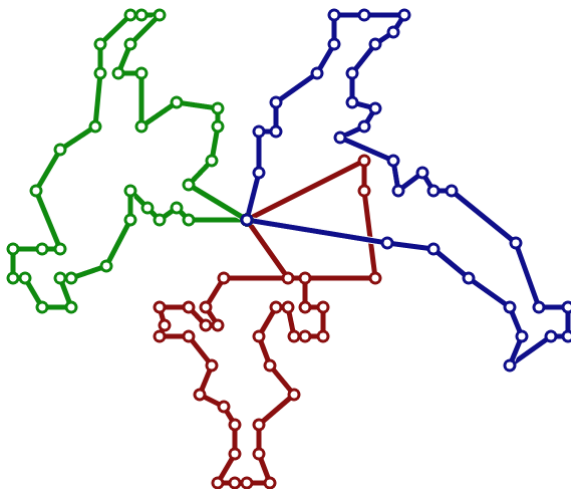
VRPTW C203 : itération 2



VRPTW C203 : itération 3



VRPTW C203 : itération 4



VRPTW C203 : itération 5

