

Complexité et Calculabilité : projet 2020/2021

La guerre des protocoles IPv4/IPv6

Épisode II : L'attaque des algorithmes combinatoires

1 Evaluation du DM

Vous réaliserez *obligatoirement* le travail par trinômes (ou exceptionnellement des binômes). Vous déclarerez les trinômes sur la plateforme moodle¹ avant le 14/10/2020, en vous répartissant à *l'intérieur de votre demi-groupe de TD*.

Le devoir consistera en trois parties :

- Un document répondant aux questions présentées dans le présent sujet, en respectant les consignes listées plus bas. **La date du rendu est le 24 octobre**. Vous allez rendre le document sur la plateforme moodle du cours.
- Un programme réalisant la réduction du problème « La guerre des protocoles » vers un SAT-solver (z3). Un squelette de code, un parseur de graphes et un exemple d'utilisation de z3 vous seront fournis ultérieurement.
- Un rapport sur l'implémentation de votre réduction, détaillant son fonctionnement, ainsi que les éventuelles optimisations apportées et difficultés rencontrées.

2 DM Première partie : à rendre le 24/10/20

On rappelle qu'une instance du problème est un graphe non orienté $G = (V, E, t)$, V étant l'ensemble des sommets, E l'ensemble des liens, et

1. <https://moodle1.u-bordeaux.fr/course/view.php?id=6109>

$t : V \rightarrow \{\text{IPv4}, \text{IPv6}\}$ une fonction qui donne le type du sommet. L'ensemble H_t est celui des arêtes hétérogènes, l'ensemble H_m est celui des arêtes homogènes.

Consignes :

- Tous les algorithmes doivent être écrits en pseudo-code de haut niveau. Éviter les détails d'implémentation ;
- Vous aurez droit à une page pour répondre aux questions de la section 3, une demi-page pour la section 4, et une page pour la section 5 (sauf si vous faites la question bonus). Soyez donc concis et précis.

3 Réduction de la taille de l'instance

1. Est-ce qu'un sommet qui n'est incident à aucune arête hétérogène peut faire partie d'une solution optimale C ? Justifier très brièvement.
2. En vous basant sur ce constat, donner un algorithme qui prend en entrée $G = (V, E, t)$ et les ensembles H_t et H_m , et donne en sortie un graphe $G' = (V', E', t)$ (avec $V' \subseteq V$ mais pas forcément $E' \subseteq E$) plus petit tel qu'un ensemble $C \subseteq V$ est une solution optimale sur l'instance d'entrée G si et seulement si C est aussi une solution optimale de l'instance G' .
Indication : raisonner sur les composantes connexes du graphe $G_{cc} = (V, H_m)$ (ou de manière équivalente sur le graphe $G_{cc} = (V, E \setminus H_t)$). Vous constaterez que $H_t \subseteq E'$.
3. Donner et justifier la complexité de l'algorithme.

4 Algorithme exact "Brute Force"

A partir de maintenant, on considérera que le graphe de l'instance d'entrée est déjà réduit grâce à l'algorithme précédent. On notera donc directement $G = (V, E, t)$ le graphe réduit et on appliquera les algorithmes suivants sur ce graphe.

1. Donner un algorithme exact qui calcule une solution optimale C de taille minimum.
2. Calculer et justifier la complexité de cet algorithme (*spoiler* : il sera forcément exponentiel).

5 Algorithme Glouton

On veut maintenant concevoir un algorithme glouton qui donne une approximation de la solution de la taille optimale.

Soient X_1, X_2, \dots, X_k les composantes connexes du graphe $G_{cc} = (V, E \setminus H_t)$. Si on ajoute un lien hétérogène à ce graphe, on va fusionner au moins deux composantes connexes. L'idée est donc de mettre un convertisseur sur un sommet qui permettra de fusionner un maximum de composantes connexes en une fois.

On peut constater que si un sommet a un convertisseur, toutes ses arêtes incidentes deviennent « utilisables ». On peut donc les retirer de H_t et les ajouter à H_m .

1. Donner une fonction qui a en entrée G, H_t et H_m et donne en sortie le sommet qui est relié au maximum de composantes connexes différentes de la sienne.
2. Donner sa complexité.

L'algorithme va donc consister à ajouter un convertisseur à ce sommet à chaque itération, et recalculer les H_t, H_m et les composantes connexes (il est inutile d'écrire l'algorithme de calcul des composantes connexes, on considérera qu'il est déjà implémenté dans une fonction).

3. Donner un critère d'arrêt de cet algorithme glouton.
4. Calculer et justifier sa complexité.
5. **Question bonus difficile à laquelle il vaut mieux ne pas répondre que donner une réponse fausse** : Calculer et justifier un facteur d'approximation de cet algorithme.