APPLICATION CPLEX

SAE S6.01 EVOLUTION D'UNE APPLICATION EXISTANTE

Floris ROBART - Tran thai Duc DINH - Bryan Lemagnen

IUT Le Havre - 2023/2024 - BUT Informatique

Formulation mathématique

Modélisation Cplex

Interface de l'application

données du problème

- C est le nombre de clients
- N est l'ensemble de client, que $N = \{1, 2, ..., C\}$
- $Vertices$ est l'ensemble de vetices (ou les noeuds), que $Vertices = \{0\} \cup N$
- A est l'ensemble des arrêtes, que $A = \{(i, j) \in V^2 : i \neq j\}$
- $dist_{ij}$ est la distance de l'arrêtes $(i, j) \in A$
- Q est la capacité
- $demande_i$ est la demande de client $i \in N$
- V est le nombre des véhicules
- $Vehicles$ est l'ensemble de véhicules

Variable de décision

- x_{ijv} ,$i \in \{0, 1, 2, .., C\}, j \in \{0, 1, 2, .., C\}, v \in \{1, 2, .., V\}$,si le véhicule v passe directtement de node i à node j, x_{ijv}=1, sinon, $x_{ijv} = 0$
- $nbreVehiculeUtilise$ number of vehicles uses
- $u_i, i \in N$ variable to eliminant the subtours (sous-tours)

Contraintes du problème

- Il y a pas de chemin de node i vers lui même (2)
- Au moins 1 véhicules utilisés (3)
- Les véhicules si il quitte le dépôt, il rentrera au dépôt à la fin (4)&(5)
- Tous les véhicules quittent la node qu'il a visité (6)
- Chaque client est passé par exactement 1 fois (7)
- La capacité ne doit pas être dépassé (8)
- Les contraites pour éliminer les sous-tours (9)&(10)

Modélisation mathématique complète

$$
\begin{aligned}
\min \quad & \sum_{i,j \in A} dist_{ij} x_{ijv} & \\
\text{s.t.} \quad & x_{iiv} = 0 & \forall i \in Vertices, \forall v \in Vehicles \\
& \sum_{j \in N, v \in Vehicles} x_{1jv} = 1 & \\
& \sum_{j \in N} x_{j1v} <= 1 & \forall v \in Vehicles \\
& \sum_{j \in N} x_{j1v} = \sum_{j \in N} x_{1jv} & \forall v \in Vehicles \\
& \sum_{i \in Vertices} x_{jiv} = \sum_{i \in Vertices} x_{ijv} & \forall j \in Vertices, \forall v \in Vehicles \\
& \sum_{v \in Vehicles, i \in Vertices} x_{ijv} = 1 & \forall j \in N \\
& \sum_{i \in Vertices, j \in N} demande_j x_{ijv} \leqq Q & \forall v \in Vehicles \\
& u_j - u_i \geqq demande_j - Q(1 - x_{ijv}) & \forall i \in N, \forall j \in N, \forall v \in Vehicles, i \neq j \\
& demande_i \leqq u_i \leqq Q & \forall i \in N
\end{aligned}
$$

Modélisation CPlex

```
//Données
int nombreNode=...;
int Q=...; //capacité max des véhicules
int V=...; //Nombre de véhicules
float demande[2..nombreNode]=...; //Demandes des clients
float dist[1..nombreNode][1..nombreNode]=...;
dvar boolean x[1..nombreNode][1..nombreNode][1..V];
```

```
dvar boolean x[1..nombreNode][1..nombreNode][1..V];

minimize sum(i in 1..nombreNode,j in 1..nombreNode,v in 1..V ) dist[i][j]*x[i][j][v];
dvar int+ u[2..nombreNode]; // u est un variable pour éliminer les sous tours
```

```
// Il y a pas de chemin de node i vers lui même
    forall (i in 1..nombreNode, v in 1..V){
      x[i][i][v] == 0;
    }
    // Au moins 1 véhicule utlisé
     sum(j in 1..nombreNode,v in 1..V) x[1][j][v] >= 1;
    // Quitter dépôt, rentrer le dépôt à la fin
    forall (v in 1..V){
            sum(j in 2..nombreNode) x[j][1][v] == sum (j in 2..nombreNode) x[1][j][v];
            sum(j in 2..nombreNode) x[j][1][v] <= 1;
    }
    // Tous les véhicules Quitte la node qu'il a visiter - nombre de fois un node est quiité = nombre de fois un node est visité
    forall (j in 1..nombreNode,v in 1..V) {
      sum(i in 1..nombreNode)x[j][i][v] == sum (i in 1..nombreNode) x[i][j][v];
    }
    //Chaque client est "entré" par 1 fois
    forall (j in 2..nombreNode) {
      sum(v in 1..V, i in 1.. nombreNode) x[i][j][v] == 1;
     }
    //Capacité ne doit pas être dépassé
    forall (v in 1..V)
    {
            sum(i in 1..nombreNode, j in 2..nombreNode) demande[j]*x[i][j][v] <= Q;
    }
```

```
//Eliminer le sous-tours
forall (i in 2..nombreNode, j in 2..nombreNode, v in 1..V : i!=j)
{
  u[j]-u[i] >= demande[j] - Q*(1-x[i][j][v]);
}
forall (i in 2..nombreNode){
  demande[i] <= u[i];
  u[i] <= Q;
}
```

Floris ROBART
Tran thai Duc DINH
Bryan Lemagnen

Passons maintenant à l’interface graphique