Comparaison d'algorithmes pour le problème d'ensemble dominant minimum dans un graphe

Bouali Zakariae

Département d'Informatique
Université de Mons

21 juin, 2019

Introduction

Définition

Un **ensemble dominant** D d'un graphe $G = (V, E)$, est un sous-ensemble de sommets tel que chaque sommet de G est soit dans D, soit voisin d'un sommet de D.

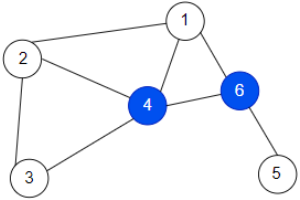


Figure: Graphe $G = (V, E)$.

Introduction

Définition

Un **ensemble dominant** D d'un graphe $G = (V, E)$, est un sous-ensemble de sommets tel que chaque sommet de G est soit dans D, soit voisin d'un sommet de D.

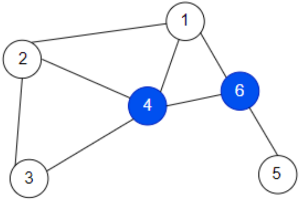


Figure: Graphe $G = (V, E)$.

Un **ensemble dominant minimum** (*Minimum Dominating Set, MDS*), est un ensemble dominant, contenant le plus petit nombre possible de sommets.

Introduction

Problème : ensemble dominant (*Dominating Set, DS*)

Entrée : un graphe $G = (V, E)$, un entier k.
Question : existe-t-il un ensemble dominant de taille au plus égale à k pour G?

Introduction

Problème : ensemble dominant (*Dominating Set, DS*)

Entrée : un graphe $G = (V, E)$, un entier k.
Question : existe-t-il un ensemble dominant de taille au plus égale à k pour G?

Applications

Introduction

Problème : ensemble dominant (*Dominating Set, DS*)

Entrée : un graphe $G = (V, E)$, un entier k.
Question : existe-t-il un ensemble dominant de taille au plus égale à k pour G?

Applications

- La synthèse de documents.

Introduction

Problème : ensemble dominant (*Dominating Set, DS*)

Entrée : un graphe $G = (V, E)$, un entier k.
Question : existe-t-il un ensemble dominant de taille au plus égale à k pour G?

Applications

- La synthèse de documents.
- Le problème de domination des reines.

Introduction

Problème : ensemble dominant (*Dominating Set, DS*)

Entrée : un graphe $G = (V, E)$, un entier k.
Question : existe-t-il un ensemble dominant de taille au plus égale à k pour G?

Applications

- La synthèse de documents.
- Le problème de domination des reines.
- Les réseaux des capteurs sans fil.

Introduction

Problème : ensemble dominant (*Dominating Set, DS*)

Entrée : un graphe $G = (V, E)$, un entier k.
Question : existe-t-il un ensemble dominant de taille au plus égale à k pour G?

Applications

- La synthèse de documents.
- Le problème de domination des reines.
- Les réseaux des capteurs sans fil.
- ...

Algorithmes exacts

Vue d'ensemble

Algorithmes exacts	Complexité en temps
Algorithme général (pour tous types de graphe) [1]	$O(1.93^n)$

Vue d'ensemble

Algorithmes exacts	Complexité en temps
Algorithme général (pour tous types de graphe) [1]	$O(1.93^n)$
Algorithme spécial pour les graphes de degré maximum égal à 3 [1]	$O(1.51^n)$

Vue d'ensemble

Algorithmes exacts	Complexité en temps
Algorithme général (pour tous types de graphe) [1]	$O(1.93^n)$
Algorithme spécial pour les graphes de degré maximum égal à 3 [1]	$O(1.51^n)$
Algorithme trivial basé sur le Set Cover [2]	$O(2^n)$
Algorithme amélioré basé sur le Set Cover [2]	$O(1.51^n)$

Set Cover

Définition

soit $U = \{x_1, x_2, \ldots, x_n\}$, un ensemble de n éléments, et soit $s = \{S_1, S_2, \ldots, S_m\}$ des sous-ensembles de U. Une couverture de U par un sous-ensemble de s est l'ensemble $I \subseteq \{1, 2, \ldots, m\}$ tel que $\bigcup_{S_{j \in I}} = U$.

Set Cover

Définition

soit $U = \{x_1, x_2, \ldots, x_n\}$, un ensemble de n éléments, et soit $s = \{S_1, S_2, \ldots, S_m\}$ des sous-ensembles de U. Une couverture de U par un sous-ensemble de s est l'ensemble $I \subseteq \{1, 2, \ldots, m\}$ tel que $\bigcup S_{j \in I} = U$.

Exemple

$$s = \{\{1,2,3\}, \{2,4,5\}, \{1,2,3,5\}, \{1,2,3,4\}\},\ U = \{1,2,3,4,5\}$$

Set Cover

Définition

soit $U = \{x_1, x_2, \ldots, x_n\}$, un ensemble de n éléments, et soit $s = \{S_1, S_2, \ldots, S_m\}$ des sous-ensembles de U. Une couverture de U par un sous-ensemble de s est l'ensemble $I \subseteq \{1, 2, \ldots, m\}$ tel que $\bigcup_{S_{j \in I}} = U$.

Exemple

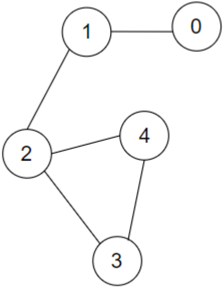
$$s = \{\{1, 2, 3\}, \{2, 4, 5\}, \{1, 2, 3, 5\}, \{1, 2, 3, 4\}\},\ U = \{1, 2, 3, 4, 5\}$$
$$I = \{1, 2\}$$

Réduction

La réduction d'une instance du problème de l'ensemble dominant minimum en une instance du problème de la couverture par ensembles.

Réduction

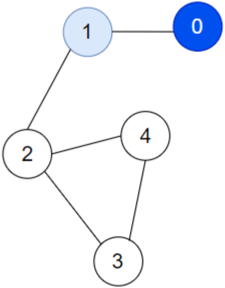
La réduction d'une instance du problème de l'ensemble dominant minimum en une instance du problème de la couverture par ensembles.



$$U = \{0, 1, 2, 3, 4\}$$

Réduction

La réduction d'une instance du problème de l'ensemble dominant minimum en une instance du problème de la couverture par ensembles.

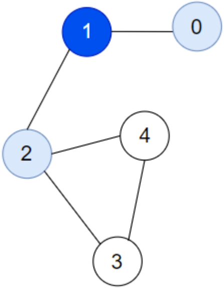


$U = \{0, 1, 2, 3, 4\}$

$s = \{\{0, 1\}\}$

Réduction

La réduction d'une instance du problème de l'ensemble dominant minimum en une instance du problème de la couverture par ensembles.

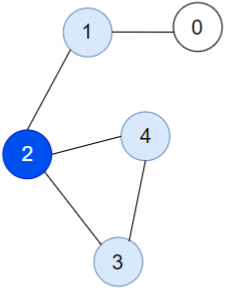


$U = \{0, 1, 2, 3, 4\}$

$s = \{\{0, 1\}, \{0, 1, 2\}\}$

Réduction

La réduction d'une instance du problème de l'ensemble dominant minimum en une instance du problème de la couverture par ensembles.

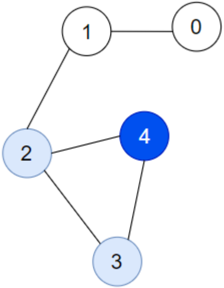


$U = \{0, 1, 2, 3, 4\}$

$s = \{\{0, 1\}, \{0, 1, 2\}, \{1, 2, 3, 4\}\}$

Réduction

La réduction d'une instance du problème de l'ensemble dominant minimum en une instance du problème de la couverture par ensembles.

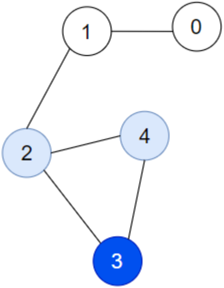


$U = \{0, 1, 2, 3, 4\}$

$s = \{\{0, 1\}, \{0, 1, 2\}, \{1, 2, 3, 4\}, \{2, 3, 4\}\}$

Réduction

La réduction d'une instance du problème de l'ensemble dominant minimum en une instance du problème de la couverture par ensembles.

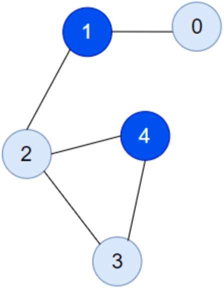


$U = \{0, 1, 2, 3, 4\}$

$s = \{\{0, 1\}, \{0, 1, 2\}, \{1, 2, 3, 4\}, \{2, 3, 4\},$
$\{2, 3, 4\}\}$

Réduction

La réduction d'une instance du problème de l'ensemble dominant minimum en une instance du problème de la couverture par ensembles.



$U = \{0, 1, 2, 3, 4\}$

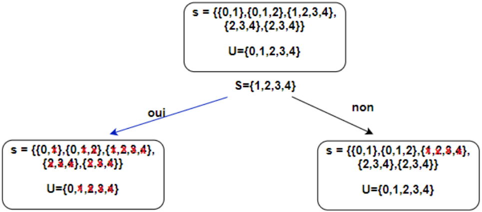
$s = \{\{0, 1\}, \{0, 1, 2\}, \{1, 2, 3, 4\}, \{2, 3, 4\}, \{2, 3, 4\}\}$

Algorithme trivial basé sur le Set Cover

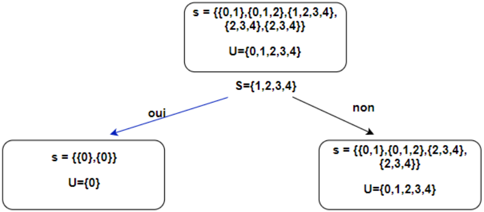
s = {{0,1},{0,1,2},{1,2,3,4},
{2,3,4},{2,3,4}}

U={0,1,2,3,4}

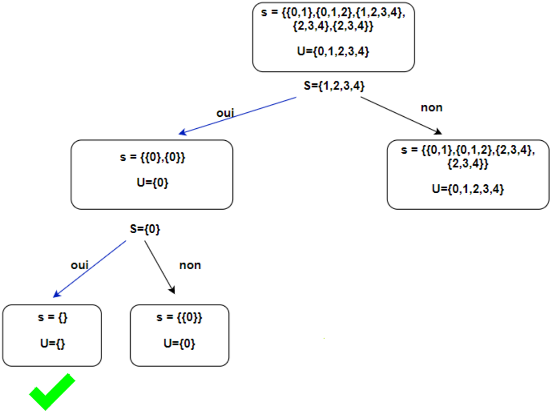
Algorithme trivial basé sur le Set Cover



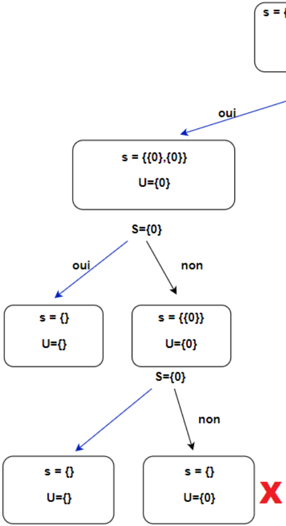
Algorithme trivial basé sur le Set Cover



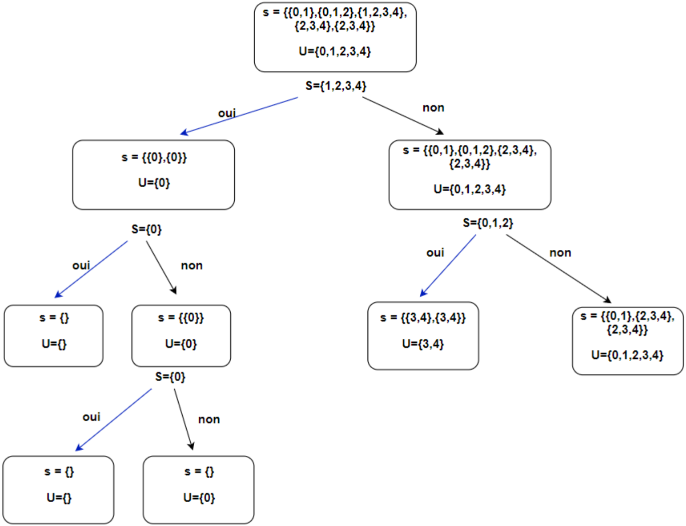
Algorithme trivial basé sur le Set Cover



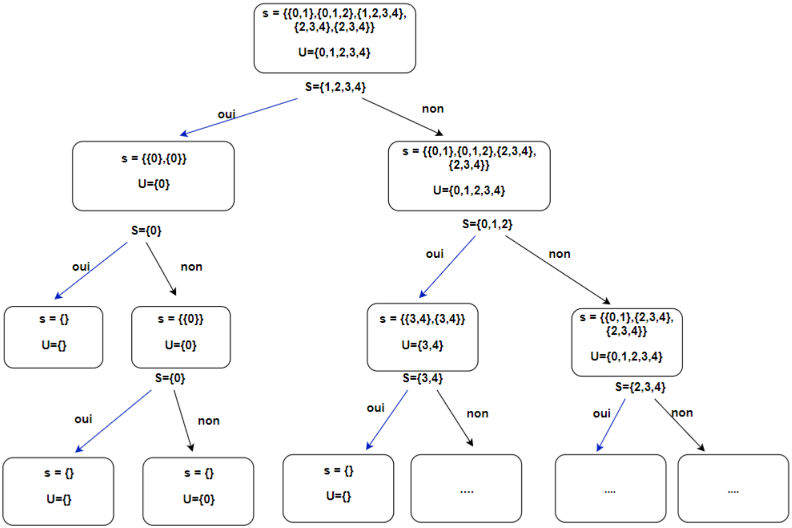
Algorithme trivial basé sur le Set Cover



Algorithme trivial basé sur le Set Cover



Algorithme trivial basé sur le Set Cover



Algorithme amélioré basé sur le Set Cover

Règle de l'élément unique

$$s = \{\{1,2\},\{1,2,3\},\{3,4\}\},\ U = \{1,2,3,4\}$$

Algorithme amélioré basé sur le Set Cover

Règle de l'élément unique

$$s = \{\{1,2\},\{1,2,3\},\{3,4\}\},\ U = \{1,2,3,4\}$$

$$\Downarrow$$

Algorithme amélioré basé sur le Set Cover

Règle de l'élément unique

$$s = \{\{1,2\},\{1,2,3\},\{3,4\}\},\ U = \{1,2,3,4\}$$

$$\Downarrow$$

$$s = \{\{1,2\},\{1,2\}\},\ U = \{1,2\}$$

Algorithme amélioré basé sur le Set Cover

Éliminer les sous-ensembles inclus

$$s = \{\{1,2,3\},\{1,2,3,5\},\{1,2,3,4\},\{2,4,5\}\},\ U = \{1,2,3,4,5\}$$

Algorithme amélioré basé sur le Set Cover

Éliminer les sous-ensembles inclus

$$s = \{\{1,2,3\},\{1,2,3,5\},\{1,2,3,4\},\{2,4,5\}\},\ U = \{1,2,3,4,5\}$$

X

Algorithme amélioré basé sur le Set Cover

Éliminer les sous-ensembles inclus

$$s = \{\cancel{\{1,2,3\}}, \{1,2,3,5\}, \{1,2,3,4\}, \{2,4,5\}\},\ U = \{1,2,3,4,5\}$$

$$\Downarrow$$

Algorithme amélioré basé sur le Set Cover

Éliminer les sous-ensembles inclus

$$s = \{\{1,2,3\},\{1,2,3,5\},\{1,2,3,4\},\{2,4,5\}\},\ U = \{1,2,3,4,5\}$$

X

$$\Downarrow$$

$$s = \{\{1,2,3,5\},\{1,2,3,4\},\{2,4,5\}\},\ U = \{1,2,3,4,5\}$$

Algorithme amélioré basé sur le Set Cover

Éliminer les sous-ensembles inclus

$$s = \{\cancel{\{1,2,3\}}, \{1,2,3,5\}, \{1,2,3,4\}, \{2,4,5\}\},\ U = \{1,2,3,4,5\}$$

$$\Downarrow$$

$$s = \{\{1,2,3,5\}, \{1,2,3,4\}, \{2,4,5\}\},\ U = \{1,2,3,4,5\}$$

Algorithme amélioré basé sur le Set Cover

Définition

Un couplage M de $G = (V, E)$ est un sous-ensemble des arêtes E deux à deux non adjacentes .

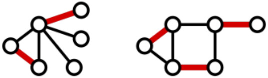


Figure: Couplages maximaux [3].

Algorithme amélioré basé sur le Set Cover

Définition

Une couverture par arêtes d'un graphe G est un ensemble d'arêtes C, tel que chaque sommet de G est incident à au moins une arête de C.

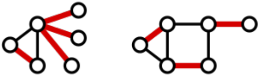
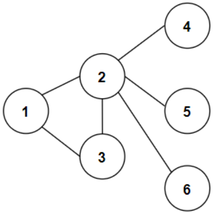
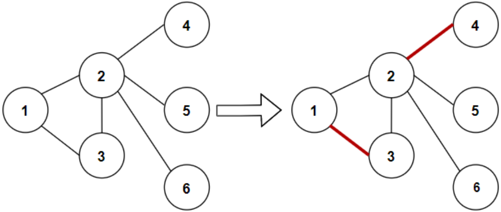


Figure: Couvertures par arêtes minimaux [4].

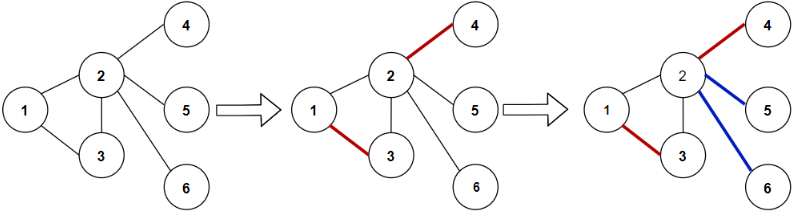
Algorithme amélioré basé sur le Set Cover



Algorithme amélioré basé sur le Set Cover



Algorithme amélioré basé sur le Set Cover



Algorithme amélioré basé sur le Set Cover

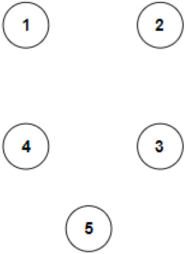
Arrêter quand la cardinalité des ensembles est au plus égale à deux

$$s = \{\{1,2\},\{2,3\},\{3,4\},\{3,5\},\{4,1\},\{4,5\}\},\ U = \{1,2,3,4,5\}$$

Algorithme amélioré basé sur le Set Cover

Arrêter quand la cardinalité des ensembles est au plus égale à deux

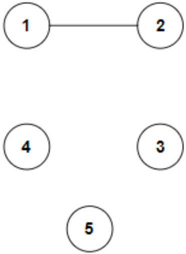
$$s = \{\{1,2\},\{2,3\},\{3,4\},\{3,5\},\{4,1\},\{4,5\}\},\ U = \{1,2,3,4,5\}$$



Algorithme amélioré basé sur le Set Cover

Arrêter quand la cardinalité des ensembles est au plus égale à deux

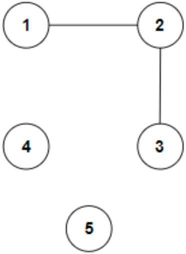
$$s = \{\{1,2\},\{2,3\},\{3,4\},\{3,5\},\{4,1\},\{4,5\}\},\ U = \{1,2,3,4,5\}$$



Algorithme amélioré basé sur le Set Cover

Arrêter quand la cardinalité des ensembles est au plus égale à deux

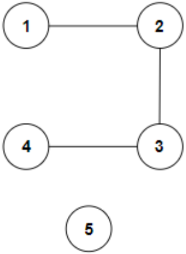
$$s = \{\{1,2\}, \{2,3\}, \{3,4\}, \{3,5\}, \{4,1\}, \{4,5\}\},\ U = \{1,2,3,4,5\}$$



Algorithme amélioré basé sur le Set Cover

Arrêter quand la cardinalité des ensembles est au plus égale à deux

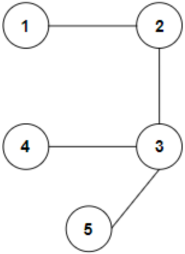
$$s = \{\{1,2\},\{2,3\},\{3,4\},\{3,5\},\{4,1\},\{4,5\}\},\ U = \{1,2,3,4,5\}$$



Algorithme amélioré basé sur le Set Cover

Arrêter quand la cardinalité des ensembles est au plus égale à deux

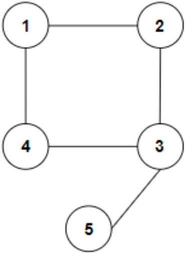
$$s = \{\{1,2\},\{2,3\},\{3,4\},\{3,5\},\{4,1\},\{4,5\}\},\ U = \{1,2,3,4,5\}$$



Algorithme amélioré basé sur le Set Cover

Arrêter quand la cardinalité des ensembles est au plus égale à deux

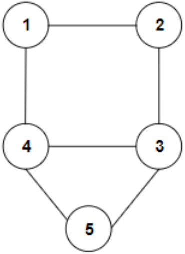
$$s = \{\{1,2\},\{2,3\},\{3,4\},\{3,5\},\{4,1\},\{4,5\}\},\ U = \{1,2,3,4,5\}$$



Algorithme amélioré basé sur le Set Cover

Arrêter quand la cardinalité des ensembles est au plus égale à deux

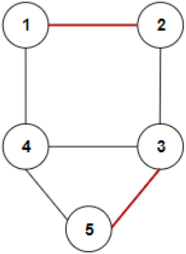
$$s = \{\{1,2\},\{2,3\},\{3,4\},\{3,5\},\{4,1\},\{4,5\}\},\ U = \{1,2,3,4,5\}$$



Algorithme amélioré basé sur le Set Cover

Arrêter quand la cardinalité des ensembles est au plus égale à deux

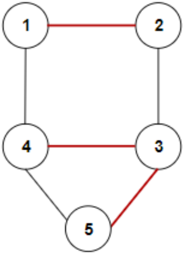
$$s = \{\{1,2\},\{2,3\},\{3,4\},\{3,5\},\{4,1\},\{4,5\}\},\ U = \{1,2,3,4,5\}$$



Algorithme amélioré basé sur le Set Cover

Arrêter quand la cardinalité des ensembles est au plus égale à deux

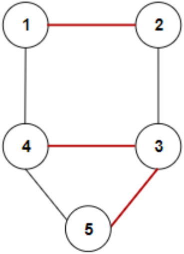
$$s = \{\{1,2\},\{2,3\},\{3,4\},\{3,5\},\{4,1\},\{4,5\}\},\ U = \{1,2,3,4,5\}$$



Algorithme amélioré basé sur le Set Cover

Arrêter quand la cardinalité des ensembles est au plus égale à deux

$$s = \{\{1,2\},\{2,3\},\{3,4\},\{3,5\},\{4,1\},\{4,5\}\},\ U = \{1,2,3,4,5\}$$



Algorithme amélioré basé sur le Set Cover

Illustration

$$U = \{0, 1, 2, 3, 4\},\ s = \{\{0, 1\}, \{0, 1, 2\}, \{1, 2, 3, 4\}, \{2, 3, 4\}, \{2, 3, 4\}\}$$

Algorithme amélioré basé sur le Set Cover

Illustration

$$U = \{0,1,2,3,4\},\ s = \{\{0,1\},\{0,1,2\},\{1,2,3,4\},\{2,3,4\},\{2,3,4\}\}$$

$$\Downarrow \text{Éliminer les sous-ensembles inclus}$$

Algorithme amélioré basé sur le Set Cover

Illustration

$$U = \{0,1,2,3,4\},\ s = \{\{0,1\},\{0,1,2\},\{1,2,3,4\},\{2,3,4\},\{2,3,4\}\}$$

$$\Big\Downarrow \text{ Éliminer les sous-ensembles inclus}$$

$$U = \{0,1,2,3,4\},\ s = \{\{0,1,2\},\{1,2,3,4\},\{2,3,4\},\{2,3,4\}\}$$

Algorithme amélioré basé sur le Set Cover

Illustration

$$U = \{0,1,2,3,4\},\ s = \{\{0,1\},\{0,1,2\},\{1,2,3,4\},\{2,3,4\},\{2,3,4\}\}$$

$$\Big\Downarrow \text{ Éliminer les sous-ensembles inclus}$$

$$U = \{0,1,2,3,4\},\ s = \{\{0,1,2\},\{1,2,3,4\},\{2,3,4\},\{2,3,4\}\}$$

$$\Big\Downarrow \text{ Règle d'élément unique}$$

Algorithme amélioré basé sur le Set Cover

Illustration

$$U = \{0,1,2,3,4\},\ s = \{\{0,1\},\{0,1,2\},\{1,2,3,4\},\{2,3,4\},\{2,3,4\}\}$$

$$\Big\Downarrow \text{ Éliminer les sous-ensembles inclus}$$

$$U = \{0,1,2,3,4\},\ s = \{\{0,1,2\},\{1,2,3,4\},\{2,3,4\},\{2,3,4\}\}$$

$$\Big\Downarrow \text{ Règle d'élément unique}$$

$$U = \{3,4\},\ s = \{\{3,4\},\{3,4\},\{3,4\}\}$$

Algorithme amélioré basé sur le Set Cover

Illustration

$$U = \{0, 1, 2, 3, 4\},\ s = \{\{0, 1\}, \{0, 1, 2\}, \{1, 2, 3, 4\}, \{2, 3, 4\}, \{2, 3, 4\}\}$$

$$\Downarrow \text{ Éliminer les sous-ensembles inclus}$$

$$U = \{0, 1, 2, 3, 4\},\ s = \{\{0, 1, 2\}, \{1, 2, 3, 4\}, \{2, 3, 4\}, \{2, 3, 4\}\}$$

$$\Downarrow \text{ Règle d'élément unique}$$

$$U = \{3, 4\},\ s = \{\{3, 4\}, \{3, 4\}, \{3, 4\}\}$$

$$\Downarrow \text{ Éliminer les sous-ensembles inclus}$$

Algorithme amélioré basé sur le Set Cover

Illustration

$$U = \{0,1,2,3,4\},\ s = \{\{0,1\},\{0,1,2\},\{1,2,3,4\},\{2,3,4\},\{2,3,4\}\}$$

$$\Downarrow \text{ Éliminer les sous-ensembles inclus}$$

$$U = \{0,1,2,3,4\},\ s = \{\{0,1,2\},\{1,2,3,4\},\{2,3,4\},\{2,3,4\}\}$$

$$\Downarrow \text{ Règle d'élément unique}$$

$$U = \{3,4\},\ s = \{\{3,4\},\{3,4\},\{3,4\}\}$$

$$\Downarrow \text{ Éliminer les sous-ensembles inclus}$$

$$U = \{3,4\},\ s = \{\{3,4\}\}$$

Algorithme amélioré basé sur le Set Cover

Illustration

$$U = \{0, 1, 2, 3, 4\},\ s = \{\{0, 1\}, \{0, 1, 2\}, \{1, 2, 3, 4\}, \{2, 3, 4\}, \{2, 3, 4\}\}$$

$$\Downarrow \text{ Éliminer les sous-ensembles inclus}$$

$$U = \{0, 1, 2, 3, 4\},\ s = \{\{0, 1, 2\}, \{1, 2, 3, 4\}, \{2, 3, 4\}, \{2, 3, 4\}\}$$

$$\Downarrow \text{ Règle d'élément unique}$$

$$U = \{3, 4\},\ s = \{\{3, 4\}, \{3, 4\}, \{3, 4\}\}$$

$$\Downarrow \text{ Éliminer les sous-ensembles inclus}$$

$$U = \{3, 4\},\ s = \{\{3, 4\}\}$$

$$\Downarrow \text{ Règle d'élément unique}$$

Algorithme amélioré basé sur le Set Cover

Illustration

$$U = \{0,1,2,3,4\},\ s = \{\{0,1\},\{0,1,2\},\{1,2,3,4\},\{2,3,4\},\{2,3,4\}\}$$

$$\Big\Downarrow \text{ Éliminer les sous-ensembles inclus}$$

$$U = \{0,1,2,3,4\},\ s = \{\{0,1,2\},\{1,2,3,4\},\{2,3,4\},\{2,3,4\}\}$$

$$\Big\Downarrow \text{ Règle d'élément unique}$$

$$U = \{3,4\},\ s = \{\{3,4\},\{3,4\},\{3,4\}\}$$

$$\Big\Downarrow \text{ Éliminer les sous-ensembles inclus}$$

$$U = \{3,4\},\ s = \{\{3,4\}\}$$

$$\Big\Downarrow \text{ Règle d'élément unique}$$

$$U = \{\},\ s = \{\}$$

Algorithme amélioré basé sur le Set Cover

Illustration

$$U = \{0,1,2,3,4\},\ s = \{\{0,1\},\{0,1,2\},\{1,2,3,4\},\{2,3,4\},\{2,3,4\}\}$$

$$\Downarrow \text{ Éliminer les sous-ensembles inclus}$$

$$U = \{0,1,2,3,4\},\ s = \{\{0,1,2\},\{1,2,3,4\},\{2,3,4\},\{2,3,4\}\}$$

$$\Downarrow \text{ Règle d'élément unique}$$

$$U = \{3,4\},\ s = \{\{3,4\},\{3,4\},\{3,4\}\}$$

$$\Downarrow \text{ Éliminer les sous-ensembles inclus}$$

$$U = \{3,4\},\ s = \{\{3,4\}\}$$

$$\Downarrow \text{ Règle d'élément unique}$$

$$U = \{\},\ s = \{\},\ I = \{2,3\}$$

Algorithmes approchés

Vue d'ensemble

Algorithmes exacts	Complexité en temps
Greedy	$O(n^2)$ [5]
Greedy random	$O(n^2)$ [5]

Vue d'ensemble

Algorithmes exacts	Complexité en temps
Greedy	$O(n^2)$ [5]
Greedy random	$O(n^2)$ [5]
Greedy reverse	$O(n^2)$ [5]

Vue d'ensemble

Algorithmes exacts	Complexité en temps
Greedy	$O(n^2)$ [5]
Greedy random	$O(n^2)$ [5]
Greedy reverse	$O(n^2)$ [5]
Algorithme génétique	$O(n^2)$ [6]

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

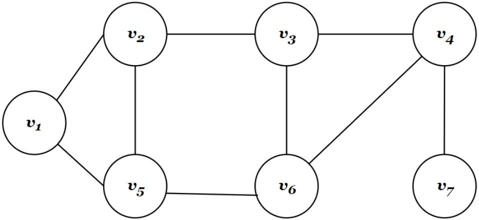


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

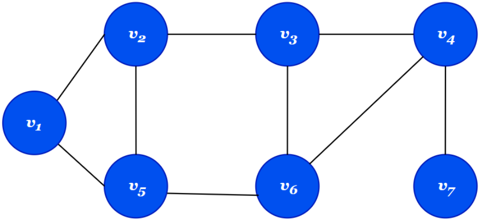


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

$$D = \{v_7, v_1, v_5, v_6, v_2, v_3, v_4\}$$
$$\uparrow$$

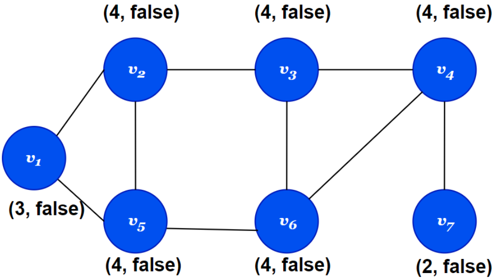


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

$$D = \{v_7, v_1, v_5, v_6, v_2, v_3, v_4\}$$
$$\uparrow$$

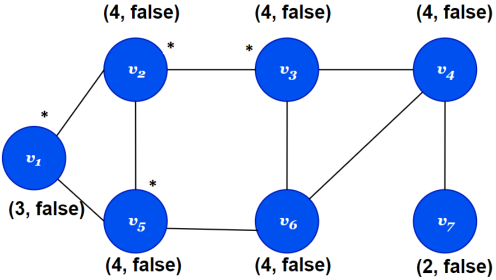


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

$$D = \{v_7, v_1, v_5, v_6, v_2, v_3, v_4\}$$
$$\uparrow$$

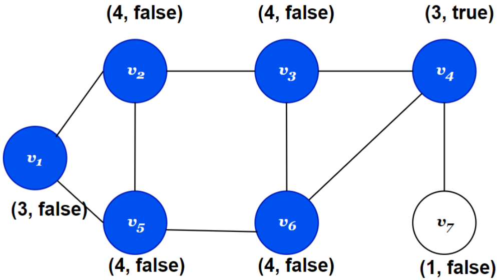


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

$$D = \{\cancel{v_7}, v_1, v_5, v_6, v_2, v_3, v_4\}$$
$$\uparrow$$

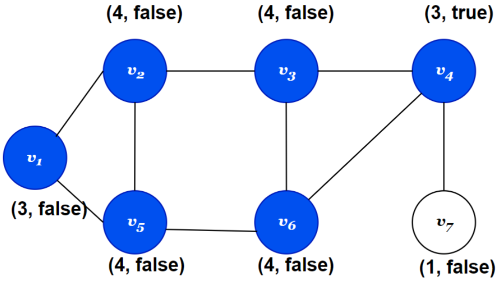


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

$$D = \{\cancel{v_7}, v_1, v_5, v_6, v_2, v_3, v_4\}$$
$$\uparrow$$

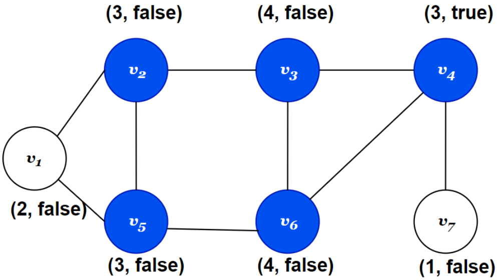


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

$$D = \{\cancel{v_7}, \cancel{v_1}, v_5, v_6, v_2, v_3, v_4\}$$
$$\uparrow$$

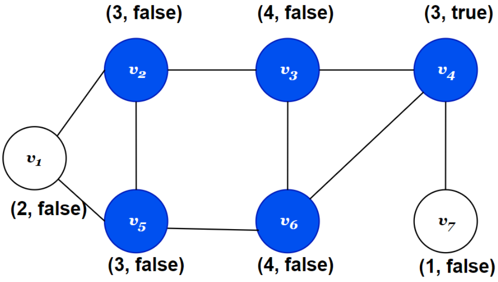


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

$$D = \{\cancel{v_7}, \cancel{v_1}, v_5, v_6, v_2, v_3, v_4\}$$
$$\uparrow$$

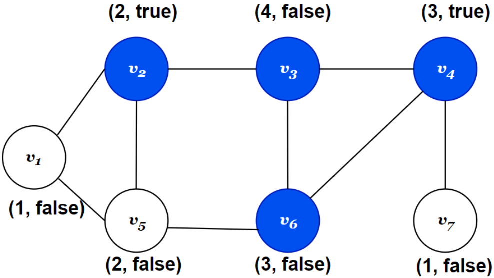


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

$$D = \{\cancel{v_7}, \cancel{v_1}, \cancel{v_5}, v_6, v_2, v_3, v_4\}$$
$$\uparrow$$

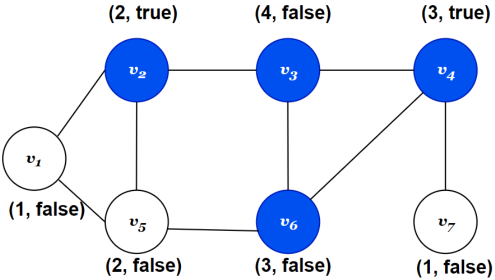


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

$$D = \{\cancel{v_7}, \cancel{v_1}, \cancel{v_5}, v_6, v_2, v_3, v_4\}$$

$$\uparrow$$

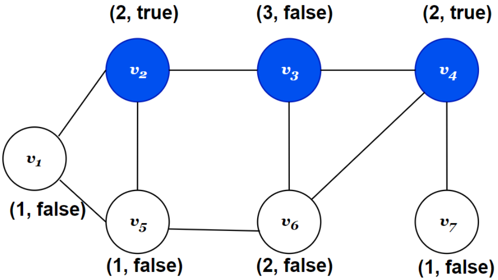


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

$$D = \{\cancel{v_7}, \cancel{v_1}, \cancel{v_5}, \cancel{v_6}, v_2, v_3, v_4\}$$

$$\uparrow$$

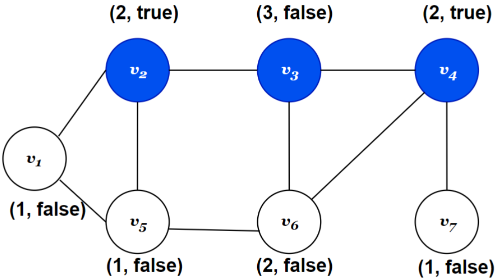


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

$$D = \{\cancel{v_7}, \cancel{v_1}, \cancel{v_5}, \cancel{v_6}, v_2, v_3, v_4\}$$

$$\uparrow$$

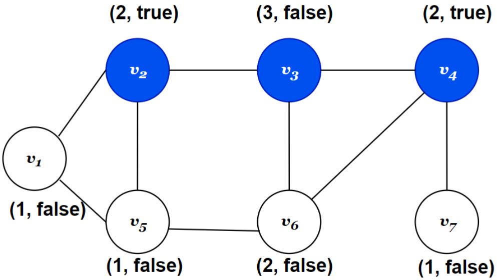


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

$$D = \{\cancel{v_7}, \cancel{v_1}, \cancel{v_5}, \cancel{v_6}, v_2, \cancel{v_3}, v_4\}$$
$$\uparrow$$

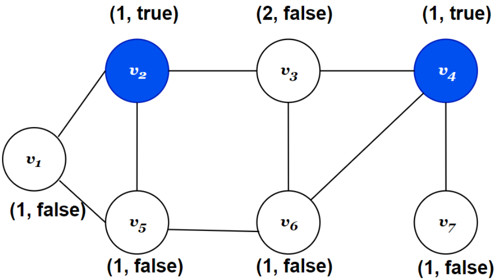


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

$$D = \{\cancel{v_7}, \cancel{v_1}, \cancel{v_5}, \cancel{v_6}, v_2, \cancel{v_3}, v_4\}$$

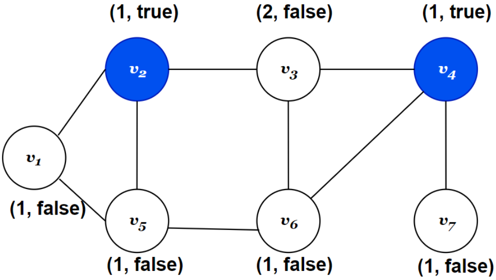


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

$$D = \{\cancel{v_7}, \cancel{v_1}, \cancel{v_5}, \cancel{v_6}, v_2, \cancel{v_3}, v_4\}$$

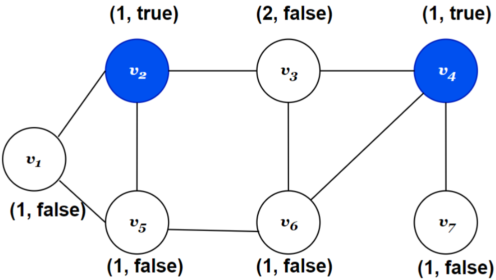


Figure: Algorithme greedy reverse.

Greedy reverse

Idée : au départ, l'ensemble D contient tous les sommets du graphe, et à chaque itération on enlève un sommet de D à condition que D reste toujours un ensemble dominant.

Illustration

$$D = \{v_2, v_4\}$$

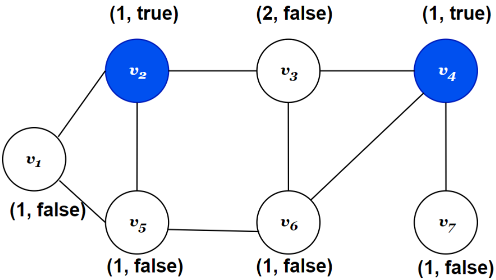


Figure: Algorithme greedy reverse.

Comparaisons

Tous types de graphe

N	4	5	6	7	8	9	10
Nombre de graphes	11	34	156	1044	12346	24668	12005168

Tous types de graphe

N	4	5	6	7	8	9	10
Nombre de graphes	11	34	156	1044	12346	24668	12005168

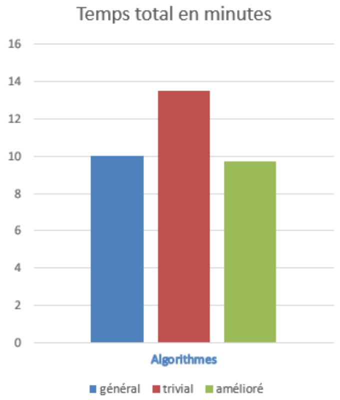


Figure: Comparaison des algorithmes exacts en fonction du temps.

Tous types de graphe

N	$\gamma(G)$	Algorithme général	Algorithme amélioré
24	8	862 *ms*	265 *ms*
26	8	2378 *ms*	1096 *ms*
30	8	10468 *ms*	7639 *ms*
32	8	9381 *ms*	5762 *ms*
34	9	3276833 *ms*	74135 *ms*
34	9	2148155 *ms*	67658 *ms*
34	10	- *ms*	77175 *ms*
34	10	- *ms*	84343 *ms*

Comparaison des heuristiques

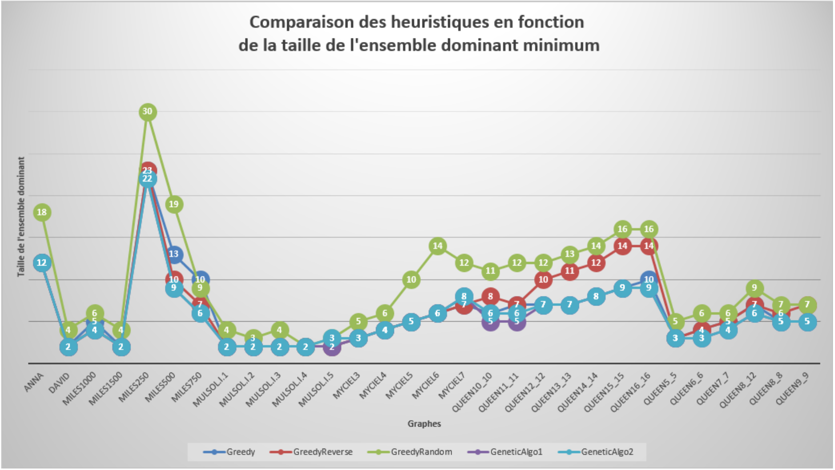


Figure: Comparaison des heuristiques en fonction de la taille du MDS.

Comparaison des heuristiques

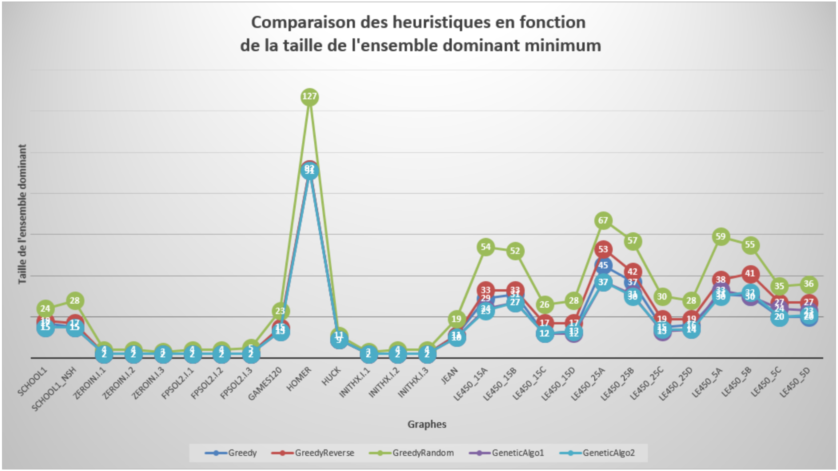


Figure: Comparaison des heuristiques en fonction de la taille du MDS.

Comparaison des heuristiques sur les graphes du réseau social *Google+*

Comparaison des heuristiques sur les graphes du réseau social *Google+*

Algorithme	gplus_200	gplus_500	gplus_2000	gplus_10000	gplus_20000	gplus_50000
Greedy	19(0%)	42(0%)	177(4.1%)	896(4%)	1809(5.4%)	4844(6%)
Greedy reverse	19 (0%)	42 (0%)	171 (0.5%)	876(1.7%)	1760(2.5%)	4751(4%)
Greedy random	37 (95%)	125 (197%)	419(146%)	2065(139%)	4064(136%)	10407(127%)
GeneticAlgo1	19 (0%)	42(0%)	198 (16%)	2350(172%)	5633 (228%)	14460 (216%)
GeneticAlgo2	19 (0%)	42(0%)	179(5.2%)	1723(100%)	5308(209%)	-(-%)
Solution optimale	19	42	170	861	$\geq$ 1716	$\geq$ 4566

Comparaison des heuristiques sur les graphes du réseau social *Google+*

Algorithme	gplus_200	gplus_500	gplus_2000	gplus_10000	gplus_20000	gplus_50000
Greedy	19(0%)	42(0%)	177(4.1%)	896(4%)	1809(5.4%)	4844(6%)
Greedy reverse	19 (0%)	42 (0%)	171 (0.5%)	876(1.7%)	1760(2.5%)	4751(4%)
Greedy random	37 (95%)	125 (197%)	419(146%)	2065(139%)	4064(136%)	10407(127%)
GeneticAlgo1	19 (0%)	42(0%)	198 (16%)	2350(172%)	5633 (228%)	14460 (216%)
GeneticAlgo2	19 (0%)	42(0%)	179(5.2%)	1723(100%)	5308(209%)	-(-%)
Solution optimale	19	42	170	861	$\geq$ 1716	$\geq$ 4566

Conclusion

- Algorithme général très performant pour les graphes qui ont un nombre de domination très petit.

Conclusion

- Algorithme général très performant pour les graphes qui ont un nombre de domination très petit.
- Algorithme amélioré très puissant lorsque le nombre de domination est plus grand.

Conclusion

- Algorithme général très performant pour les graphes qui ont un nombre de domination très petit.
- Algorithme amélioré très puissant lorsque le nombre de domination est plus grand.
- Algorithme génétique très performant pour les graphes de grande taille (< 2000).

Conclusion

- Algorithme général très performant pour les graphes qui ont un nombre de domination très petit.
- Algorithme amélioré très puissant lorsque le nombre de domination est plus grand.
- Algorithme génétique très performant pour les graphes de grande taille (< 2000).
- L'algorithme greedy, et l'algorithme greedy reverse sont les plus performants pour les graphes de taille très large (≥ 2000).

Conclusion

- Algorithme général très performant pour les graphes qui ont un nombre de domination très petit.
- Algorithme amélioré très puissant lorsque le nombre de domination est plus grand.
- Algorithme génétique très performant pour les graphes de grande taille (< 2000).
- L'algorithme greedy, et l'algorithme greedy reverse sont les plus performants pour les graphes de taille très large (≥ 2000).

Questions?

Fedor V Fomin, Dieter Kratsch, and Gerhard J Woeginger.
Exact (exponential) algorithms for the dominating set problem.
In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 245–256. Springer, 2004.

Johan MM Van Rooij and Hans L Bodlaender.
Exact algorithms for dominating set.
Discrete Applied Mathematics, 159(17):2147–2164, 2011.

Wikipedia.
Matching.
https://en.wikipedia.org/wiki/Matching_(graph_theory), 2010 (accessed December 8, 2018).

Wikipedia.
Edge cover problem.
https://en.wikipedia.org/wiki/Edge_cover, 2010 (accessed December 8, 2018).

Laura A Sanchis.
Experimental analysis of heuristic algorithms for the dominating set problem.
Algorithmica, 33(1):3–18, 2002.

Anupama Potluri and Alok Singh.
Two hybrid meta-heuristic approaches for minimum dominating set problem.
pages 97–104, 12 2011.