

# Votre Plan d'Exécution de Requêtes est un Circuit Intégré : Changer de Métier

Ahcène Boukorca\*, Ladjel Bellatreche\*, Sid-Ahmed Benali Senouci \*\*, Zoé Faget \*

\* LIAS/ISAE-ENSMA, Téléport 2 - 1 avenue Clément Ader - BP 40109,  
86961 Futuroscope Chasseneuil Cedex - France  
(boukorca,bellatreche, zoe.faget)@ensma.fr,  
\*\* Mentors Graphics  
38330 Montbonnot-Saint-Martin - France  
sid-ahmed\_senouci@mentor.com

**Résumé.** Dans la première génération des bases de données, les optimiseurs étaient conçus pour optimiser des requêtes individuelles. Après identification des interactions entre les requêtes, des optimiseurs étaient proposés pour offrir une optimisation multiple. La difficulté de cette optimisation est l'identification des expressions communes entre les requêtes. Ce problème est connu comme NP-difficile Sellis et Ghosh (1990). Pour résoudre ce problème, des solutions basées sur la fusion des arbres algébriques des requêtes ont été proposées. Elles souffrent du problème de passage à l'échelle. Dans cet article, nous proposons l'utilisation de la théorie de graphes fortement utilisée dans le domaine de la conception des circuits intégrés pour la résolution de ce problème. Premièrement, nous définissons l'analogie entre notre problème et celui de la conception des circuits électroniques. Des algorithmes issus de la théorie des graphes sont ensuite proposés. Finalement, une évaluation de nos propositions est effectuée à l'aide de l'outil HMETIS.

## 1 Introduction

L'accès efficace à la donnée a toujours été un enjeu important dans le monde des bases de données. Cette dimension s'est accentuée avec l'arrivée du big data. Cet accès est assuré par l'intermédiaire de requêtes complexes souvent interdépendantes (Ahmad et al. (2011); Sellis (1988); Yang et al. (1997a)). Les utilisateurs des bases de données exigent la performance des accès aux données. Pour satisfaire cette contrainte, les optimiseurs de requêtes se sont développés. L'optimiseur a deux tâches principales à accomplir en un temps raisonnable : d'abord, il lui faut trouver l'ensemble des plans d'exécution pour une même requête. Ensuite, il lui faut choisir parmi ces alternatives la plus performante pour exécuter la requête en se basant sur certains critères. L'existence de ces multiples alternatives est due aux caractéristiques des opérations algébriques (commutativité, associativité du produit cartésien, inversion sélection-projection, inversion sélection-produit cartésien, etc.). Deux types d'optimisateurs existent : (i) les optimiseurs de requêtes individuelles et (ii) les optimiseurs de requêtes multiples. Dans