

Projet RORT 2022

Optimisation du placement de fonctions de service dans les réseaux télécoms

Alain Faye – Massinissa Mérabet

Définition du problème

- Des commodités (des clients) veulent acheminer des données d'un point à un autre d'un réseau télécom
- Les données doivent être analysées
- Les données peuvent être analysées le long de leur parcours
- Les données sont analysées par des ressources logicielles (fonctions)
- Les ressources logicielles sont partageables entre les commodités

Définition du problème

- Graphe $G = (V, E)$ orienté
 - Les nœuds = machines , arc = communication possible entre 2 machines
 - Le temps de parcours d'un arc est une donnée du problème appelée latence (propre à chaque arc)
- Commodités doivent acheminer des données d'une source à un puit. Les données doivent avoir été analysées quand elles arrivent au puit. Le traitement peut se faire en cours de route de la source au puit
 - Une commodité k transfère une certaine quantité b_k de données (bande passante) d'un noeud s_k à un nœud t_k
 - Les données de la commodité doivent arriver à destination dans un délai fixé l_k (latence max)
 - Les données d'une commodité sont soumises à certains traitements et dans un ordre fixé

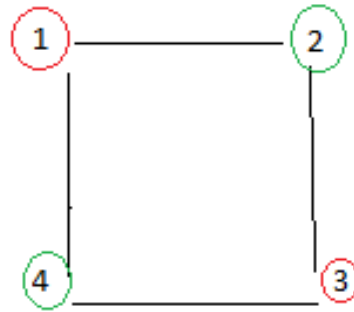
Définition du problème

- Des fonctions réalisent des traitements sur les données
 - Fonction f peut traiter une certaine quantité max. de donnée (capacité de f)
 - Les fonctions sont installées sur les nœuds du graphe
 - Installation d'une fonction f sur un nœud u induit un coût c_{fu}
 - On peut installer plusieurs fois la même fonction f sur un même nœud si la demande de traitement l'exige. Cependant un nœud u a une capacité limitée (nombre de fonctions max que l'on peut installer)
 - L'installation d'une première fonction sur un nœud u induit un coût d'ouverture propre au nœud u
- Certaines commodités peuvent imposer des contraintes d'exclusion entre fonctions installées sur un même nœud

Définition du problème

- Objectif minimiser le coût total
 - Coût total = somme des coûts d'activation des nœuds et des coûts d'installation des fonctions

Exemple



Latence = 1 sur chaque arc

2 commodités

- de 1 vers 3 quantité 20 latence 5 fonctions demandées 1, 2, 3
- de 2 vers 4 quantité 2 latence 3 fonctions demandées 2, 1

3 fonctions identiques

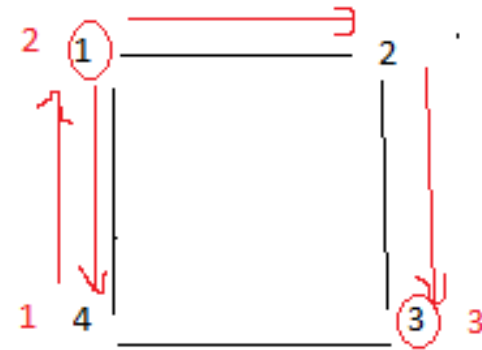
- capacité 10 coût installation par sommet = 1

coût d'ouverture d'un noeud = 1 nombre fonctions max par noeud = 3

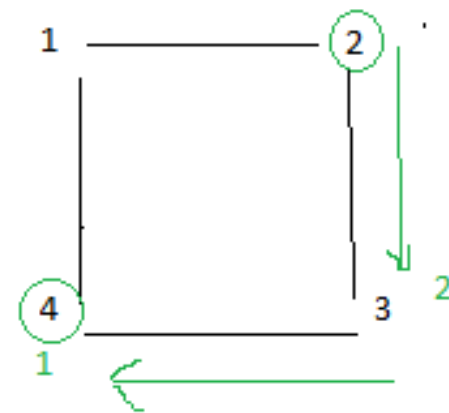
Exclusion

- pour les deux commodités, les fonctions 1 et 2 ne peuvent pas être installées sur le même noeud

Exemple



commodité 1 en rouge
elle utilise la fonction 1 au noeud 4
puis la fonction 2 au noeud 1
puis passe par le noeud 2 et
utilise la fonction 3 au noeud 3



commodité 2 en vert
elle utilise la fonction 2 au noeud 3
puis la fonction 1 au noeud 4

sur le noeud 1 sont installées 2 fonctions n°2
sur le noeud 3 sont installées 1 fonction n°2 et 2 fonctions n°3
sur le noeud 4 sont installées 3 fonctions n°1
coût total = 11

Fichiers de données

- Graph.txt
 - Une ligne d'entête
 - Puis le nombre de nœuds et le nombre d'arcs
 - Une ligne par arc
 - Pour chaque arc
 - extrémité initiale (attention les nœuds sont numérotés à partir de zéro)
 - extrémité terminale
 - le nombre de fonctions que peut accueillir le nœud initial
 - le nombre de fonctions que peut accueillir le nœud terminal
 - latence de l'arc

Graph.txt

Node_1 Node_2 Capacity_node1 Capacity_node2 Latency_arc node_cost

nb_nodes 4

nb_arcs 8

0 1 3 3 1

0 3 3 3 1

1 0 3 3 1

1 2 3 3 1

2 1 3 3 1

2 3 3 3 1

3 2 3 3 1

3 0 3 3 1

Fichiers de données

- Commodity.txt
 - Une ligne d'entête
 - Puis une ligne indiquant le nombre de commodités
 - Chaque ligne suivante décrit une commodité
 - Nœud de départ
 - Noeud d'arrivée
 - Quantité de données
 - Latence maximale

Commodity.txt

Source Destination Bandwidth Latency

nb_commodities 2

0 2 20 5

1 3 2 3

Fichiers de données

- Fct_Commod.txt
 - Une ligne par commande
 - Chaque ligne donne les fonctions dans l'ordre dans lequel elles doivent intervenir
 - Attention les fonctions sont numérotées à partir de zéro

Fct_Commod.txt

0 1 2

1 0

Fichiers de données

- Functions.txt
 - Une ligne d'entête
 - Puis une ligne indiquant le nombre de fonctions
 - Puis une ligne par fonction
 - Quantité de données que peut traiter la fonction (capacité)
 - Les coûts de placement sur chacun des nœuds du graphe

Functions.txt

capacity cost_by_node

nb_functions 3

10 1 1 1 1

10 1 1 1 1

10 1 1 1 1

Fichiers de données

- Affinity.txt
 - Une ligne par commodité
 - Chaque ligne indique un couple de fonctions qui ne peuvent être installées sur un même nœud pour cette commodité (attention fonctions numérotées à partir de zéro)
 - Si la ligne est vide cela signifie qu'il n'y a pas de contrainte d'exclusion pour cette commodité

Affinity.txt

0 1

1 0

Fichiers de données

- Le coût d'ouverture d'un nœud n'est pas, ou mal, indiqué dans le fichier Graph.txt
- On prendra un coût quelconque par exemple 1

Travail à faire

- Modèle mathématique linéaire
 - variables 0-1 et entières
 - Formulation en variables d'arcs
- Travail complémentaire au choix
 - Décomposition de Dantzig-Wolfe et variables de chemins
 - Recherche d'inégalités valides
 - Résolution heuristique , méta-heuristique
 - autre si proposition

Travail à faire

- Etude numérique
 - Borne inférieure : relaxation continue, décomposition de Dantzig-Wolfe
 - Borne supérieure : heuristique, méta-heuristique
- Instances disponibles
 - https://github.com/mouaciahlam/SNDLib_instances)
 - "di-yuan" et "pdh " les plus faciles
- Instances simples pour la mise au point
http://web4.ensiie.fr/~faye/mpro/MPRO_reseau/

Technologies

- Julia – JuMP
- Solveur au choix : GLPK , CPLEX, ...