Activités, problèmes, résultats

Clément Legrand-Lixon

22 Mai 2018 Arrivée dans le laboratoire (Cristal). Rencontre de l'équipe (doctorants et stagiaires). Installation du poste de travail (Ubuntu, espace de travail : VSC, latex, R). Découverte du problème, recherche d'informations (variantes : multi-dépôts, avec capacité, avec limite de temps...). Lecture des articles *A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem* (K. Altınel, T. Oncan), et *A simple, deterministic, and efficient knowledge-driven heuristic for the vehicle routing problem* (Florian Arnold, Kenneth Sorensen). Vu administration pour la gratification de stage.

23 Mai 2018 Fin de lecture des articles, résumé des articles tapés en tex. Implémentation de l'algorithme de Clarke and Wright (C W), en python. Mise en place d'une heuristique basique (avec paramètre λ). Tests de l'algo sur des instances fabriquées aléatoirement. Tentative pour trouver une valeur optimale pour le paramètre selon la taille de l'entrée. Résultats disponibles avec la figure 1.

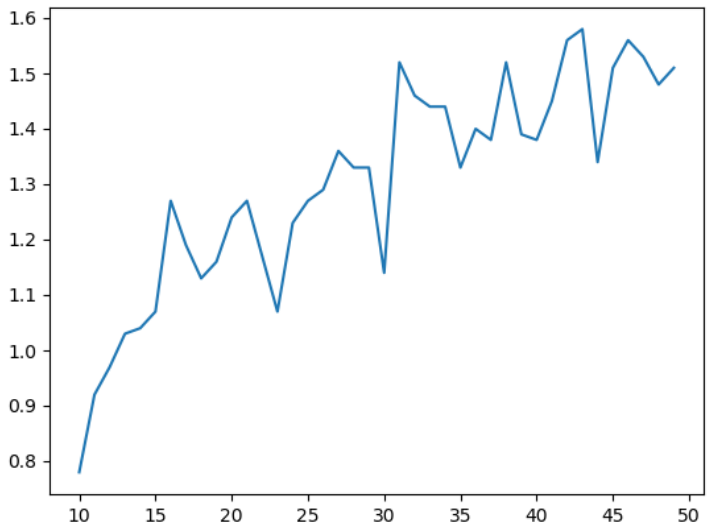

FIGURE 1 – Valeur de lambda optimale

24 Mai 2018 Rencontre avec encadrante senior : Marie-Éléonore, bilan sur les premiers jours. Obtention du code C++ de l'heuristique de CW (complète, mais quelques erreurs présentes à corriger). Objectif : Pour la semaine prochaine, faire une présentation reprenant la nouvelle

heuristique développée dans l'article de F. Arnold et K. Sorensen. Et éventuellement implémentation des opérateurs locaux utilisés dans l'heuristique en expliquant les intérêts et limites de chaque (complexité, avantage...). Début de la présentation. Recherche d'articles détaillant les opérateurs utilisés. Peu de résultats (page wikipédia pour Lin Kernighan).

25 Mai 2018 Rencontre avec Lætitia Jourdan (responsable de l'équipe). Implémentation de l'heuristique de Lin-Kernighan : pour une tournée donnée, optimise la visite des clients (utilisée pour TSP). Commence par réaliser 2-opt (cherche un changement entre deux arêtes qui améliore la tournée). Si trouvé on passe 3-opt (echange de 3 aretes), jusqu'à k-opt (k choisi). Puis applique la meilleure modif (la meilleure i-opt). S'arête lorsque plus d'améliorations possibles).

Implémentation de 2-opt en python, tests sur quelques instances. On part de 2, et en appliquant 2-opt on obtient 3. 'Dépôt représenté en bleu).

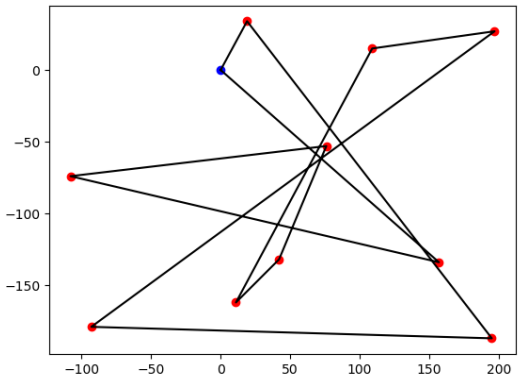

FIGURE 2 – Instance initiale

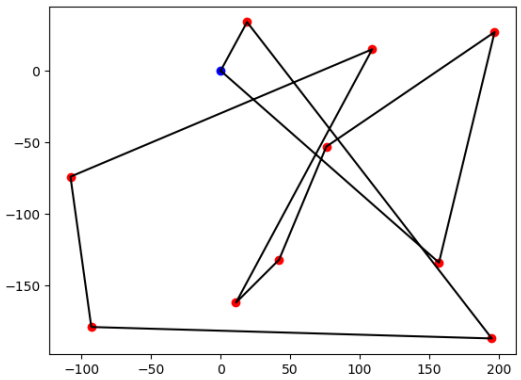

FIGURE 3 – Optimisation avec 2-opt

Rencontre de problèmes avec des instances supérieures, et LK. → A résoudre lundi. Opérateur à utiliser principalement sur des petites portions de route, complexité élevée $O(n^k)$ avec n le nombre de clients.

28 Mai 2018 Objectifs : finir d'implémenter LK, tester, et finir le développement dans la présentation. Améliorer présentation. Implémenter un autre opérateur *Cross-exchange* ou *Ejection-chain*.

Réalisation :