

24/10/2017

Rapport TP1 LOG2810

(STRUCTURES DISCRETES)

**Auteurs :**

Etienne Laval

Constantin Bouis 1783438

Xavier Ralay 1786307

**Tables des matières :**

[1.Introduction 2](#_Toc496607632)

[2. Drones 3](#_Toc496607633)

# **1.Introduction**

Dans le but de venir en aide à un étudiant du département du génie informatique et génie logiciel nous avons mis au point un projet qui répond aux critères de ses attentes. Ce dernier a notamment eu besoin de notre aide dans le cadre de ses stages.

Pour commencer, venant juste de se trouver un stage auprès d’une start-up de drones qui aimerait populariser les livraisons de colis sur l’île de Montréal par des drones donc par voies aériennes, la start-up lui met déjà en charge d’un grand projet. En effet, il sera en charge d’un algorithme ayant pour but d’optimiser les chemins qu’emprunteront les drones lors de la livraison d’un colis pour ainsi une minimiser les coûts d’exploitation. Cet algorithme rappel plusieurs notions théorique de graphes du cours de LOG2810: Structures Discrètes. Cependant, ayant avoir stipuler dans son C.V d’avoir réussi le cours, le nouveau stagiaire n’a malheureusement pas très bien assimilé les notions du cours. Il a donc désespérément besoin de notre aide. Pour cela nous l’avons aidé à implémenter quelques composantes de son algorithme.

Ensuite, l’étudiant ne s’en arrête pas là. En effet, dans le cadre de son prochain stage, cependant cette fois ci dans une autre compagnie. Pour ce stage, le jeune étudiant doit remettre un autre programme ou il s’agit encore de notions du cours de Structures Discrètes. En effet, son programme doit pouvoir déterminer un ordre er tracer un diagramme de Hasse d’après une liste de sommets et d’arêtes orientées ente ces sommets dont il aura tracé le graphe. Il a donc encore besoin de notre aide.

Par suite, nous allons vous expliquer les moyens qu’on a utilisé pour mener a bien ce TP. Puis lors de l’élaboration du TP, nous avons rencontré notamment quelques difficultés qu’on vous expliquera ensuite.

# **2. Drones**

Pour la mise en place de l’architecture des graphs, nous avons repris le travail qu’Etienne Laval avait déjà fait pour un autre projet d’algorithmique dans son école en France. (ConsultablesurGithubhttps://github.com/AlgoAvanc/ProjetAlgo).

Il s’est agi de largement en élaguer les morceaux inutiles puis de construire les fonctionnalités requises par ce TP dessus.

Cette architecture va comme suit :

Trois classes : Graph, Edge et Node.

Les nodes (nœuds) contiennent leur identifiant (un String) ainsi que mention des edges (arcs) qui partent d’eux sous la forme d’une liste.

Les edges sont orientés, ils contiennent une référence à leur nœud de départ et à leur nœud d’arrivée ainsi qu’un poids.

Les graphs contiennent trois listes : celle de leur edges, celle de leur nodes et celle des id de leur nodes (pour les retrouver plus rapidement).

A cela nous avons adapté un module de lecture de fichiers .txt et ajouté le fait d’avoir des nœuds de recharges de drones (par booléen sur les nœuds)

Ayant la fonction de création de graph, la mise à jour du graph n’est qu’une recréation de ce graph.

Pour ce qui est de l’implémentation du calcul de chemin par un drone c’était assez simple, il y a deux variantes par rapport au Dijkstra proposé :

-On tient aussi un compte de la charge de batterie une fois arrivée à un nœud (en plus de renseigner le nœud précédent et la distance/temps au premier nœud)

-On ne met à jour la distance/temps à un nœud que si celui-ci est assez proche pour que le drone ne tombe pas en panne (on ajoute bien sûr le temps de recharge dans le décompte du temps le cas échéant, de même que la recharge elle-même).

Pour l’utilisation de la batterie, on entre en paramètre le taux de consommation (en part de batterie par minute de la technique testée). Ainsi, si une fois l’algorithme testé, le nœud visé a un parent dans l’algorithme de Dijkstra, alors, il y a bien une possibilité de trajet. Si ce n’est pas le cas, on le réessaie avec en taux de consommation d’énergie, celui des drones plus puissants et si cela ne marche toujours pas, on sait que le voyage est impossible.

En termes d’architecture cet algorithme et les tableaux de temps/précédent/consommation sont dans une classe (nommée DijkstraPlus). Ainsi, on peut faire correspondre les données déjà calculées dans un objet rattaché à chaque nœud.