KUSNO Louis, SUN Jixiang
G97

Our Garden

La recette

Nous avons choisi une recette aux fruits rouges avec "fraisier des bois", "framboisier", "cerisier" et "cassis" comme ingrédients.

Exécution du program

Le ficher **./code/main.py** rassemble toutes les fonctions et crée tous les diagrammes.

Première version

Pour commencer nous avons utilisé l'algorithme de BFS pour trouver le cycle avec le moins de nœuds. Cependant, le fait d'avoir plusieurs ingrédients signifie que le cycle entre, "fraisier des bois", "framboisier", "cerisier" et "cassis" dans cette ordre, ne peut être pas celui le plus courts. Pour surmonter ce problème, nous avons calculé le chemin le plus courts pour toutes les permutations possibles des ingrédients. Par conséquent, cela posse une grande limitation de la scalabilité de notre algorithme car le nombre de permutation augmente de n !.

Le code de la première version se trouve dans **./code/garden_gen.py** et le graphe du jardin dans **./garden_v1.png**.

Deuxième version

La première version a une autre limitation, dans le monde « réel », il faut une quantité de composte pour pouvoir ajouter une certaine espèce. C'est pour ça que nous avons utilisé l'algorithme Dijkstra avec les données pondérées (**./json/favorisePoids.json**) pour trouver le chemin le moins cher. Ce code prend aussi en compte toutes les permutations possibles des ingrédients mais aussi ces limitations, expliciter dans la première version.

Le code de la deuxième version se trouve dans **./code/garden_gen_weight.py** et le graphe du jardin dans **./garden_v2.png**.

Troisième version

Nous avons beau se rapprocher de la réalité avec la deuxième version mais, ce n'est toujours pas représentatif du monde réel. Dans cette version nous prenons aussi en compte la différence entre les conditions optimal de chaque plante et ceux du jardin. Cela consiste à prendre l'écart entre les deux et de l'ajouter au coût de composte (l'ordre de grandeur des écarts et du coût de composte sont à peu près équivalents donc nous les avons juste ajoutés ensemble).

KUSNO Louis, SUN Jixiang
G97

Le ficher **./json/favorisePoids.json** enregistre la quantité de compost nécessaire pour mettre la planteB près de la planteA **{"planteA" : {"planteB" : cout_compost}}**. Pour prendre en compte les conditions de l'utilisateur (soit le jardin), nous avons décidé d'ajouter l'écart entre toutes les conditions de la planteA et de l'utilisateur, puis de les multiplier par leurs poids respectifs. Ensuite, d'ajouter ce nombre a tous les couts de planteA a planteX **{"planteA" : {"planteX" : cout_compost + ecarts * poids}}**. En calculant un cycle, le program prend bien tous les coûts totaux donc aussi les écarts.

Le code de cette 3eme version se trouve dans **./code/garden_gen_bio.py**, il crée un nouveau ficher **./json/favoriseBioindicator.json** mais le code qui s'occupe de trouver le cycle le moins cher est toujours celui de la deuxième version, **./code/garden_gen_weight.py**, il suffit juste de modifier le fichier json utilisé. Le graph de ce jardin ce trouve dans **./garden_v3.png**.

Comparaison

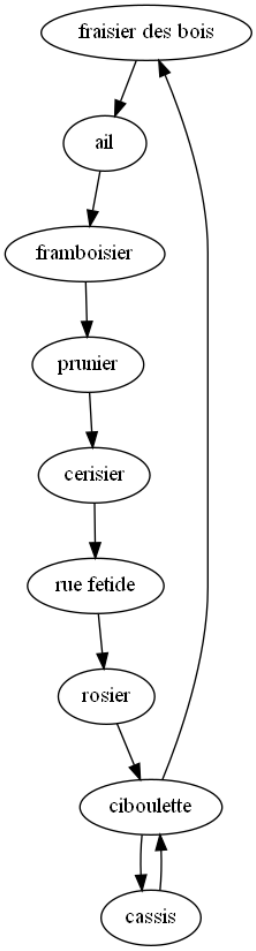

Figure 1 Première version

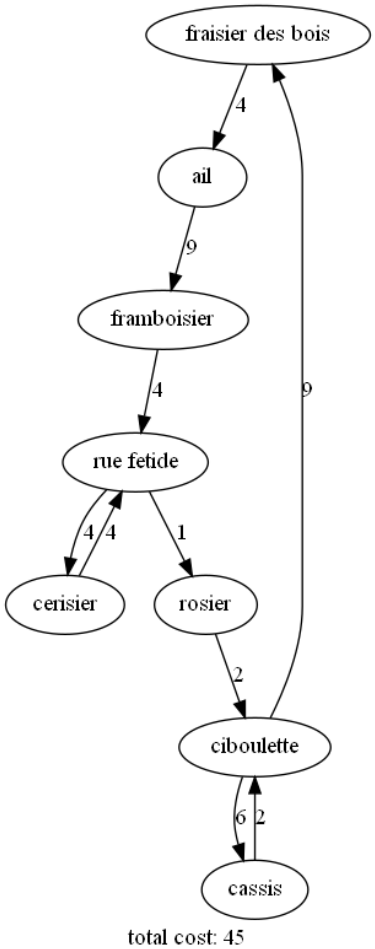

Figure 2 Deuxième version

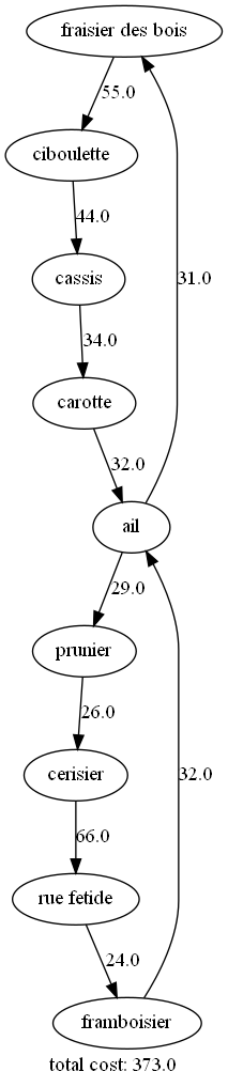

Figure 3 Troisième version

Nous pouvons bien voir les différences entre les graphes. Ceci nous montre que prendre en compte plus de conditions aboutis à différents résultats qui nous rapproche de la réalité (si les données sont vraies) mais rend l'algorithme de plus en plus complexe.