

Manuel d'utilisateur

Projet M1 Androide

Bornes inférieures pour le

Partitionnement des Graphes

Encadrent: Viet Hung-Nguyen

Étudiants : Lucas Berterottière et Marc-Vincent Pereira

Mai 2018

1. Fichier Génération d'instances

`graphe_creux(n)` : Une première fonction qui prend en paramètre un nombre n de nœuds et renvoie un graphe pseudo-creux `networkx` comme expliqué en partie 2 du rapport.

`graphe_creux_dens(n,dens)` : La même fonction avec cette fois-ci la densité comme paramètre, qui est un nombre d'arêtes par rapport à celui du graphe complet.

`testCycle(n,dens,nb)` : cette fonction retourne la taille moyenne des cycles de la base des graphes de taille n , de densité dens .

`printTCycle(nmin,nmax,dens,prec)` : cette fonction utilise `testCycle` pour afficher la courbe de la taille moyenne des cycles de la base en fonction de n . Elle a permis de tracer la figure 2 du rapport.

`graphe_creux_dens_weight(n,dens)` : même fonction que `graphe_creux_dens` avec des arêtes de poids ± 1 comme décrit dans la partie 2.4 du rapport.

`graphe_creux_dens_weight2(n,dens)` : même fonction que `graphe_creux_dens` avec des arêtes de poids gaussiens comme décrit dans la partie 2.4 du rapport. Cette fonction sera utilisée pour la suite du projet.

`testNbArretes(int_min,int_max,prec,dens)` : cette fonction établit le nombre d'arêtes en fonction du nombre de sommet compris entre `int_min` et `int_max`.

`afficherTest(int_min,int_max,prec,dens)` : cette fonction nous permet d'afficher les résultats de `testNbArretes`. Elle nous a permis de tracer la figure 1 du rapport.

`afficher_graph_dens_weight(n,dens)` : cette fonction nous permet d'afficher des graphes pondérés. Elle nous a permis d'afficher les figures 7 et 8 du rapport.

`cycle_L(G,L)` : cette fonction nous permet de retirer les cycles de longueur supérieur à L de la base des cycles de G .

`cycle_vect(G,L)` : cette fonction nous permet d'obtenir une expression vectorielle des cycles de G .

2. Fichier ProgLineaire

Resolutionplne1 et resolutionplne2 sont les seules fonctions à appeler pour obtenir le résultat du programme linéaire.

resolutionplne1(G,k,param=True) : exécute le programme linéaire et affiche la solution trouvée, k est le nombre de partition voulue, G est le graphe dont on souhaite partitionner. Si param=True alors on résout le programme linéaire en variable entière sinon param=False et on résout le programme linéaire relaxé.

resolutionplne2(G,k,L,param=True) : identique à resolutionplne1 sauf qu'il y a un paramètre en plus L qui est la taille des cycles maximale que l'on prend dans la base de cycle.

testTime(nbmin,nbmax,dens,L) : affiche le temps comparé entre la version exacte du plne et la version relaxée en pl.

testResult(n,dens,Lmin,Lmax) : renvoie le résultat et le temps d'exécution du pl pour un graphe creux(n,dens) et différents L entre Lmin et Lmax.

test2(n,dens,Lmin,Lmax,prec) : affiche les résultats et le temps d'exécution du pl en fonction de L

test3(nmin,nmax,Lmin,Lmax,dens,prec) : affiche les résultats et le temps d'exécution du pl de différents L en fonction du nombre de nœuds

Callback est la fonction qui est appelée par le solveur à chaque solution du programme trouvée.

constructGraphe0(Node,listEdge,listEdgeValue) : Node le nombre de nœud du graphe, listEdge une liste des variable gurobi sur les arêtes (les x_{ij} pour (i,j) appartenant à E l'ensemble des arêtes.), listEdgeValue la liste des valeurs des variable x_{ij} . Cette fonction retourne un graphe networkx construit seulement avec les arêtes (i,j) tel que $x_{ij}=1$

contraintes(varEdges,varNode,m,k,listcycle=None) : varEdges les variables sur arêtes (x_{ij}), varNode les variables sur les sommets (x_i), m le modèle, k le nombre de partition à produire, listcycle la liste des cycle à utiliser pour produire les contraintes de cycle homogène. Cette fonction crée les contraintes du modèle.

objectif(var) : crée la fonction objectif en fonction des variables sur les arêtes.

creervar(G,m,param=True) : crée les variables pour le modèle m sur les sommets et les arêtes en fonction d'un graphe G si param=True les variables sont binaires sinon elles sont continues.