

# PACE Challenge 2021 : heuristique pour le cluster editing problem \*

Ulysse Prieto<sup>5</sup>, Valentin Bartier<sup>1</sup>, Gabriel Bathie<sup>2</sup>,  
Nicolas Bousquet<sup>3</sup>, Marc Heinrich<sup>4</sup>, Théo Pierron<sup>3</sup>

<sup>1</sup> G-SCOP, Université Grenoble Alpes, France

<sup>2</sup> École Normale Supérieure de Lyon, France

<sup>3</sup> LIRIS, CNRS, Université Claude Bernard Lyon 1, Université de Lyon, France

<sup>4</sup> University of Leeds, UK

<sup>5</sup> LocalSolver, 24 Avenue Hoche, Paris, France

uprieto@localsolver.com

**Mots-clés** : *recherche locale, cluster editing, large neighborhood search*

## 1 Introduction

Le challenge PACE<sup>1</sup> (Parameterized Algorithms and Computational Experiments Challenge) est un challenge annuel organisé depuis 2015 ayant pour but d'aider au développement pratique d'algorithmes de résolution de problèmes complexes. Le challenge PACE 2021 était consacré à la résolution du CLUSTER EDITING PROBLEM. Nous allons présenter ici  $\mu$ Solver, un solveur développé par Gabriel Bathie, Valentin Bartier, Nicolas Bousquet, Marc Heinrich, Théo Pierron et Ulysse Prieto qui s'est classé 3<sup>ème</sup> dans la partie heuristique du challenge 2021.

## 2 Cluster Editing Problem

Le CLUSTER EDITING PROBLEM est un problème NP-complet dont le but est de trouver le nombre minimum d'édits (ajout ou suppression d'arêtes) nécessaires à transformer un graphe donné en un graphe de cluster, i.e. un graphe dont chaque composante connexe est une clique.

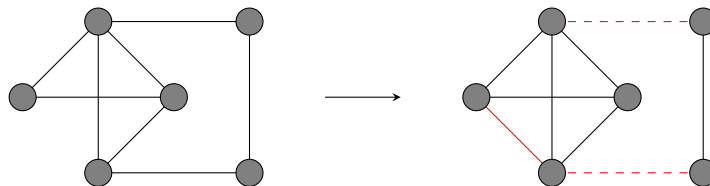


FIG. 1 – Exemple de transformation d'un graphe en un graphe de cluster, il y a ici un ajout et deux suppression d'arêtes.

## 3 Méthode de résolution

La méthode de résolution que nous avons mis au point se base sur deux étapes principales :

---

\*Ce travail a été supporté par le projet ANR GrR (ANR-18-CE40-0032).

1. Voir <https://pacechallenge.org/about/> pour plus de détails.

- Une étape initiale de prétraitement qui simplifie le problème en appliquant de manière répétée un certain nombre de règles. Ces règles dites de "kernelization" permettent de déduire un ensemble d'édicions qu'on sait appartenir à une solution optimale.
- Une solution initiale évidente est par la suite graduellement améliorée par large neighborhood search (LNS). Des ensembles de sommets proches dans le graphe initial sont retirés de la solution avant d'y être réintroduit de manière gloutonne.

Il est également important de noter que dans la 2<sup>ème</sup> étape de résolution, nous considérons les différentes composantes connexes du graphe simplifié comme des sous problèmes indépendants. En effet ajouter une arête entre deux composantes connexes distinctes ne fait jamais partie de la solution optimale.

Le code source et une description plus détaillée du solveur que nous avons soumis pour ce challenge est disponible dans [1].

## 4 Résultats

La méthode présentée ci dessus a été testée sur les 200 instances du challenge PACE<sup>2</sup>. Ces 200 instances sont soit tirées de jeux de données du monde réel, soit générées de manière aléatoire. Elles sont de taille assez variable, les plus grandes ayant plus de 2.5 millions de nœuds et 2.5 millions d'arêtes (voir [2] pour plus de détails sur la répartition et le choix des instances).

Pour chacun des participants, 10 minutes de calcul étaient alloué par instance. Pour chaque instance, la taille d'édicion minimale  $s_{\min}$  parmi l'ensemble des participants est comparée à la taille d'édicion  $s$  d'une soumission. Le score associé à la résolution d'une instance est alors de  $100 \cdot s_{\min}/s$ . Le score total est calculé comme la moyenne des scores sur l'ensemble des instances.

La table 1 présente le score total des 5 premières équipes du classement. La très grande efficacité d'une recherche locale simple sur le CLUSTER EDITING PROBLEM a malheureusement probablement poussé le classement à être déterminé plus par la qualité d'implémentation que par la qualité des heuristiques développées. Les 4 premiers solveurs utilisent en effet une recherche locale ayant pour mouvement de base le déplacement d'un unique sommet d'un cluster vers un autre. L'utilisation de règles de kernelization ne semble pas avoir joué un rôle important dans la compétition, le vainqueur n'en utilisant aucune. Une présentation détaillée des résultats et des méthodes des participants peut être trouvée dans [2].

Classement	1	2	3	4	5
Score	99.9992	99.9982	99.9977	99.9884	99.9793

TAB. 1 – Score total des 5 première équipes du classement heuristique,  $\mu$ Solver étant classé 3<sup>ème</sup>.

## Références

- [1] Valentin Bartier, Gabriel Bathie, Nicolas Bousquet, Marc Heinrich, Théo Pierron, and Ulysse Prieto. Pace 2021 musolver, 2021. <https://zenodo.org/record/4947325>.
- [2] Leon Kellerhals, Tomohiro Koana, André Nichterlein, and Philipp Zschoche. The pace 2021 parameterized algorithms and computational experiments challenge : Cluster editing, 2021. <https://pacechallenge.org/2021/pace21-report.pdf>.

---

2. <https://github.com/PACE-challenge/Cluster-Editing-PACE-2021-instances>