**Projet : Recherche opérationnelle**

**Analyse de la complexité expérimentale des algorithmes de flot**

**Introduction**

Dans ce projet de recherche opérationnelle, nous avons étudié et comparé trois algorithmes de résolution de problèmes de flots : Ford-Fulkerson (FF), Pousser-Réétiqueter (PR), et Flot à coût minimal (MIN). L’objectif est de comprendre empiriquement leurs complexités respectives à l’aide de graphes aléatoires, en comparant les résultats obtenus aux complexités théoriques présentées dans le cours.

**Méthodologie expérimentale**

Pour chaque taille de graphe n ∈ {10, 20, 40, 100, 400}, nous générons 100 graphes aléatoires. Chaque graphe contient environ n²/2 arêtes, avec des capacités et des coûts entiers tirés uniformément dans [1, 100]. Nous appliquons successivement les algorithmes FF, PR et MIN à chaque graphe, et mesurons le temps d’exécution de chacun.

Scripts utilisés :

- `generationGraphe.py` : génération des graphes aléatoires

- `testComplexite.py` : mesures de temps sur chaque algorithme

- `plots.py` : génération des graphiques à partir des résultats

**Résultats obtenus**

Nuage de points pour Ford-Fulkerson

Une image contenant texte, ligne, Tracé, diagramme

Le contenu généré par l’IA peut être incorrect.

Nuage de points pour Pousser-Réétiqueter

Une image contenant texte, ligne, diagramme, Tracé

Le contenu généré par l’IA peut être incorrect.

Nuage de points pour Flot à coût minimal

Une image contenant texte, ligne, Tracé, diagramme

Le contenu généré par l’IA peut être incorrect.

Comparaison des complexités FF vs PR

Une image contenant ligne, Tracé, diagramme, texte

Le contenu généré par l’IA peut être incorrect.

Courbes d'enveloppe maximale (θFF, θPR, θMIN)

Une image contenant ligne, texte, Tracé, diagramme

Le contenu généré par l’IA peut être incorrect.

**Analyse des courbes et comparaisons**

Les courbes et nuages de points permettent d'observer les tendances de complexité des algorithmes :

- θFF(n) : la croissance semble quadratique (O(n²)), comme prédit par Edmonds-Karp

- θPR(n) : complexité plus élevée, proche de O(n³), comme indiqué dans le cours

- θMIN(n) : très coûteux, cohérent avec l’usage de Bellman-Ford

**Conclusion**

Ce travail expérimental nous a permis de confronter les prédictions théoriques aux résultats mesurés. Les observations confirment les complexités prévues dans le cours. Ford-Fulkerson est le plus léger, tandis que PR et MIN deviennent très coûteux pour les grands graphes. Cette analyse pourrait être approfondie avec des tailles supérieures (n > 1000) et d'autres variantes d'algorithmes.