

# Ensemble Dominant.

BM Bui-Xuan

**Graphe géométrique :** Un graphe géométrique dans un plan 2D est défini par un ensemble de points dans le plan appelés sommets, et un seuil sur la distance entre les points : il existe une arête entre deux sommets si et seulement si la distance Euclidienne entre les deux sommets est inférieure à ce seuil.

**Problème Ensemble Dominant dans un graphe :** Etant donné un graphe  $G = (V, E)$ , le problème Ensemble Dominant consiste à calculer un plus petit sous-ensemble de sommets  $D \subseteq V$  tel que tout sommet  $v \in V$  du graphe est soit un élément de  $D$ , soit un voisin d'un élément de  $D$ .

## 1 Glouton

Proposer une méthode d'heuristique glouton au problème Ensemble Dominant. Implémenter cette méthode dans le fichier `canevas`.

## 2 Local searching naïf

Proposer et implémenter une heuristique naïve dans le fichier `canevas`. Il est recommandé de s'inspirer des méthodes vues pendant la dernière séance, par exemple, avec la RÈGLE naïve. On peut remarquer, comme la séance précédente, que le temps de calcul peut poser problème. Nous proposons d'améliorer ce temps de calcul par une méthode basée sur le caractère géométrique des données, dans la section suivante.

## 3 Optimisation propre aux graphes géométriques

Nous remarquons que la RÈGLE de local searching naïve décrite dans le TME précédente est de grande complexité : à  $k$  fixé la complexité au pire cas est en  $O(n^{2k-1})$ , avec  $n$  le nombre de sommets du graphe. Il existe une optimisation du temps d'exécution populaire dans le monde "algo-prog", mais bien controversée de manière plus générale<sup>1</sup> : l'utilisation des instructions *break* et *continue*. Par exemple, considérons que nous projetons de remplacer trois sommets  $A, B, C$  dans une solution au problème Ensemble Dominant par deux sommets extérieurs. Nous devons dans un cas général examiner toute paire de sommets extérieurs en recherche de celle où le remplacement nous conduit toujours à une solution valide au problème Ensemble Dominant. Cependant, si les trois sommets  $A, B, C$  sont trop éloignés dans le modèle géométrique du graphe (cf. la distance de seuil pour qu'une arête existe), alors il est inutile de chercher la paire de sommets extérieurs pour le remplacement : elle n'existe pas. Une utilisation judicieuse des instructions *break* peut accélérer grandement les recherches du *local searching*.

Proposer et implémenter une accélération du temps de calcul pour la Règle naïve pour  $k = 3$ .

## 4 Autre optimisation géométrique (facultative pour le projet)

On remarque qu'un graphe géométrique peut être découpé en morceaux convexes : le plan Euclidien étant connexe par arcs. S'inspirer des éléments dans [Nieberg, Hurink and Kern. *ACM Transactions on Algorithm*, 2008] et proposer une optimisation en temps de calcul de la précédente heuristique. Implémenter et tester le temps de calcul.

1. Cette pratique est contraire aux *best-practices IT* et contraire à la plupart des *design patterns* les plus utilisés dans l'industrie actuelle, sauf (à la limite) si ces instructions sont placées en tout début de boucle.