Le voyageur de commerce, méthode de *branch and bound*.

Pierre Chatelain, Kevin Quirin

ENS Cachan - Antenne de Bretagne

15 août 2010

TSP

Le problème du voyageur de commerce (TSP) consiste à trouver le chemin le plus court qui passe par n villes, étant données les distances.
Formellement, TSP s'écrit :

TSP

Soit $G = (V, E, \omega)$ un graphe complet ($E = V \times V$), où ω est une fonction de poids sur E. Trouver le cycle hamiltonien de G de poids minimum.

La solution naïve est en complexité $O(|V|!)$, ce qui la rend inexploitable pour plus de 15 villes.
TSP est NP-complet, on utilise donc deux types d'algorithmes :

- Les algorithmes exacts
- Les algorithmes d'approximations

On va étudier ici un algorithme exact, la méthode du branch and bound (séparation et évaluation).

Données du problème

- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ est un graphe complet à n sommets.

Données du problème

- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ est un graphe complet à n sommets.
- Ce graphe est représenté par sa matrice d'adjacence $C = (c_{i,j})$, qui sera modifiée au cours de l'algorithme.

Données du problème

- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ est un graphe complet à n sommets.
- Ce graphe est représenté par sa matrice d'adjacence $C = (c_{i,j})$, qui sera modifiée au cours de l'algorithme.
- Un chemin t dans le graphe est représenté par une suite d'arêtes, et son coût est défini par :

$$z(t) = \sum_{(i,j) \in t} c_{i,j}$$

Données du problème

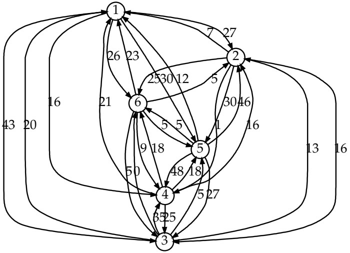
- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ est un graphe complet à n sommets.
- Ce graphe est représenté par sa matrice d'adjacence $C = (c_{i,j})$, qui sera modifiée au cours de l'algorithme.
- Un chemin t dans le graphe est représenté par une suite d'arêtes, et son coût est défini par :

$$z(t) = \sum_{(i,j) \in t} c_{i,j}$$

But

Trouver t hamiltonien minimisant z.

Exemple



∞	27	43	16	30	26
7	∞	16	1	30	25
20	13	∞	35	5	0
21	16	25	∞	18	18
12	46	27	48	∞	5
23	5	5	9	5	∞

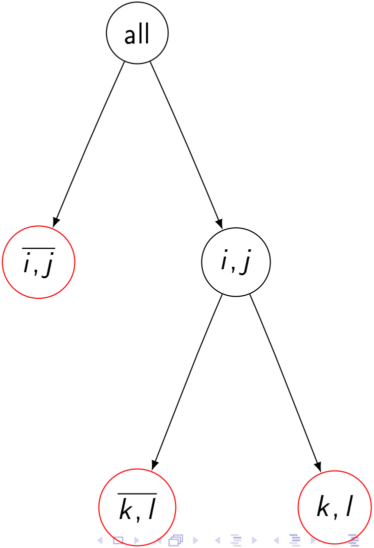
Exemple

∞	27	**43**	16	30	26
7	∞	16	1	**30**	25
20	**13**	∞	35	5	0
21	16	25	∞	18	18
12	46	27	48	∞	**5**
23	5	5	**9**	5	∞

$$z = 43 + 13 + 30 + 5 + 9 + 21 = 121$$

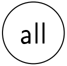
Représentation par un arbre binaire

- L'ensemble des cycles hamiltoniens possibles du graphe est représenté par les feuilles d'un arbre.



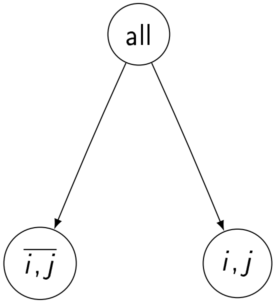
Représentation par un arbre binaire

- L'ensemble des cycles hamiltoniens possibles du graphe est représenté par les feuilles d'un arbre.
- Initialement, l'arbre a un seul noeud, représentant tous les cycles.



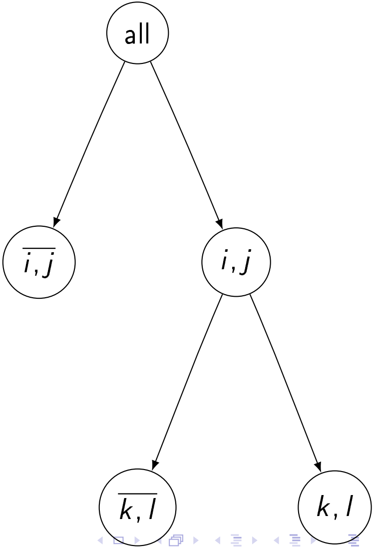
Représentation par un arbre binaire

- L'ensemble des cycles hamiltoniens possibles du graphe est représenté par les feuilles d'un arbre.
- Initialement, l'arbre a un seul noeud, représentant tous les cycles.
- Le noeud i, j représente tous les cycles contenant l'arête (i, j), et le noeud $\overline{i, j}$ tous les cycle qui ne la contiennent pas.



Représentation par un arbre binaire

- L'ensemble des cycles hamiltoniens possibles du graphe est représenté par les feuilles d'un arbre.
- Initialement, l'arbre a un seul noeud, représentant tous les cycles.
- Le noeud i, j représente tous les cycles contenant l'arête (i, j), et le noeud $\overline{i, j}$ tous les cycle qui ne la contiennent pas.
- Chaque étage de l'arbre divise ainsi l'ensemble des cycles.



Réduction

∞	27	43	16	30	26
7	∞	16	1	30	25
20	13	∞	35	5	0
21	16	25	∞	18	18
12	46	27	48	∞	5
23	5	5	9	5	∞

Réduction

∞	27	43	16	30	26
6	∞	15	0	29	24
20	13	∞	35	5	0
21	16	25	∞	18	18
12	46	27	48	∞	5
23	5	5	9	5	∞

Réduction sur une ligne ($h = 1$)

Réduction

∞	22	43	16	30	26
6	∞	15	0	29	24
20	8	∞	35	5	0
21	11	25	∞	18	18
12	41	27	48	∞	5
23	0	5	9	5	∞

Réduction sur une colonne ($h = -5$)

Réduction

∞	11	27	0	14	10
1	∞	15	0	29	24
15	13	∞	35	5	0
0	0	9	∞	2	2
2	41	22	43	∞	0
13	0	0	4	0	∞

Réduction sur les lignes, puis les colonnes : $h = 48$

$$z(t) = h + z1(t) \geq 48$$

Initialisation

Meilleur parcours

- On définit Z_0 par le coût du meilleur parcours trouvé.

Initialisation

Meilleur parcours

- On définit Z_0 par le coût du meilleur parcours trouvé.
- Initialement, $Z_0 = +\infty$.

Initialisation

Meilleur parcours

- On définit Z_0 par le coût du meilleur parcours trouvé.
- Initialement, $Z_0 = +\infty$.

Réduction et initialisation de l'arbre

- L'arbre contient seulement le noeud *all*.

Initialisation

Meilleur parcours

- On définit Z_0 par le coût du meilleur parcours trouvé.
- Initialement, $Z_0 = +\infty$.

Réduction et initialisation de l'arbre

- L'arbre contient seulement le noeud *all*.
- La matrice est réduite, et le coefficient de réduction est retenu pour le noeud :

Choix d'un noeud

Critère de choix

- Idée : Partitionner l'ensemble des parcours en T_1, T_2, où T_1 contient "probablement" le parcours optimal, et T_2 a peu de chances de le contenir.

Choix d'un noeud

Critère de choix

- Idée : Partitionner l'ensemble des parcours en T_1, T_2, où T_1 contient "probablement" le parcours optimal, et T_2 a peu de chances de le contenir.
- On choisit de partitionner selon une arête de coût nul.

Choix d'un noeud

Critère de choix

- Idée : Partitionner l'ensemble des parcours en T_1, T_2, où T_1 contient "probablement" le parcours optimal, et T_2 a peu de chances de le contenir.
- On choisit de partitionner selon une arête de coût nul.
- Parmi les choix restants, on maximise la fonction :

$$\theta(i,j) = \min_{k \neq j}(c_{i,k}) + \min_{k \neq i}(c_{k,j})$$

Exemple

∞	11	27	0	14	10
1	∞	15	0	29	24
15	13	∞	35	5	0
0	0	9	∞	2	2
2	41	22	43	∞	0
13	0	0	4	0	∞

$$\theta(1,4) = 10 + 0 = 10$$

Exemple

∞	11	27	10	14	10
1	∞	15	1	29	24
15	13	∞	35	5	5
1	0	9	∞	2	2
2	41	22	43	∞	2
13	0	9	4	2	∞

$$argmax(\theta(i,j)) = (1,4)$$

Simplification de la matrice

1	∞	15		29	24
15	13	∞		5	0
∞	0	9		2	2
2	41	22		∞	0
13	0	0		0	∞

Réduction

[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
0	∞	14	[redacted]	28	23
15	13	∞	[redacted]	5	0
∞	0	9	[redacted]	2	2
2	41	22	[redacted]	∞	0
13	0	0	[redacted]	0	∞

Mise à jour de Z_0

Fin de l'algorithme

Complexité au pire

- Au pire, on explore tous les chemins possibles $\rightarrow O((n-1)!)$

Complexité au pire

- Au pire, on explore tous les chemins possibles $\rightarrow O((n-1)!)$
- On va étudier la complexité en moyenne.

Complexité en moyenne

On définit un arbre aléatoire $T(b, d)$ par :

Complexité en moyenne

On définit un arbre aléatoire $T(b, d)$ par :

- $T(b, d)$ est de profondeur d.

Complexité en moyenne

On définit un arbre aléatoire $T(b, d)$ par :

- $T(b, d)$ est de profondeur d.
- Les nombres de fils des noeuds sont indépendants, identiquement distribués, d'espérance b.

Complexité en moyenne

On définit un arbre aléatoire $T(b, d)$ par :

- $T(b, d)$ est de profondeur d.
- Les nombres de fils des noeuds sont indépendants, identiquement distribués, d'espérance b.
- Les coûts des arêtes sont indépendants et identiquement distribués.

Complexité en moyenne

On définit un arbre aléatoire $T(b, d)$ par :

- $T(b, d)$ est de profondeur d.
- Les nombres de fils des noeuds sont indépendants, identiquement distribués, d'espérance b.
- Les coûts des arêtes sont indépendants et identiquement distribués.
- Le coût d'un noeud est la somme des coûts des arêtes de la racine à ce noeud.

But

On va donc chercher le nombre de branches que doit explorer l'algorithme pour arriver à la feuille de coût minimal.

But

On va donc chercher le nombre de branches que doit explorer l'algorithme pour arriver à la feuille de coût minimal.

On note p_0 la probabilité que le coût d'une arête soit 0.

Théorème

Dans un arbre aléatoire $T(b, d)$, quand $d \to \infty$:

Théorème

Dans un arbre aléatoire $T(b, d)$, quand $d \to \infty$:

- Si $bp_0 < 1$, la recherche de la feuille de coût minimal est de complexité exponentielle.

Théorème

Dans un arbre aléatoire $T(b, d)$, quand $d \to \infty$:

- Si $bp_0 < 1$, la recherche de la feuille de coût minimal est de complexité exponentielle.
- Si $bp_0 \geq 1$, la recherche de la feuille de coût minimal est de complexité polynomiale.

Théorème

Dans un arbre aléatoire $T(b, d)$, quand $d \to \infty$:

- Si $bp_0 < 1$, la recherche de la feuille de coût minimal est de complexité exponentielle.
- Si $bp_0 \geq 1$, la recherche de la feuille de coût minimal est de complexité polynomiale.

On voit donc qu'il y a une transition de complexité en $bp_0 = 1$, d'exponentielle à polynomiale.

Dans le cas du TSP, l'hypothèse d'indépendance n'est plus réaliste.

Dans le cas du TSP, l'hypothèse d'indépendance n'est plus réaliste.

Cependant, des études expérimentales montrent qu'il y a une transition de complexité en fonction du nombre de valeurs possibles pour les distances entres les villes.