



Monte Carlo Tree Search

CentraleSupélec – Gif

ST2 – Gif



Plan

- 1 Introduction
- 2 Construction de l'Arbre
- 3 Simulation Monte Carlo
- 4 Problème de Bandit
- 5 Amélioration de l'Algorithme
- 6 Conclusion

Problématique rencontrée

Problème concret

- Étant donné une liste de ville et les distances entre ces villes, quel est le plus court chemin qui passe une fois par chaque ville et termine à la ville de départ ?

C'est le problème du **voyageur de commerce** (en anglais *Traveling Salesman Problem* ou **TSP**).



Plan

- 1 Introduction
- 2 Construction de l'Arbre**
- 3 Simulation Monte Carlo
- 4 Problème de Bandit
- 5 Amélioration de l'Algorithme
- 6 Conclusion

Gérer la complexité

Beaucoup de problèmes d'optimisation rencontrés sont *NP*-complet. Il est donc nécessaire de développer des méthodes pour les traiter au mieux.

Méthodes de résolution

- Méthodes exactes (**Branch & Bound**, Programmation Linéaire, ...)
- Heuristiques dédiées (**d'approximation** ou sans garantie de performance)
- Métaheuristiques (**Recuit Simulé**, Tabou, Algorithmes Génétiques, ...)

Gérer la complexité

Beaucoup de problèmes d'optimisation rencontrés sont *NP*-complet. Il est donc nécessaire de développer des méthodes pour les traiter au mieux.

Méthodes de résolution

- Méthodes exactes - Cours 9
- Heuristiques dédiées - Cours 10
- Métaheuristiques - Cours 10



Plan

- 1 Introduction
- 2 Construction de l'Arbre
- 3 Simulation Monte Carlo**
- 4 Problème de Bandit
- 5 Amélioration de l'Algorithme
- 6 Conclusion



Algorithme

Algorithme

```
1: function backtracking(s)
2:   if terminal(s) then
3:     return verify(s)
4:   end if
5:   for all  $c \in \text{children}(s)$  do
6:     if backtracking(c) then
7:       return TRUE
8:     end if
9:   end for
10:  return FALSE
11: end function
```



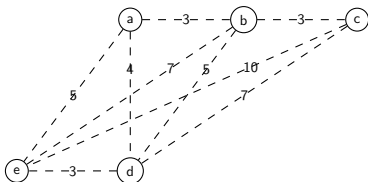

Plan

- 1 Introduction
- 2 Construction de l'Arbre
- 3 Simulation Monte Carlo
- 4 Problème de Bandit**
- 5 Amélioration de l'Algorithme
- 6 Conclusion

Efficacité de la borne : cas du voyageur de commerce

Remarques

- Le parcours doit passer par toutes les villes.
- Pour chaque ville il faut arriver et repartir.
- Au mieux, pour chaque ville, on arrivera et on repartira par les deux arcs les plus petits



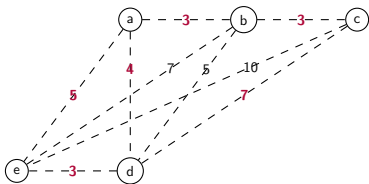
Efficacité de la borne : cas du voyageur de commerce

Remarques

- Le parcours doit passer par toutes les villes.
- Pour chaque ville il faut arriver et repartir.
- Au mieux, pour chaque ville, on arrivera et on repartira par les deux arcs les plus petits

Idée

La borne utilise les distances aux deux voisins les plus proches

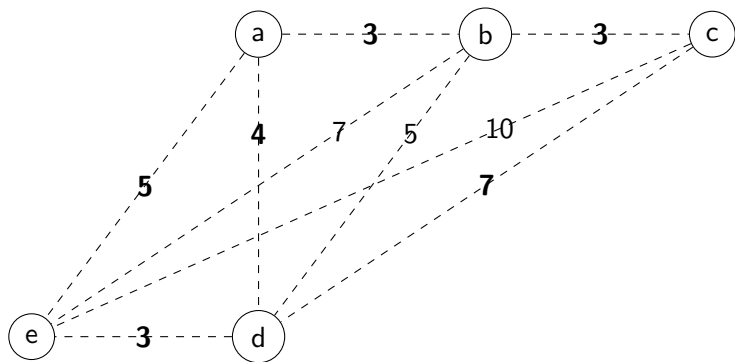




Plan

- 1 Introduction
- 2 Construction de l'Arbre
- 3 Simulation Monte Carlo
- 4 Problème de Bandit
- 5 Amélioration de l'Algorithme**
- 6 Conclusion

Exemple borne



Calcul de la borne

trajet déjà effectué = \emptyset

$$\underbrace{(3+4)}_a + \underbrace{(3+3)}_b + \underbrace{(3+7)}_c + \underbrace{(3+4)}_d + \underbrace{(3+5)}_e \bigg/ 2 = 19$$



Plan

- 1 Introduction
- 2 Construction de l'Arbre
- 3 Simulation Monte Carlo
- 4 Problème de Bandit
- 5 Amélioration de l'Algorithme
- 6 Conclusion**