ISEN | école d'ingénieurs
LILLE

Algorithmes sur les graphes

Problème du voyageur de commerce:
Algorithme de Little

ISEN | école d'ingénieurs
LILLE

Introduction

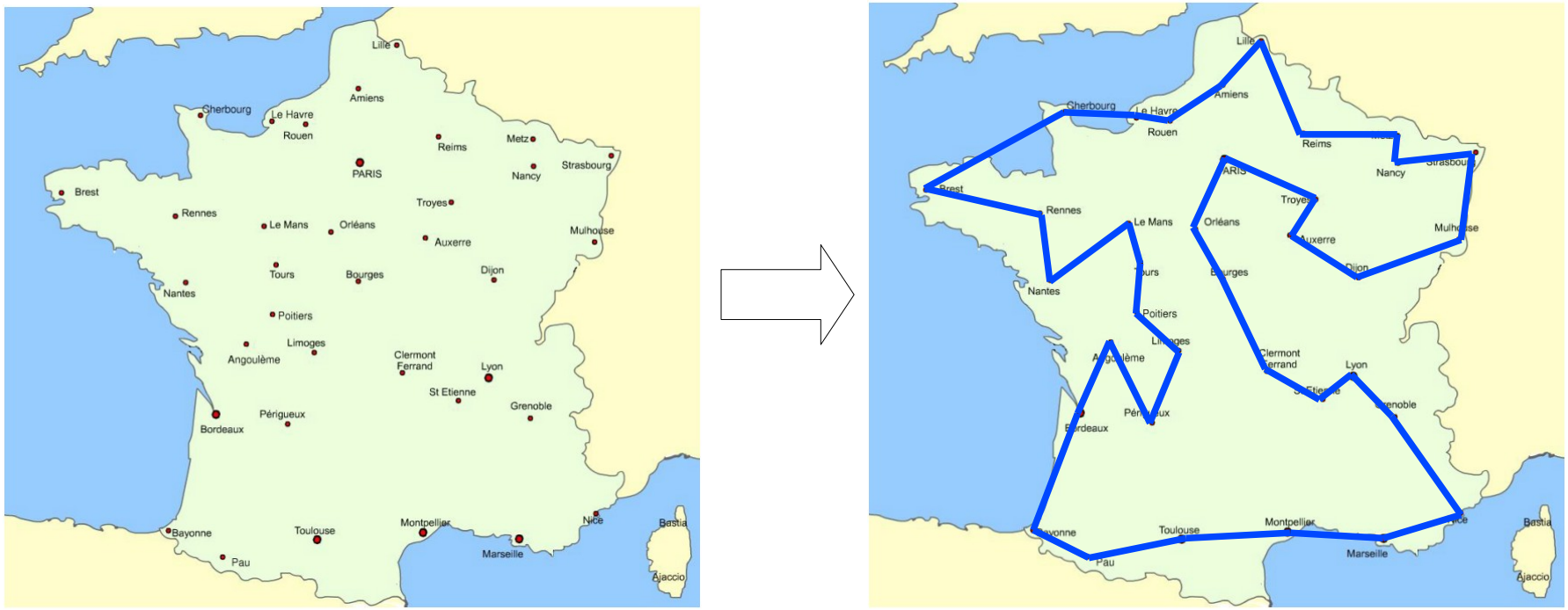
ISEN | école d'ingénieurs
LILLE

Formulation du problème

- Problème du voyageur de commerce (*Traveling Salesman Problem - TSP*) :

Calculer une tournée longueur minimale passant une et une seule fois par n villes

- Exemple avec n = 36 :



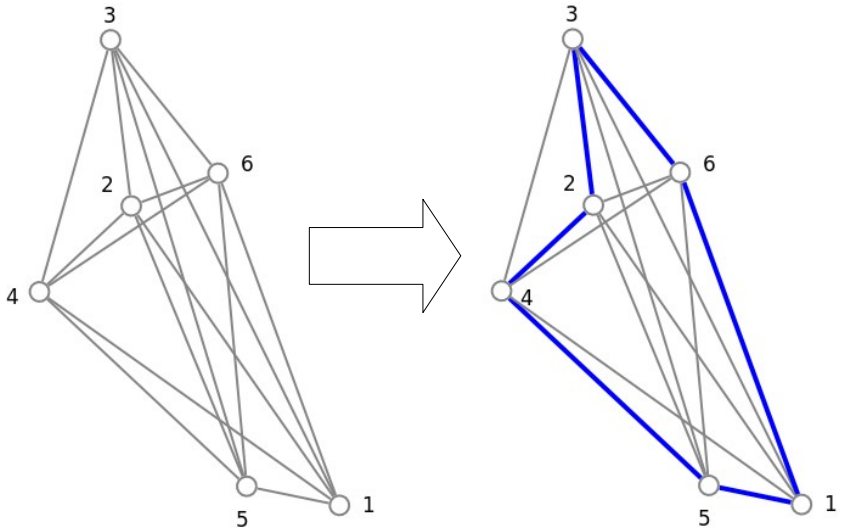
Formalisation du problème

Soit G un graphe d'ordre n avec :

- $V(G)$ l'ensemble des nœuds, chaque nœud modélisant une ville
- $E(G)$ l'ensemble des arcs. Le poids $w(e)$ associé à chaque arc $e \in E(G)$ modélise au coût de déplacement entre deux villes. Notons que les poids ne sont pas nécessairement symétriques.

Le problème du voyageur de commerce consiste à calculer un cycle hamiltonien de coût minimal dans G.

Exemple avec n = 6 :

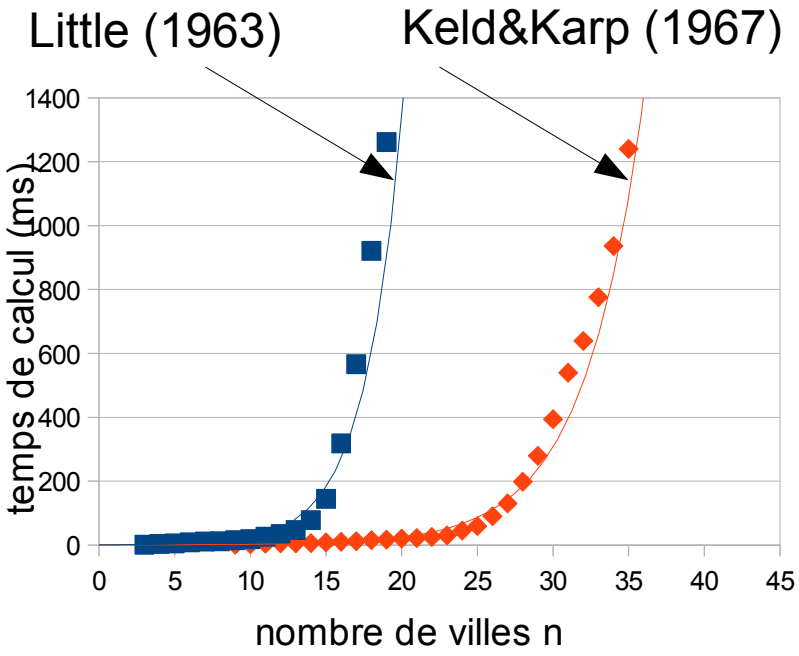


Algorithmes de résolution exacte

Garantissent une solution optimale, mais ne peuvent s'exécuter en un temps en temps polynomial en n.

- Force brute : complexité temporelle factorielle en n
- Programmation dynamique : complexité spatiale + temporelle exponentielle en n
- Branch & Bound : complexité temporelle exponentielle en n

Au mieux, le temps de calcul requis est exponentiel en n, mais des stratégies astucieuses et des structures de données optimisées permettent d'augmenter la taille de problème n traitable en un temps de calcul donné.



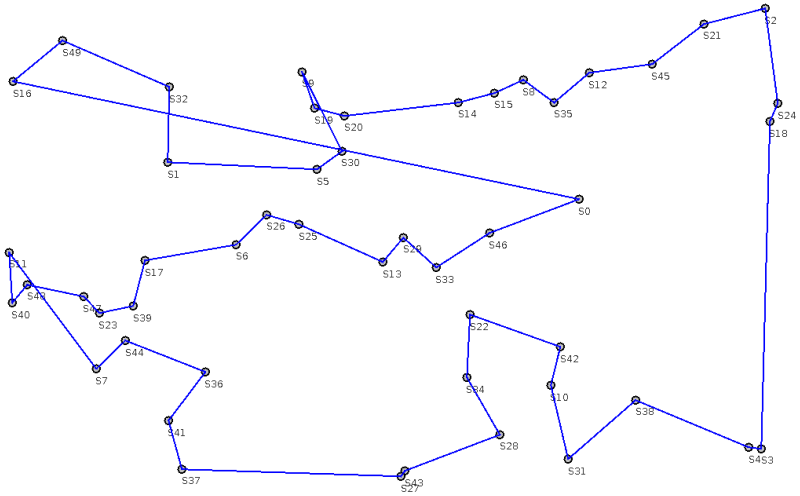
ISEN | école d'ingénieurs
LILLE

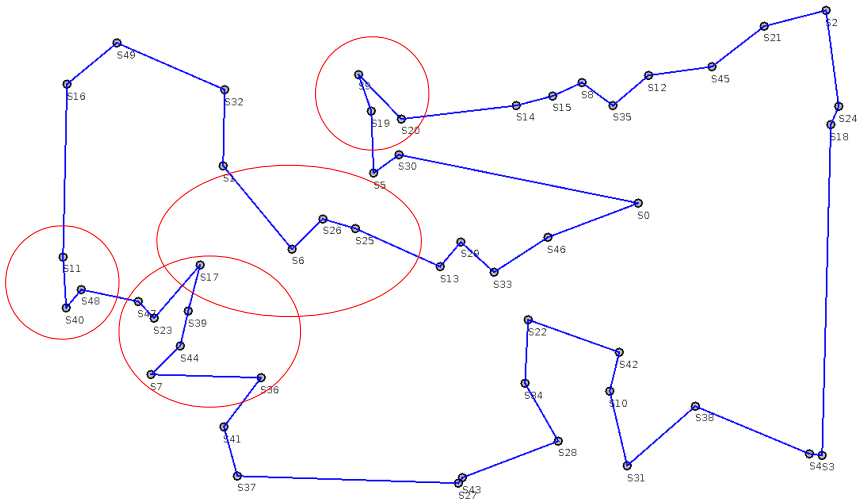
Algorithmes de résolution approchée

Se basent sur des heuristiques (règles empiriques)

- Heuristiques de construction (élaboration *gloutonne* une tournée)
- Heuristiques d'amélioration (modifications *locales* d'une tournée existante)
- Métaheuristiques (souvent stochastiques et/ou inspirées du monde du vivant)
- ...

Exemple :





Construction : Nearest Neighbor

→ choisit la ville la plus proche

Amélioration : 2-opt

→ « décroise » les trajets

ISEN | école d'ingénieurs
LILLE

Le branch & bound

Optimisation combinatoire

Un problème d'optimisation combinatoire consiste à minimiser/maximiser une fonction $f(x1,x2,...,xk)$

- Chaque variable xi appartient à un domaine Di fini et discret (donc énumérable)
- Des contraintes restreignent les valeurs utilisables dans Di (*réalisables*)

Exemple (programme linéaire) :

$min\ f(x1, x2) = 3x1 - 2x2$

Domaines :

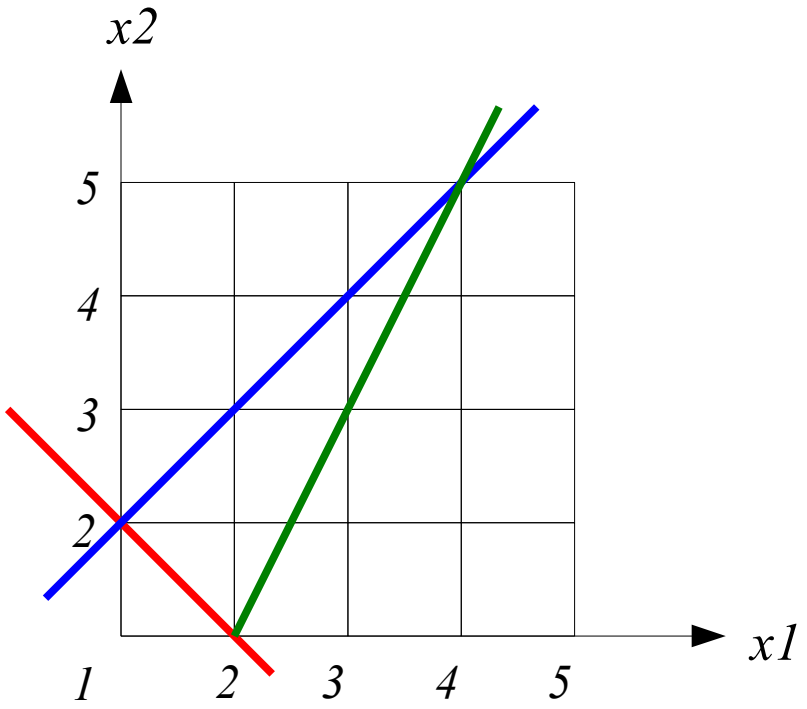
$x1$ et $x2$ dans $\{1,2,3,4\}$

Contraintes :

$x1 + x2 \geq 3$

$x2 - x1 \leq 1$

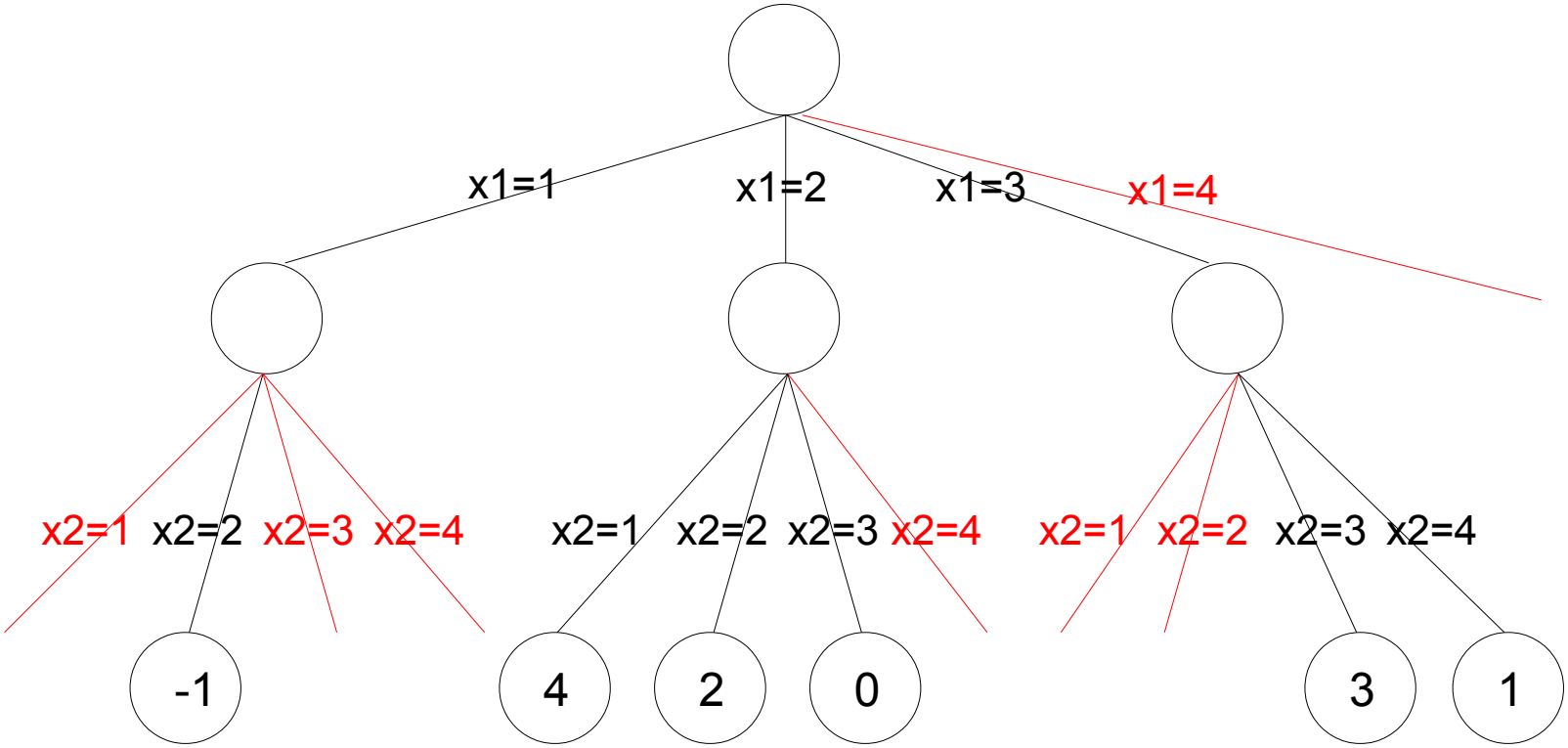
$2x1 - x2 \leq 3$



ISEN école d'ingénieurs
LILLE

Optimisation combinatoire

Résolution par énumération



Le branch & bound

Etapes

- *Branch* (séparation) : on sépare le problème initial en sous-problèmes (par exemple, en découpant les intervalles initiaux en sous-intervalles) ; on obtient un arbre dans lequel chaque nœud modélise un sous-problème

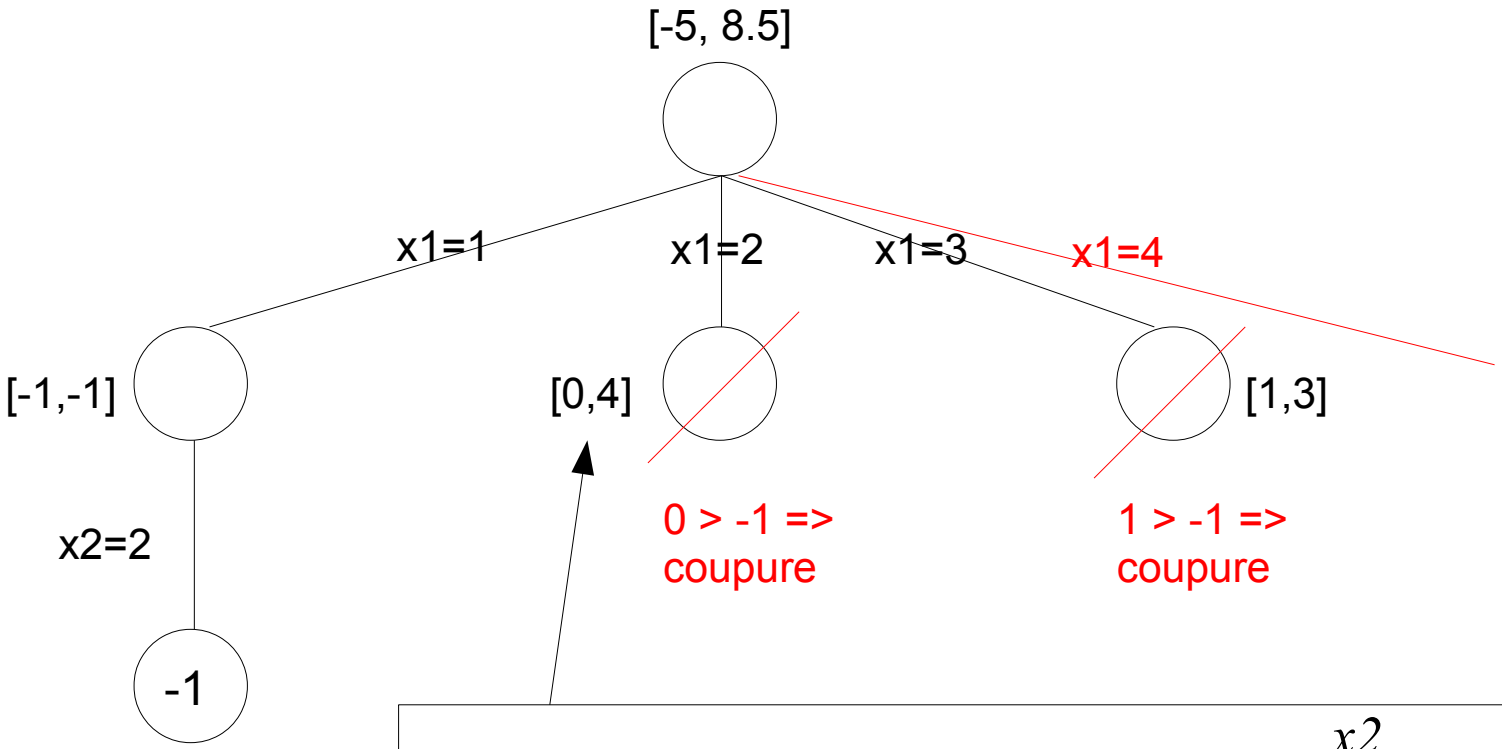
- *Bound* (évaluation) : pour chaque nœud, on évalue un intervalle pour la fonction objectif (par exemple, en propageant les intervalles des variables dans les contraintes)

- *Cut* (coupure) : si un nœud N a été évalué à [a,b] et s'il existe un autre nœud N' évalué à [c,d] tel que a > d, alors on peut couper la branche au niveau de N sans risque de manquer la solution optimale

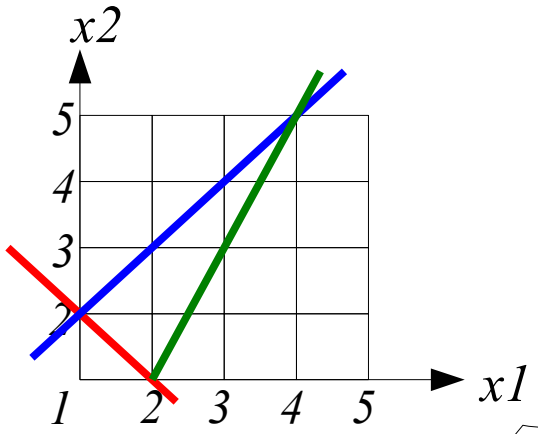
ISEN école d'ingénieurs LILLE

Optimisation combinatoire

Résolution par branch & bound



$x1=2 \Rightarrow x2 \in [1,3] \Rightarrow f \in [0,4]$

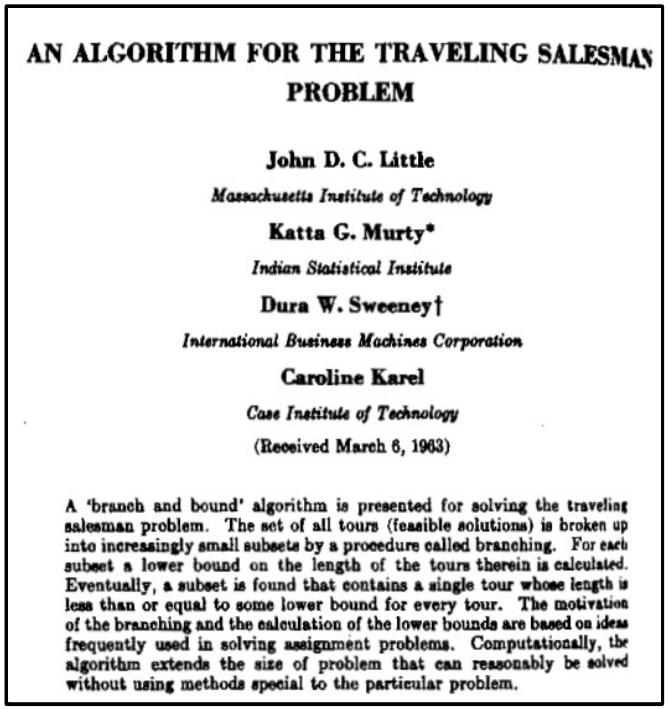


ISEN | école d'ingénieurs
LILLE

Principe de l'algorithme de Little

ISEN école d'ingénieurs
LILLE

L'algorithme de Little
Source



AN ALGORITHM FOR THE TRAVELING SALESMAN PROBLEM

John D. C. Little
Massachusetts Institute of Technology
Katta G. Murty*
Indian Statistical Institute
Dura W. Sweeney†
International Business Machines Corporation
Caroline Karel
Case Institute of Technology

(Received March 6, 1963)

A 'branch and bound' algorithm is presented for solving the traveling salesman problem. The set of all tours (feasible solutions) is broken up into increasingly small subsets by a procedure called branching. For each subset a lower bound on the length of the tours therein is calculated. Eventually, a subset is found that contains a single tour whose length is less than or equal to some lower bound for every tour. The motivation of the branching and the calculation of the lower bounds are based on ideas frequently used in solving assignment problems. Computationally, the algorithm extends the size of problem that can reasonably be solved without using methods special to the particular problem.

Little, J. D., Murty, K. G., Sweeney, D. W., & Karel, C. An algorithm for the traveling salesman problem. Operations research, 11(6), 972-989, 1963

Fichier PDF

L'algorithme de Little

Principe

L'algorithme de Little est un algorithme de résolution du TSP par Branch & Bound

- La *séparation* consiste à considérer l'inclusion ou l'exclusion d'un trajet (i,j) dans une tournée. Chaque séparation produisant deux branches, l'arbre de recherche est binaire (fig. 1b)

- L'*évaluation* fournit une borne inférieure du coût de la tournée en effectuant des opérations sur la matrice de coûts (fig 1a)

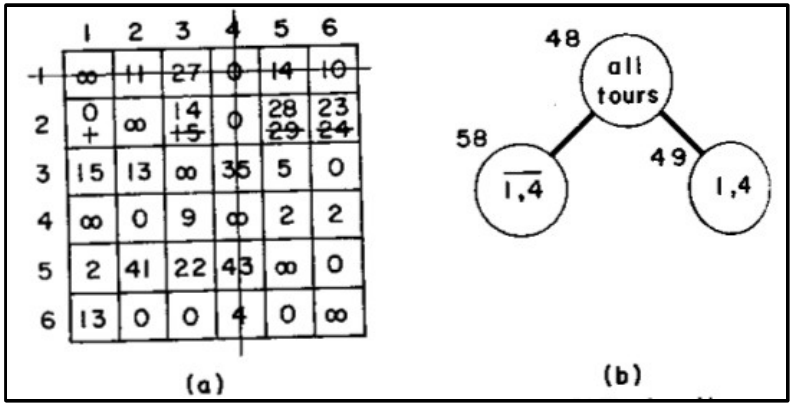


Figure 1 (tirée de l'article)

ISEN école d'ingénieurs
LILLE

L'algorithme de Little
Principe

- On développe l'arbre de façon préférentielle vers la droite (inclusion de trajets) pour rapidement aboutir à une tournée complète T, de coût c

→ *Dans la figure, T contient les trajets (1,4), (2,1), (5,6), (3,5), (4,3) et (6,2), ce qui donne T = (1,4,3,5,6,2,1), de coût 63*

- On développe ensuite les autres branches pour tenter de trouver une meilleure tournée que T. Dès qu'un nœud a un coup supérieur ou égal à c, on peut interrompre la branche

→ *Dans la figure, toutes les branches sont de coût ≥ 63, on est donc sûr que T est la tournée optimale*

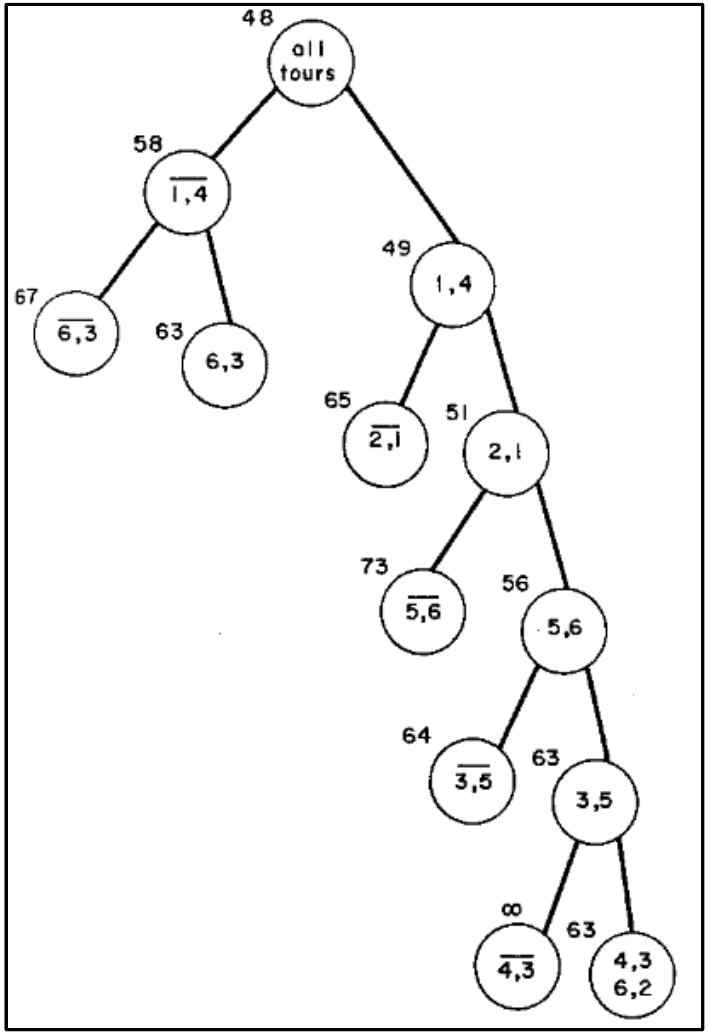


Figure 2 (tirée de l'article)

ISEN | école d'ingénieurs
LILLE

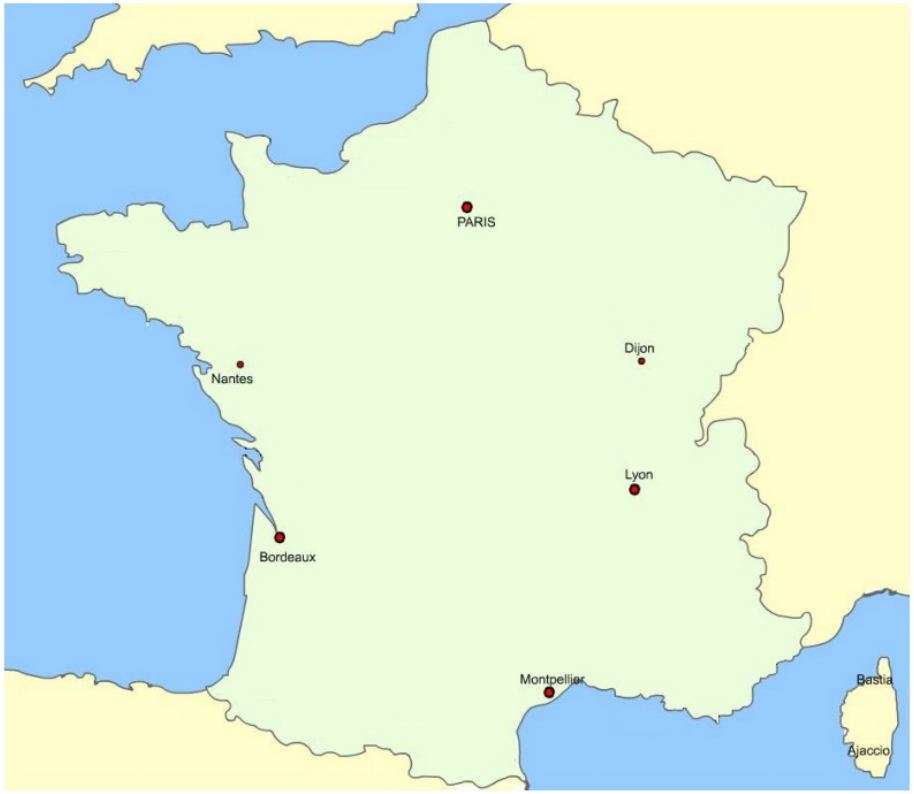
L'algorithme de Little par l'exemple

L'algorithme de Little
Déroulement sur un exemple

6 villes

B: Bordeaux
L: Lyon
N: Nantes
P: Paris
M: Montpellier
D: Dijon
60 tournées possibles

(120 si coûts asymétriques)



ISEN école d'ingénieurs
LILLE

L'algorithme de Little
Visite du nœud 1

Arbre de recherche

La racine va contenir une première borne inférieure pour le coût de la tournée optimale, tous les trajets étant encore possibles

nœud à visiter

nœud en cours de visite

nœud visité

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 1

Arbre de recherche

Matrice de coûts

	B	L	N	P	M	D
B		780	320	580	480	660
L	780		700	460	300	200
N	320	700		380	820	630
P	580	460	380		750	310
M	480	300	820	750		500
D	660	200	630	310	500	

Contient ici les distances en km pour chaque trajet possible (matrice symétrique)

 *nœud à visiter*

 *nœud en cours de visite*

 *nœud visité*

ISEN école d'ingénieurs
LILLE

L'algorithme de Little
Visite du nœud 1

Arbre de recherche

Matrice de coûts

	B	L	N	P	M	D
B		780	320	580	480	660
L	780		700	460	300	200
N	320	700		380	820	630
P	580	460	380		750	310
M	480	300	820	750		500
D	660	200	630	310	500	

Peut contenir d'autres types de coûts (temps de parcours, énergie dépensée...) la rendant asymétrique

nœud à visiter *nœud en cours de visite* *nœud visité*

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 1

Arbre de recherche

Matrice de coûts

	B	L	N	P	M	D	min ligne:
B		780	320	580	480	660	320
L	780		700	460	300	200	200
N	320	700		380	820	630	320
P	580	460	380		750	310	310
M	480	300	820	750		500	300
D	660	200	630	310	500		200

Réduction de la matrice: lignes

nœud à visiter — nœud en cours de visite — nœud visité

ISEN | école d'ingénieurs
LILLE

L'algorithme de Little
Visite du nœud 1

Arbre de recherche

Matrice de coûts

Suppression du min ligne par ligne

	B	L	N	P	M	D	min ligne:
B		460	0	260	160	340	320
L	580		500	260	100	0	200
N	0	380		60	500	310	320
P	270	150	70		440	0	310
M	180	0	520	450		200	300
D	460	0	430	110	300		200

Total supprimé: 1650

Réduction de la matrice: lignes

nœud à visiter — nœud en cours de visite — nœud visité

ISEN | école d'ingénieurs
LILLE

L'algorithme de Little
Visite du nœud 1

Arbre de recherche

Matrice de coûts

	B	L	N	P	M	D
B		460	0	200	60	340
L	580		500	200	0	0
N	0	380		0	400	310
P	270	150	70		340	0
M	180	0	520	390		200
D	460	0	430	50	200	

Total supprimé: 160

min colonne:	0	0	0	60	100	0

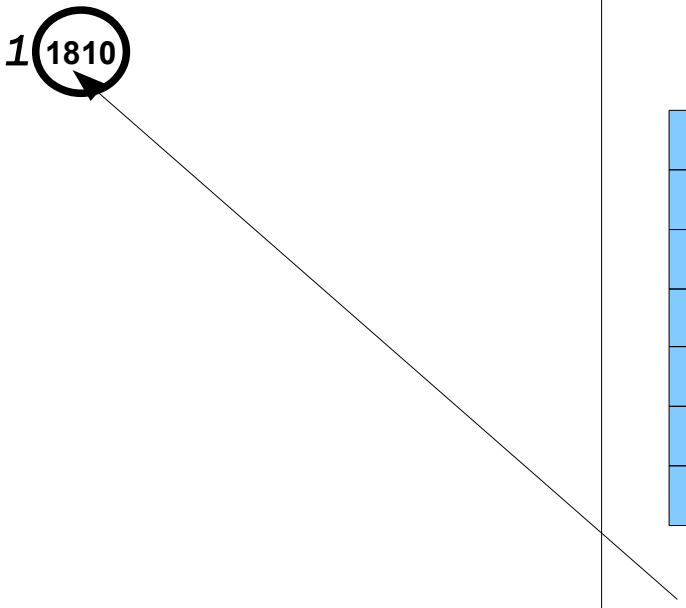
Réduction de la matrice: colonnes

nœud à visiter — nœud en cours de visite — nœud visité

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 1

Arbre de recherche



Matrice de coûts

	B	L	N	P	M	D
B		460	0	200	60	340
L	580		500	200	0	0
N	0	380		0	400	310
P	270	150	70		340	0
M	180	0	520	390		200
D	460	0	430	50	200	

Évaluation par défaut du nœud:
1650 + 160 = 1810

nœud à visiter — *nœud en cours de visite* — *nœud visité*

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 1

Arbre de recherche

Matrice de coûts

Regret = min ligne + min colonne

	B	L	N	P	M	D	min ligne :
B		460	0**(130)**	200	60	340	60
L	580		500	200	0	0	
N	0	380		0	400	310	
P	270	150	70		340	0	
M	180	0	520	390		200	
D	460	0	430	50	200		

min colonne: 70

Calcul des regrets
dans toutes les cases de valeur 0

nœud à visiter — *nœud en cours de visite* — *nœud visité*

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 1

Arbre de recherche

1 (1810)

Matrice de coûts

Regret = min ligne + min colonne

	B	L	N	P	M	D
B		460	0(**130**)	200	60	340
L	580		500	200	0(**60**)	0
N	0	380		0	400	310
P	270	150	70		340	0
M	180	0	520	390		200
D	460	0	430	50	200	

min ligne : 0

min colonne: 60

Calcul des regrets
dans toutes les cases de valeur 0

nœud à visiter — *nœud en cours de visite* — *nœud visité*

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 1

Arbre de recherche

Matrice de coûts

	B	L	N	P	M	D
B		460	0**(130)**	200	60	340
L	580		500	200	0**(60)**	0**(0)**
N	0**(180)**	380		0**(50)**	400	310
P	270	150	70		340	0**(70)**
M	180	0**(180)**	520	390		200
D	460	0**(50)**	430	50	200	

Calcul des regrets
dans toutes les cases de valeur 0

nœud à visiter — *nœud en cours de visite* — *nœud visité*

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 1

Arbre de recherche

1 (1810)

Matrice de coûts

	B	L	N	P	M	D
B		460	0**(130)**	200	60	340
L	580		500	200	0**(60)**	0**(0)**
N	0**(180)**	380		0**(50)**	400	310
P	270	150	70		340	0**(70)**
M	180	0**(180)**	520	390		200
D	460	0**(50)**	430	50	200	

On privilégie l'étude du trajet de regret maximal: Nantes-Bordeaux

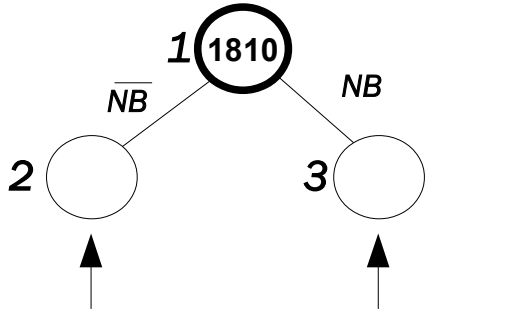
(en cas d'égalité: choix arbitraire)

nœud à visiter — *nœud en cours de visite* — *nœud visité*

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 1

Arbre de recherche



Matrice de coûts

	B	L	N	P	M	D
B		460	0**(130)**	200	60	340
L	580		500	200	0**(60)**	0**(0)**
N	0**(180)**	380		0**(50)**	400	310
P	270	150	70		340	0**(70)**
M	180	0**(180)**	520	390		200
D	460	0**(50)**	430	50	200	

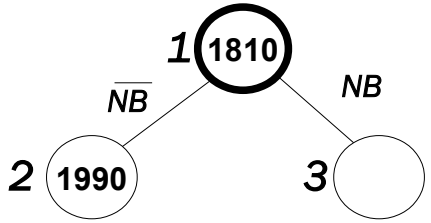
On crée deux fils (2 et 3) correspondant à l'inclusion ou à l'exclusion du trajet Nantes-Bordeaux dans la tournée

nœud à visiter — *nœud en cours de visite* — *nœud visité*

ISEN | école d'ingénieurs
LILLE

L'algorithme de Little
Visite du nœud 1

Arbre de recherche



= 1810 + 180 (regret)

Matrice de coûts

	B	L	N	P	M	D
B		460	0**(130)**	200	60	340
L	580		500	200	0**(60)**	0**(0)**
N	0**(180)**	380		0**(50)**	400	310
P	270	150	70		340	0**(70)**
M	180	0**(180)**	520	390		200
D	460	0**(50)**	430	50	200	

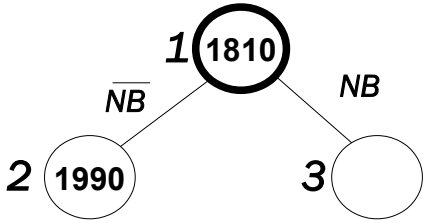
L'évaluation d'un nœud de type "trajet exclu" est déjà connue, sans autre calcul: il s'agit de la valeur du père + la valeur de regret

nœud à visiter — *nœud en cours de visite* — *nœud visité*

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 1

Arbre de recherche



Matrice de coûts

	B	L	N	P	M	D
B		460	0(**130**)	200	60	340
L	580		500	200	0(**60**)	0(**0**)
N	0(**180**)	380		0(**50**)	400	310
P	270	150	70		340	0(**70**)
M	180	0(**180**)	520	390		200
D	460	0(**50**)	430	50	200	

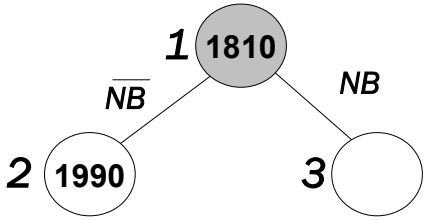
L'évaluation d'un nœud de type "trajet inclus" correspond à l'évaluation par défaut de ce nœud (obtenue par réduction de la matrice des liaisons encore possibles)

nœud à visiter — *nœud en cours de visite* — *nœud visité*

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 1

Arbre de recherche



Matrice de coûts

	B	L	N	P	M	D
B		460	0(**130**)	200	60	340
L	580		500	200	0(**60**)	0(**0**)
N	0(**180**)	380		0(**50**)	400	310
P	270	150	70		340	0(**70**)
M	180	0(**180**)	520	390		200
D	460	0(**50**)	430	50	200	

On visite en priorité les nœuds de type "trajet inclus". Les nœuds de type "trajet exclu" (ici 2) sont empilés pour une visite ultérieure

Pile

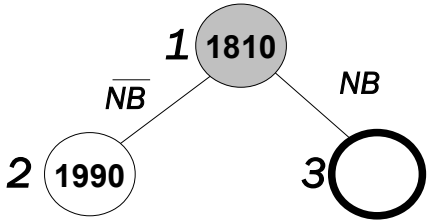
2

nœud à visiter — nœud en cours de visite — nœud visité

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 3

Arbre de recherche



Matrice de coûts

	L	N	P	M	D
B	460	0	200	60	340
L		500	200	0	0
P	150	70		340	0
M	0	520	390		200
D	0	430	50	200	

On manipule une nouvelle matrice: la ligne de la ville départ (Nantes) et la colonne de la ville d'arrivée (Bordeaux) du trajet ont été supprimées.

nœud à visiter

nœud en cours de visite

nœud visité

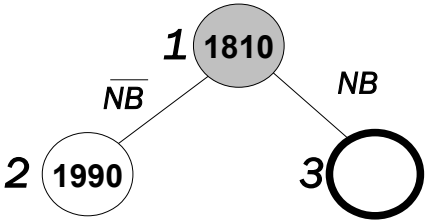
Pile

2

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 3

Arbre de recherche



Matrice de coûts

	L	N	P	M	D
B	460		200	60	340
L		500	200	0	0
P	150	70		340	0
M	0	520	390		200
D	0	430	50	200	

On supprime tous les trajets aboutissent à des sous-tournées incomplètes (n'incluant pas toutes les villes)
→ ici BN, aboutissant à la sous-tournée NB-BN

nœud à visiter *nœud en cours de visite* *nœud visité*

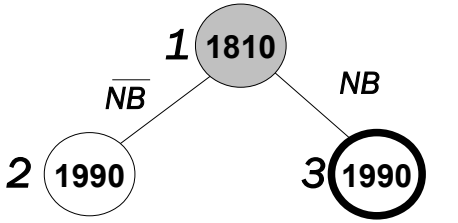
Pile

2

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 3

Arbre de recherche



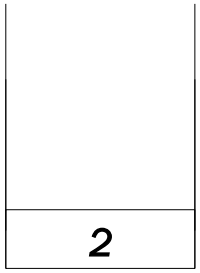
= 1810 + 180

Matrice de coûts

	L	N	P	M	D	supprimé:
B	400		90	0	280	60
L		430	150	0	0	0
P	150	0		340	0	0
M	0	450	340		200	0
D	0	360	0	200		0
supprimé:	0	70	50	0	0	Total supprimé: 180

Réduction de la matrice

Pile

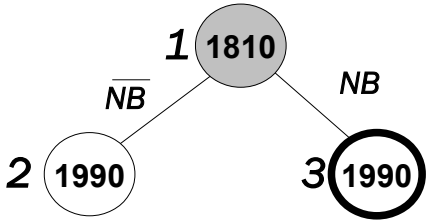


nœud à visiter — nœud en cours de visite — nœud visité

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 3

Arbre de recherche



Matrice de coûts

	L	N	P	M	D
B	400		90	0(**140**)	280
L		430	150	0(**0**)	0(**0**)
P	150	0(**360**)		340	0(**0**)
M	0(**200**)	450	340		200
D	0(**0**)	360	0(**90**)	200	

Calcul des regrets

nœud à visiter

nœud en cours de visite

nœud visité

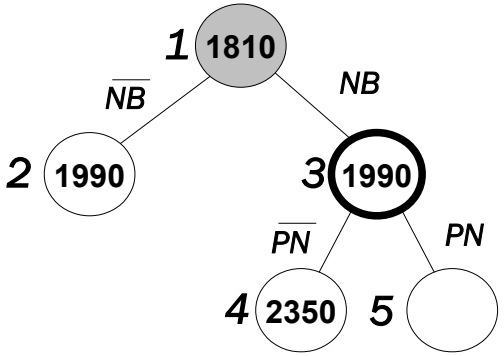
Pile

2

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 3

Arbre de recherche



= 1990 + 360

Matrice de coûts

	L	N	P	M	D
B	400		90	0(**140**)	280
L		430	150	0(**0**)	0(**0**)
P	150	0(**360**)		340	0(**0**)
M	0(**200**)	450	340		200
D	0(**0**)	360	0(**90**)	200	

Regret maximal: Paris-Nantes (360)

- *Création des fils 4 et 5.*
- *Affectation de la valeur de 4*
- *Empilage de 4*

Pile

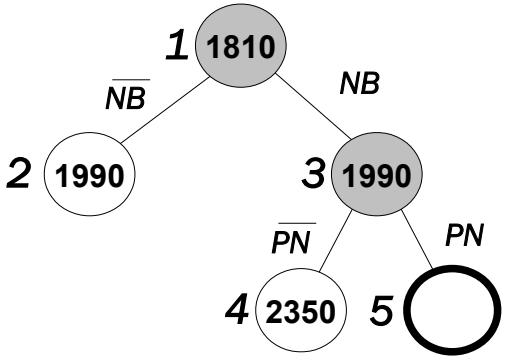
4
2

nœud à visiter — nœud en cours de visite — nœud visité

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 5

Arbre de recherche



Matrice de coûts

	L	P	M	D
B	400	90	0	280
L		150	0	0
M	0	340		200
D	0	0	200	

Pile

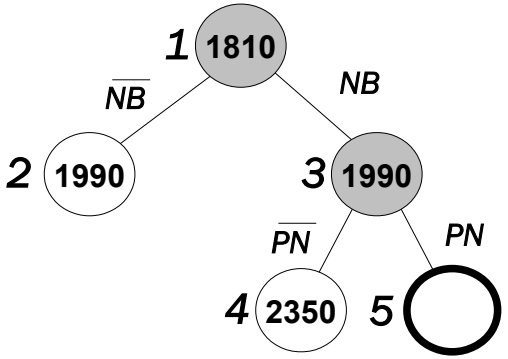
4
2

nœud à visiter — nœud en cours de visite — nœud visité

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 5

Arbre de recherche



Matrice de coûts

	L	P	M	D
B	400		0	280
L		150	0	0
M	0	340		200
D	0	0	200	

Élimination du trajet BP

(aboutissant à la sous-tournée PN-NB-BP)

Pile

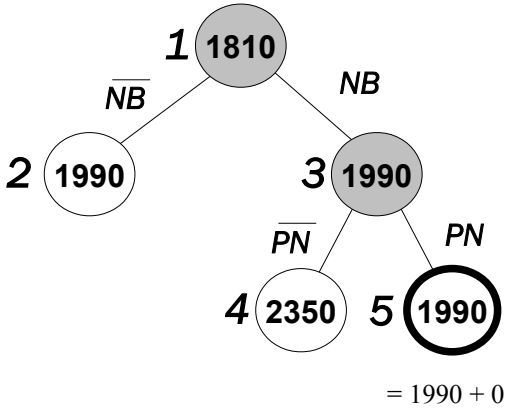
4
2

nœud à visiter — nœud en cours de visite — nœud visité

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 5

Arbre de recherche



Matrice de coûts

	L	P	M	D	supprimé:
B	400		0	280	0
L		150	0	0	0
M	0	340		200	0
D	0	0	200		0
supprimé:	0	0	0	0	Total supprimé: 0

Réduction de la matrice

nœud à visiter — nœud en cours de visite — nœud visité

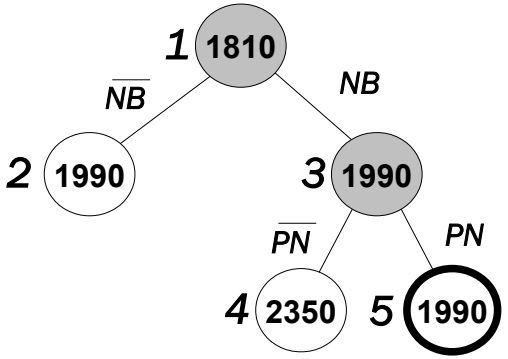
Pile

4
2

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 5

Arbre de recherche



Matrice de coûts

	L	P	M	D
B	400		0(**280**)	280
L		150	0(**0**)	0(**200**)
M	0(**200**)	340		200
D	0(**0**)	0(**150**)	200	

Calcul des regrets

nœud à visiter — nœud en cours de visite — nœud visité

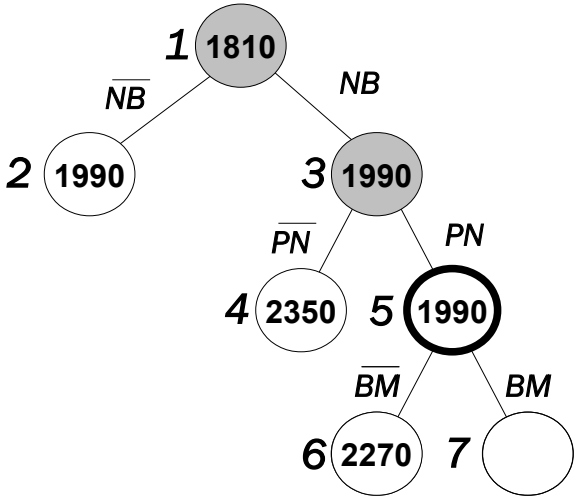
Pile

4
2

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 5

Arbre de recherche



= 1990 + 280

Matrice de coûts

	L	P	M	D
B	400		0(**280**)	280
L		150	0(**0**)	0(**200**)
M	0(**200**)	340		200
D	0(**0**)	0(**150**)	200	

Regret maximal: Bordeaux-Montpellier (280)

- *Création des fils 6 et 7.*
- *Affectation de la valeur de 6*
- *Empilage de 6*

Pile

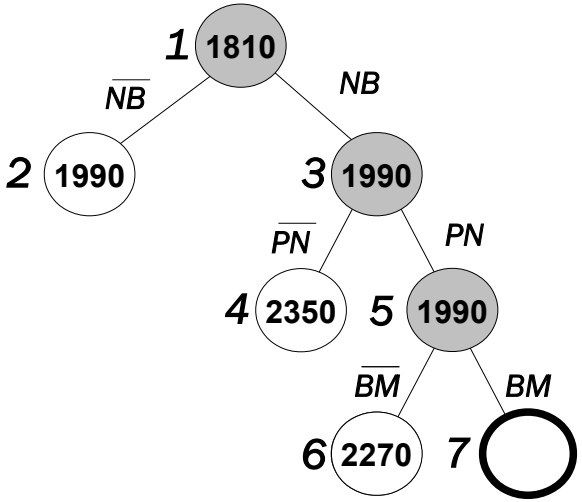
6
4
2

nœud à visiter — *nœud en cours de visite* — *nœud visité*

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 7

Arbre de recherche



Matrice de coûts

	L	P	D
L		150	0
M	0		200
D	0	0	

Élimination du trajet MP

(aboutissant à la sous-tournée PN-NB-BM-MP)

Pile

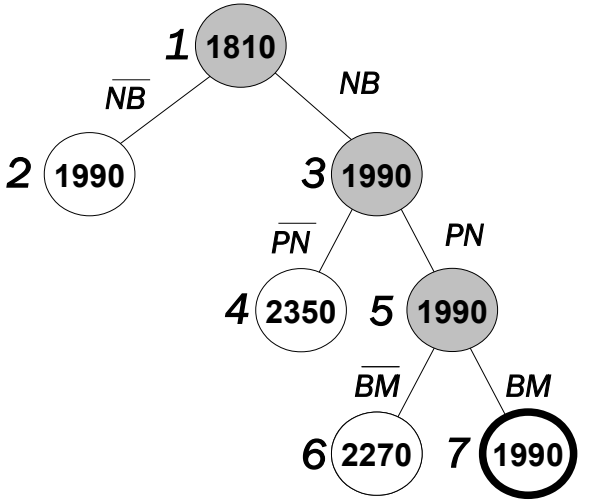
6
4
2

nœud à visiter — nœud en cours de visite — nœud visité

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 7

Arbre de recherche



= 1990 + 0

Matrice de coûts

	L	P	D	supprimé:
L		150	0	0
M	0		200	0
D	0	0		0
supprimé:	0	0	0	Total supprimé: 0

Réduction de la matrice

Pile

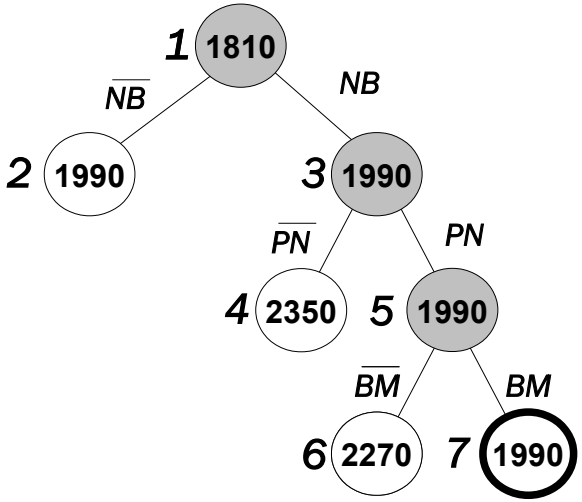
6
4
2

nœud à visiter — nœud en cours de visite — nœud visité

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 7

Arbre de recherche



Matrice de coûts

	L	P	D
L		150	0**(350)**
M	0**(200)**		200
D	0**(0)**	0**(150)**	

Calcul des regrets

nœud à visiter

nœud en cours de visite

nœud visité

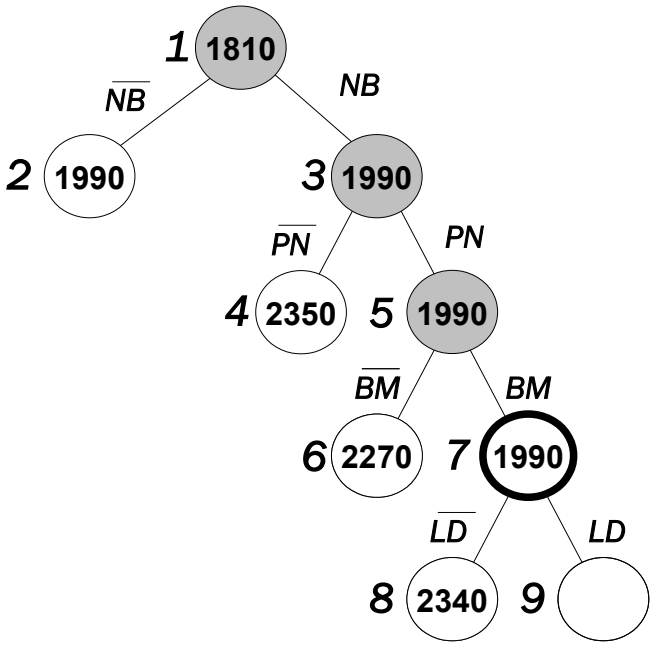
Pile

6
4
2

ISEN | école d'ingénieurs
LILLE

L'algorithme de Little
Visite du nœud 7

Arbre de recherche



= 1990 + 350

Matrice de coûts

	L	P	D
L		150	0**(350)**
M	0**(200)**		200
D	0**(0)**	0**(150)**	

Regret maximal: Lyon-Dijon (350)

- *Création des fils 8 et 9.*
- *Affectation de la valeur de 8*
- *Empilage de 8*

Pile

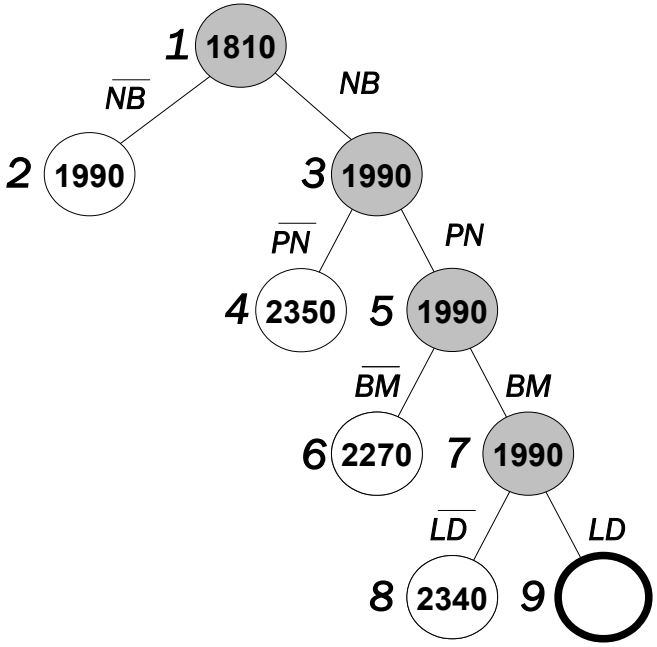
8
6
4
2

nœud à visiter — *nœud en cours de visite* — *nœud visité*

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 9

Arbre de recherche



Matrice de coûts

	L	P
M	0	
D	0	0

nœud à visiter

nœud en cours de visite

nœud visité

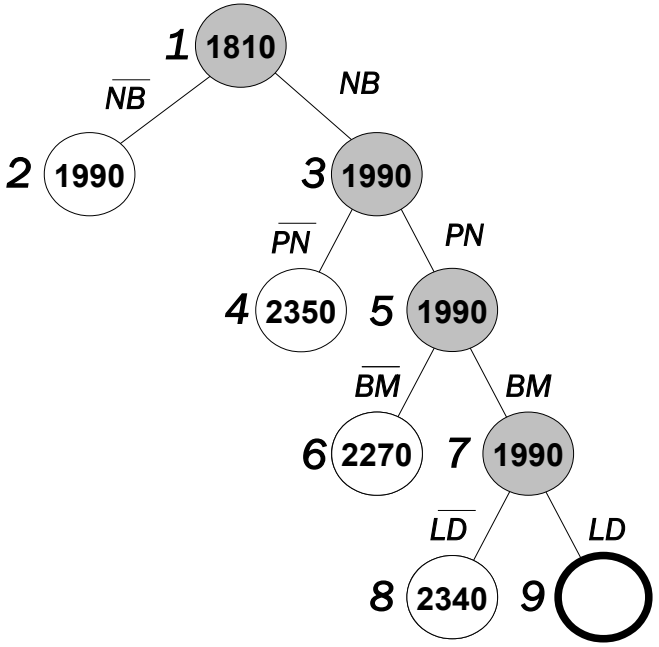
Pile

8
6
4
2

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 9

Arbre de recherche



Matrice de coûts

	L	P
M	0	
D		0

Élimination du trajet DL

(aboutissant à la sous-tournée LD-DL)

Pile

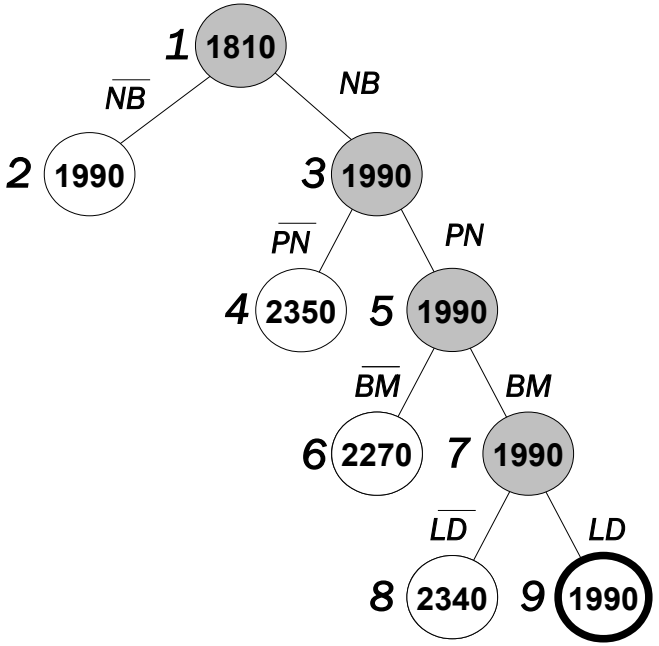
8
6
4
2

nœud à visiter — nœud en cours de visite — nœud visité

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 9

Arbre de recherche



= 1990 + 0

Matrice de coûts

	L	P	supprimé:
M	0		0
D		0	0
supprimé:	0	0	Total supprimé: 0

Réduction de la matrice

Pile

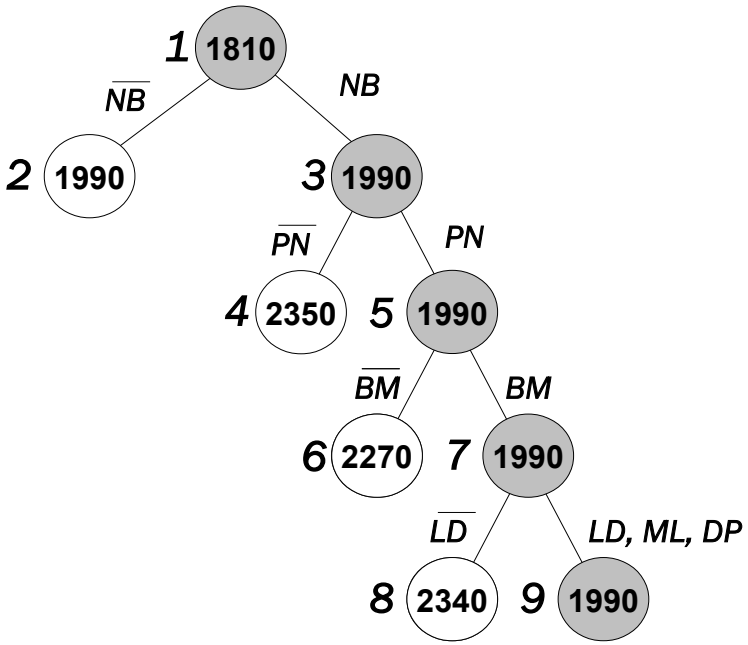
8
6
4
2

nœud à visiter — nœud en cours de visite — nœud visité

ISEN | école d'ingénieurs
LILLE

L'algorithme de Little
Visite du nœud 9

Arbre de recherche



Matrice de coûts

	L	P
M	0	
D		0

Il n'y a plus de choix possible: il faut intégrer les trajets ML et DP pour finir la tournée

Pile

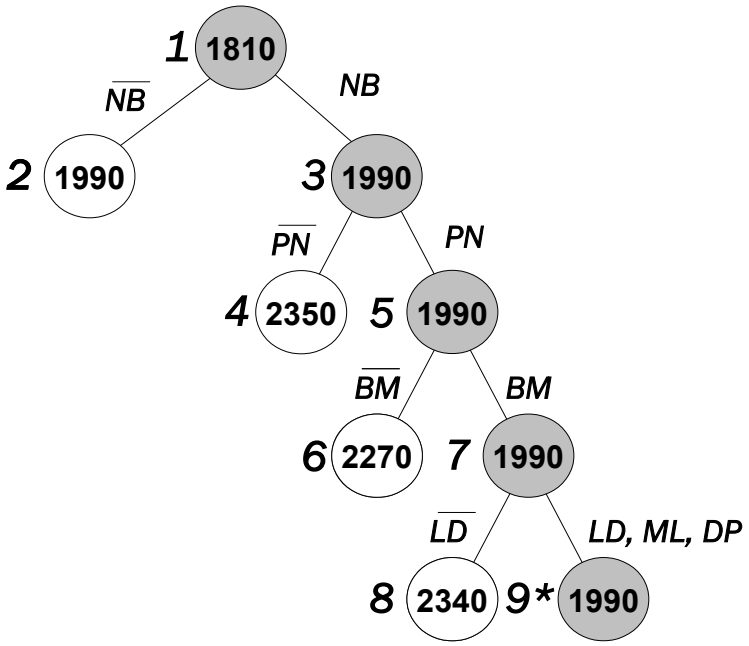
8
6
4
2

nœud à visiter — nœud en cours de visite — nœud visité

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Premier résultat

Arbre de recherche



Branche terminée.

Une première solution trouvée, contenant les trajets:

NB, PN, BM, LD, ML et DP

On peut donc construire une tournée au départ de n'importe quelle ville, par exemple Bordeaux:

BM – ML – LD – DP – PN – NB

Valeur de référence (*): 1990 km

Pile

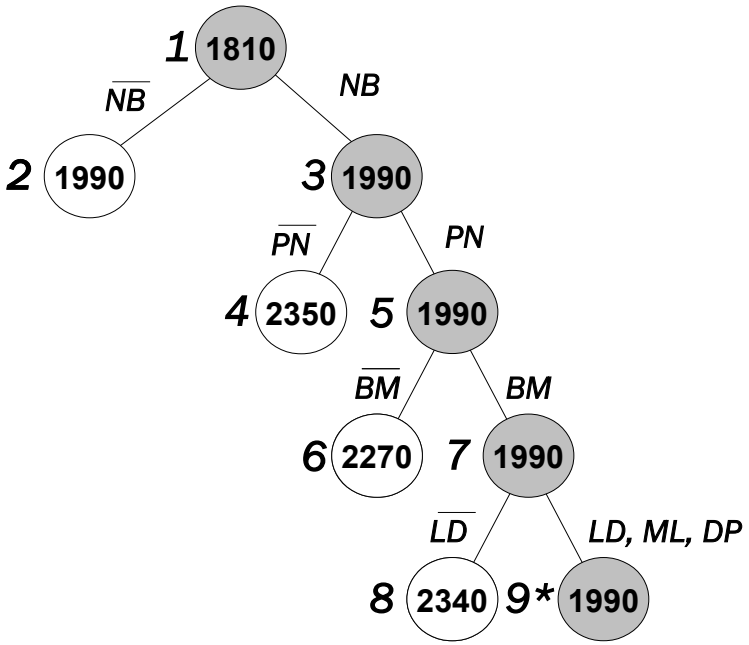
8
6
4
2

nœud à visiter — *nœud en cours de visite* — *nœud visité*

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Backtracking

Arbre de recherche



On visite à présent les nœuds empilés pour tenter d'améliorer cette première solution.

nœud à visiter

nœud en cours de visite

nœud visité

* référence :1990

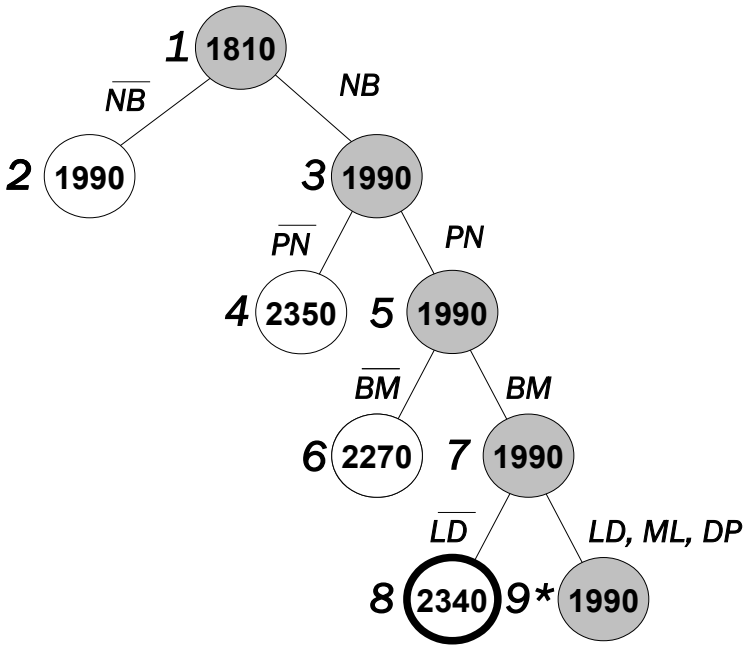
Pile

8
6
4
2

ISEN école d'ingénieurs
LILLE

L'algorithme de Little
Visite du nœud 8

Arbre de recherche



Valeur du nœud 8: 2340 ≥ référence.

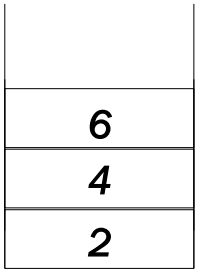
nœud à visiter

nœud en cours de visite

nœud visité

* référence :1990

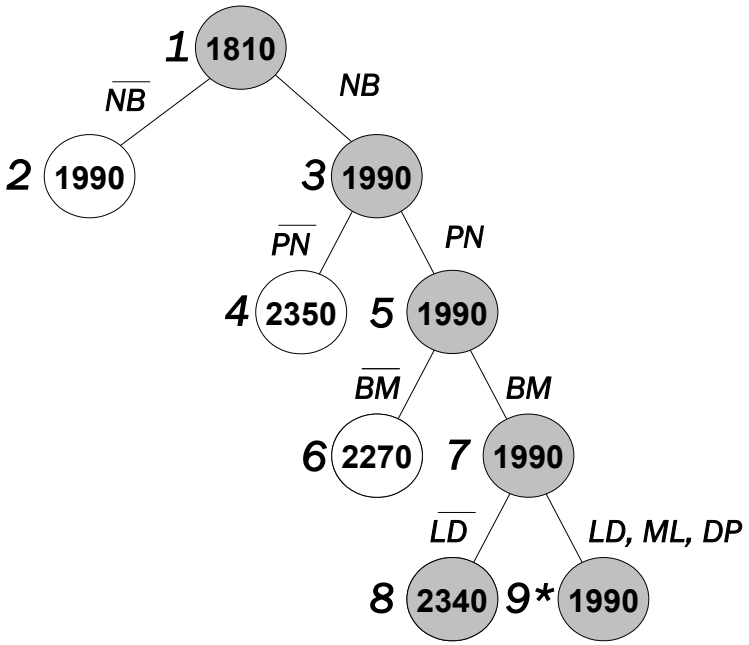
Pile



ISEN école d'ingénieurs
LILLE

L'algorithme de Little
Visite du nœud 8

Arbre de recherche



Valeur du nœud 8: 2320 ≥ référence.

Pas d'amélioration possible.

nœud à visiter — nœud en cours de visite — nœud visité

* référence :1990

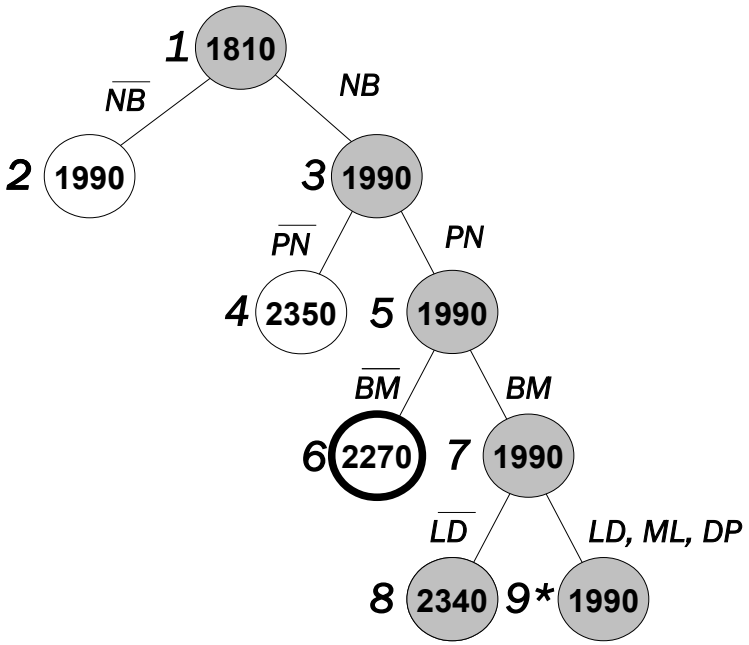
Pile

6
4
2

ISEN école d'ingénieurs
LILLE

L'algorithme de Little
Visite du nœud 6

Arbre de recherche



Valeur du nœud 6: 2270 ≥ référence.

nœud à visiter

nœud en cours de visite

nœud visité

* référence :1990

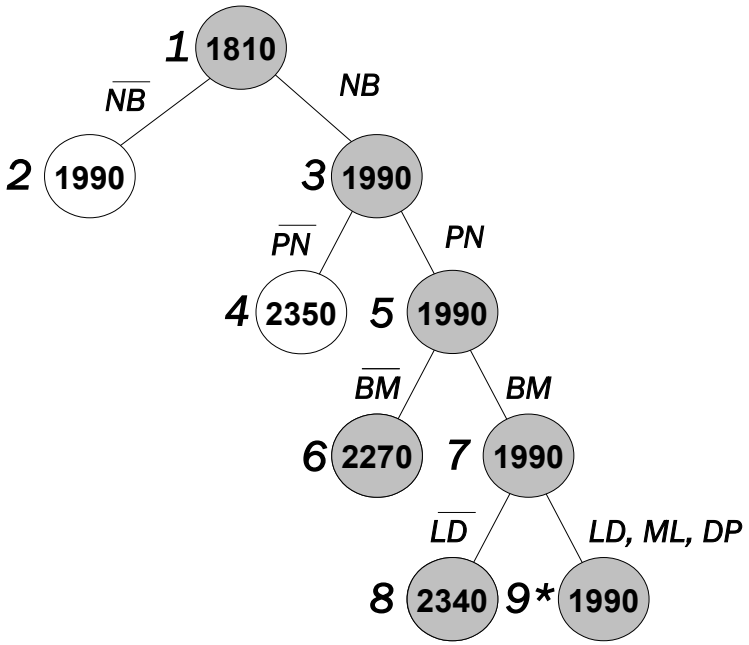
Pile

4
2

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 6

Arbre de recherche



Valeur du nœud 6: 2270 ≥ référence.

Pas d'amélioration possible.

nœud à visiter — nœud en cours de visite — nœud visité

* référence :1990

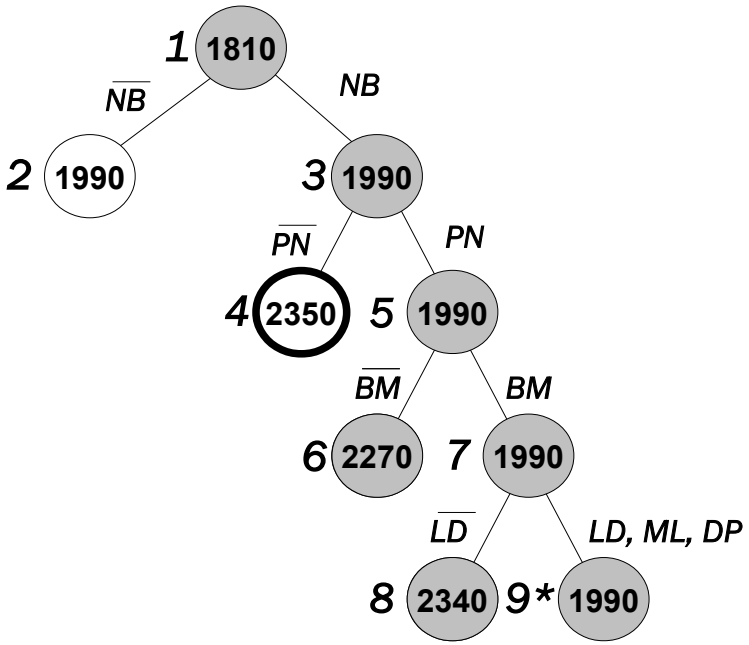
Pile

4
2

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 4

Arbre de recherche



Valeur du nœud 4: 2350 ≥ référence.

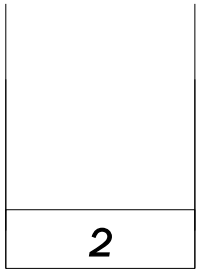
nœud à visiter

nœud en cours de visite

nœud visité

** référence :1990*

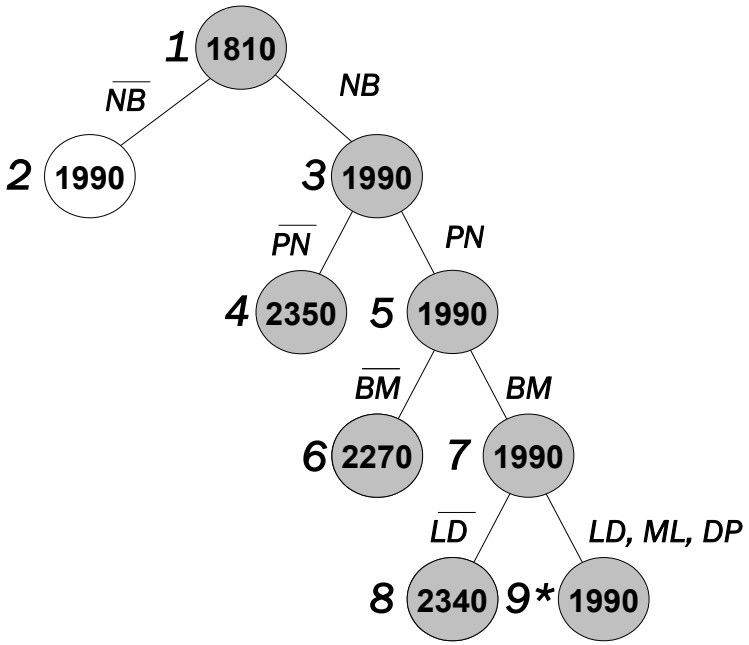
Pile



ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 4

Arbre de recherche



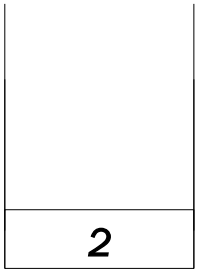
Valeur du nœud 4: 2350 ≥ référence.

Pas d'amélioration possible.

nœud à visiter — nœud en cours de visite — nœud visité

* référence :1990

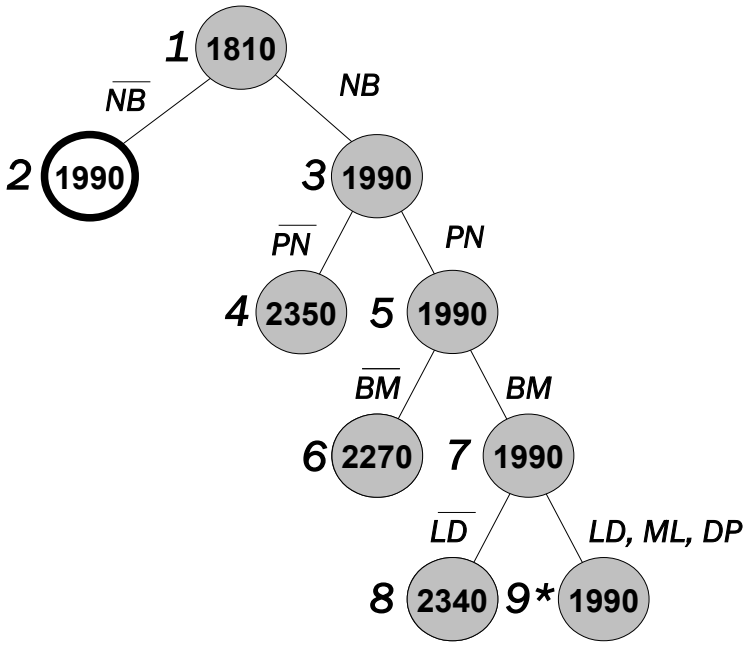
Pile



ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 2

Arbre de recherche



Valeur du nœud 2: 1990 ≥ référence.

nœud à visiter

nœud en cours de visite

nœud visité

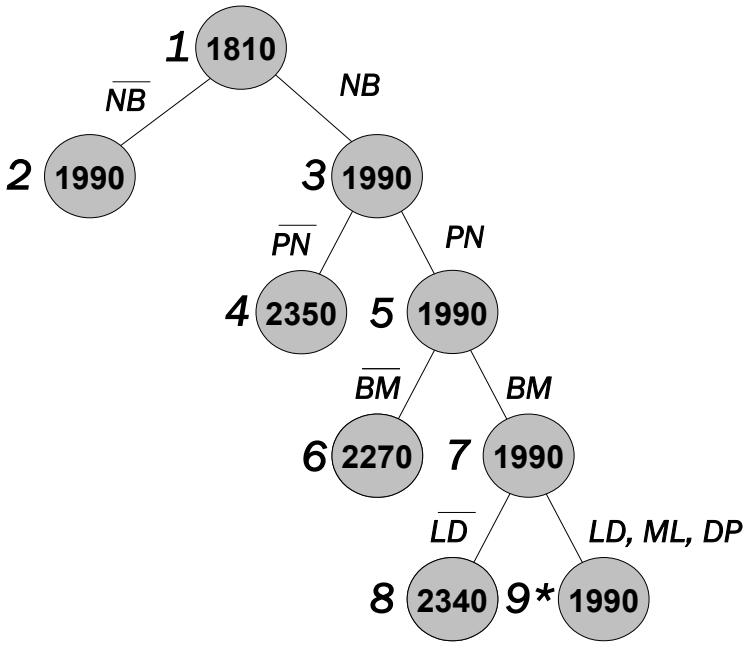
** référence :1990*

Pile

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Visite du nœud 2

Arbre de recherche



Valeur du nœud 2: 1990 ≥ référence.

Pas d'amélioration possible.

nœud à visiter

nœud en cours de visite

nœud visité

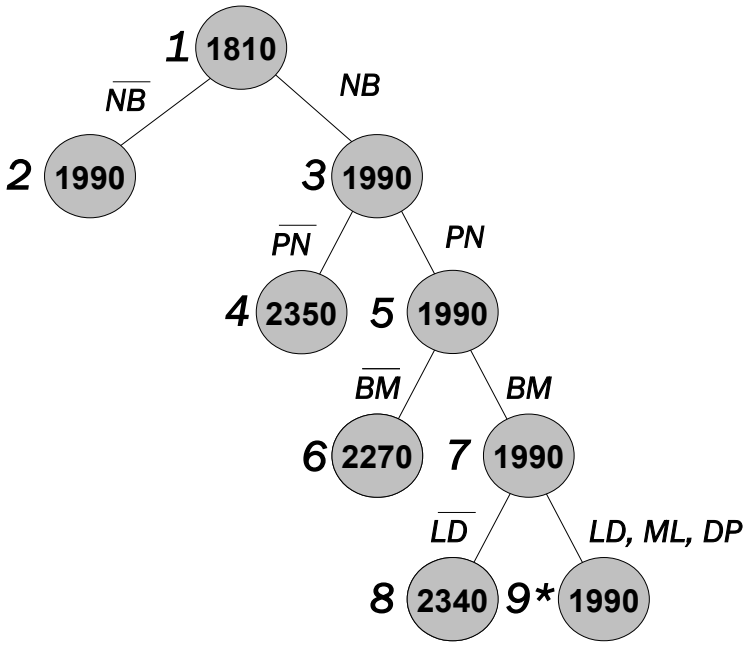
* référence :1990

Pile

ISEN école d'ingénieurs
LILLE

L'algorithme de Little
Solution finale

Arbre de recherche



Fin de l'algorithme.

Tournée optimale au départ de Bordeaux:

BM – ML – LD – DP – PN – NB

Valeur: 1990 km

#nœuds visités: 9 << 565

Nombre maximal de nœuds visités
(cf. exercice d'approfondissement)

nœud à visiter

nœud en cours de visite

nœud visité

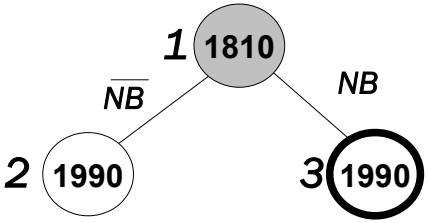
** référence :1990*

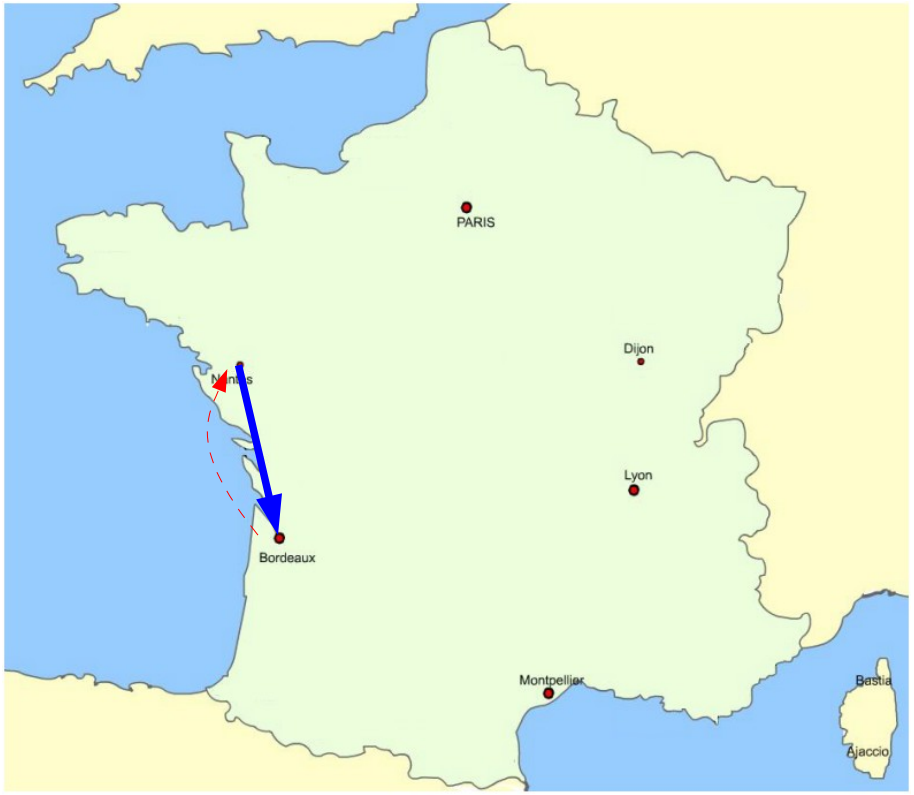
Pile

ISEN école d'ingénieurs LILLE

L'algorithme de Little
En images

Arbre de recherche





Trajet inclus dans la tournée

Trajet parasite

nœud à visiter

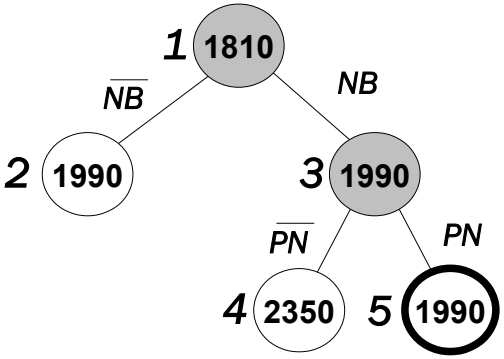
nœud en cours de visite

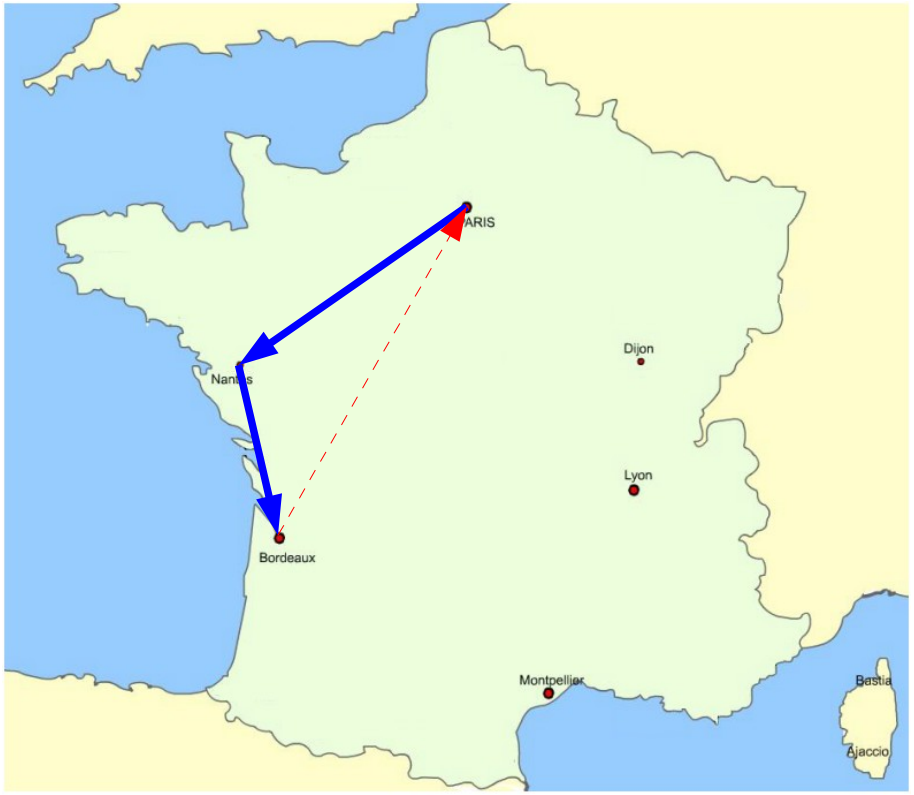
nœud visité

ISEN | école d'ingénieurs
LILLE

L'algorithme de Little
En images

Arbre de recherche





Trajet inclus dans la tournée

Trajet parasite

nœud à visiter

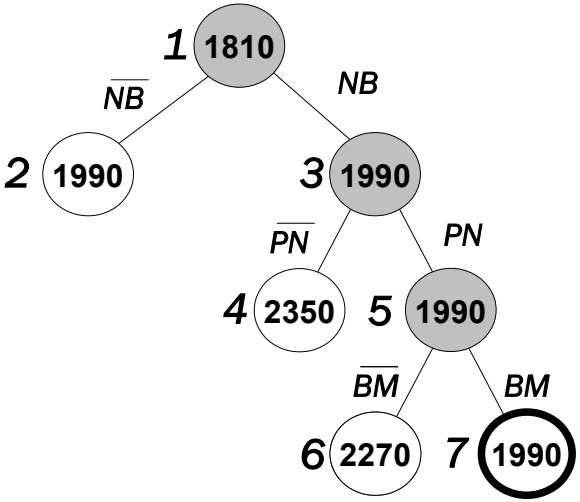
nœud en cours de visite

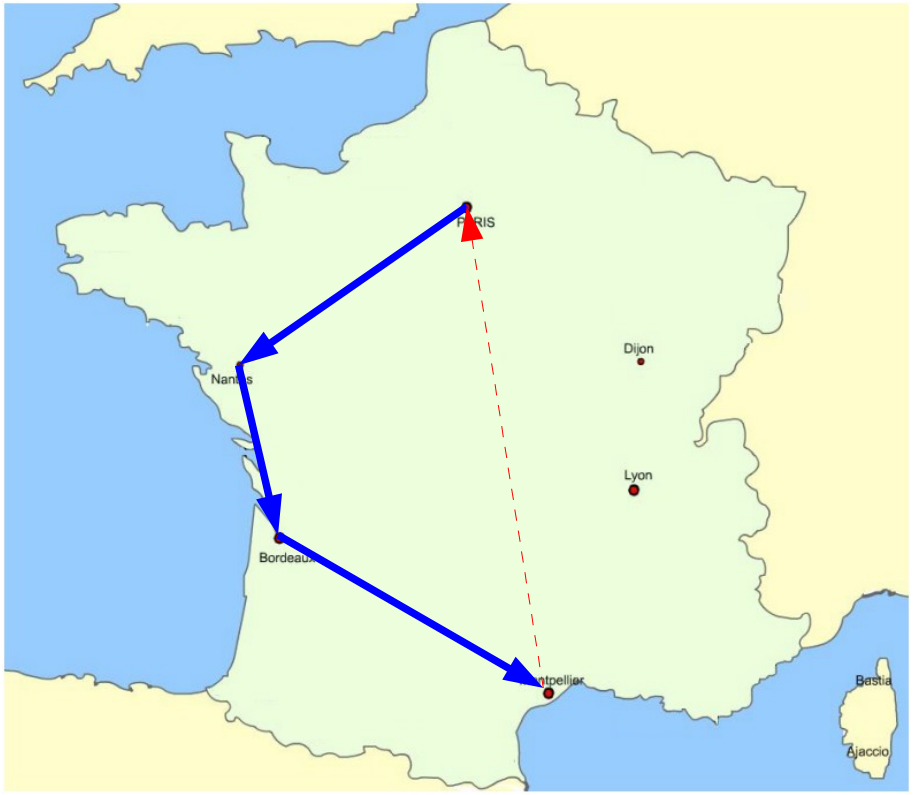
nœud visité

ISEN | école d'ingénieurs
LILLE

L'algorithme de Little
En images

Arbre de recherche





Trajet inclus dans la tournée

Trajet parasite

nœud à visiter

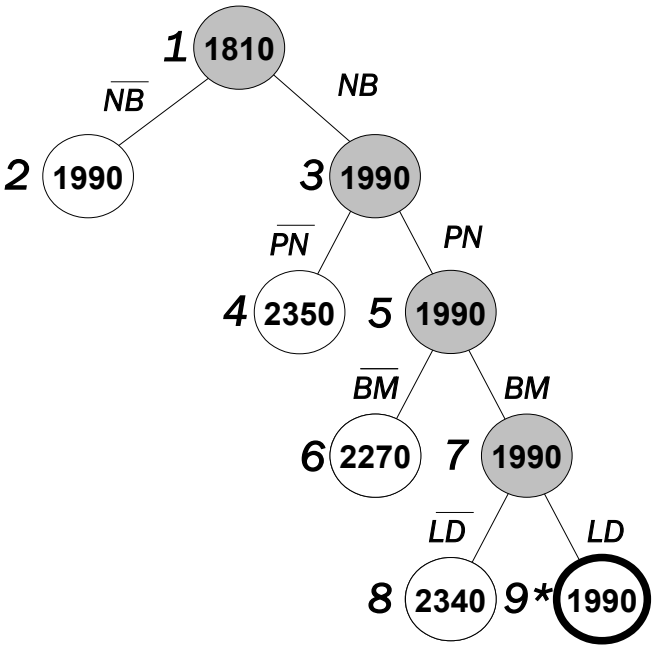
nœud en cours de visite

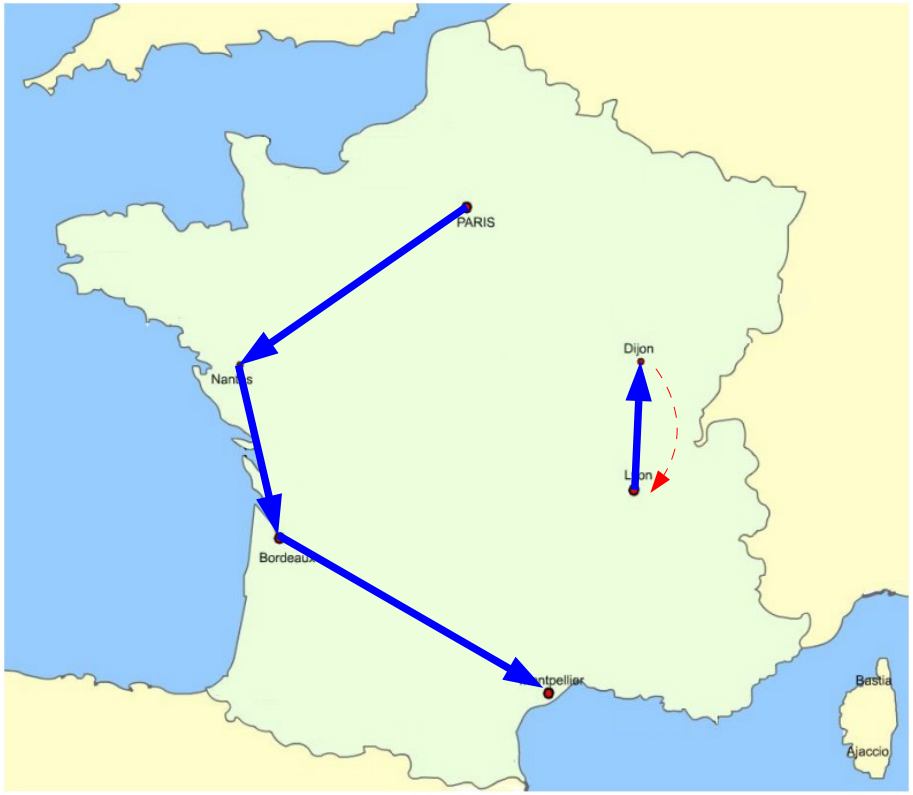
nœud visité

ISEN école d'ingénieurs LILLE

L'algorithme de Little
En images

Arbre de recherche





nœud à visiter

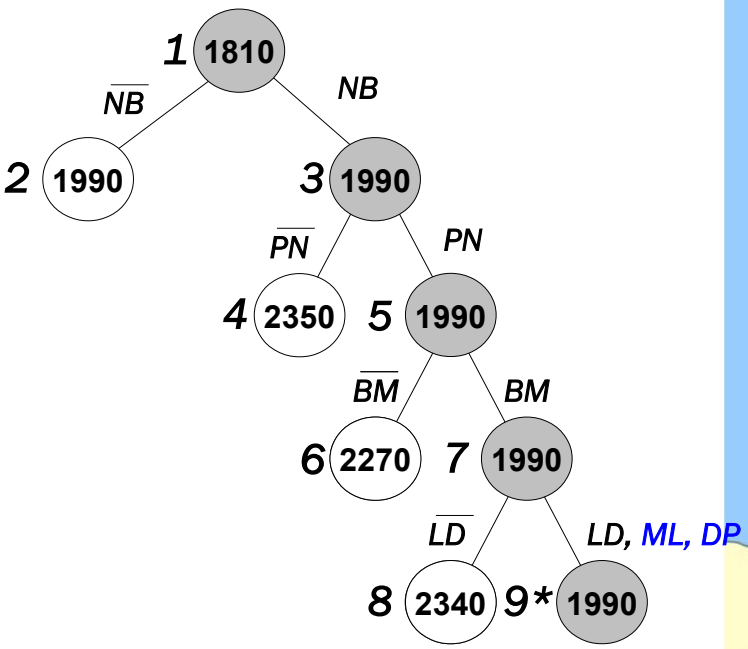
nœud en cours de visite

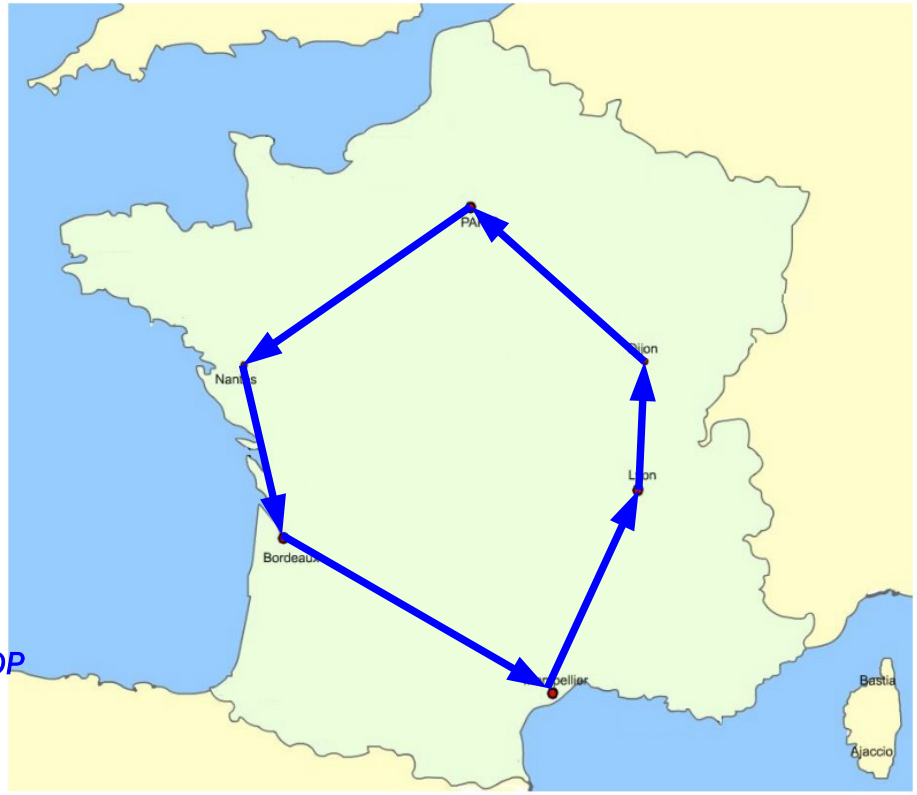
nœud visité

ISEN | école d'ingénieurs
LILLE

L'algorithme de Little
En images

Arbre de recherche





nœud à visiter

nœud en cours de visite

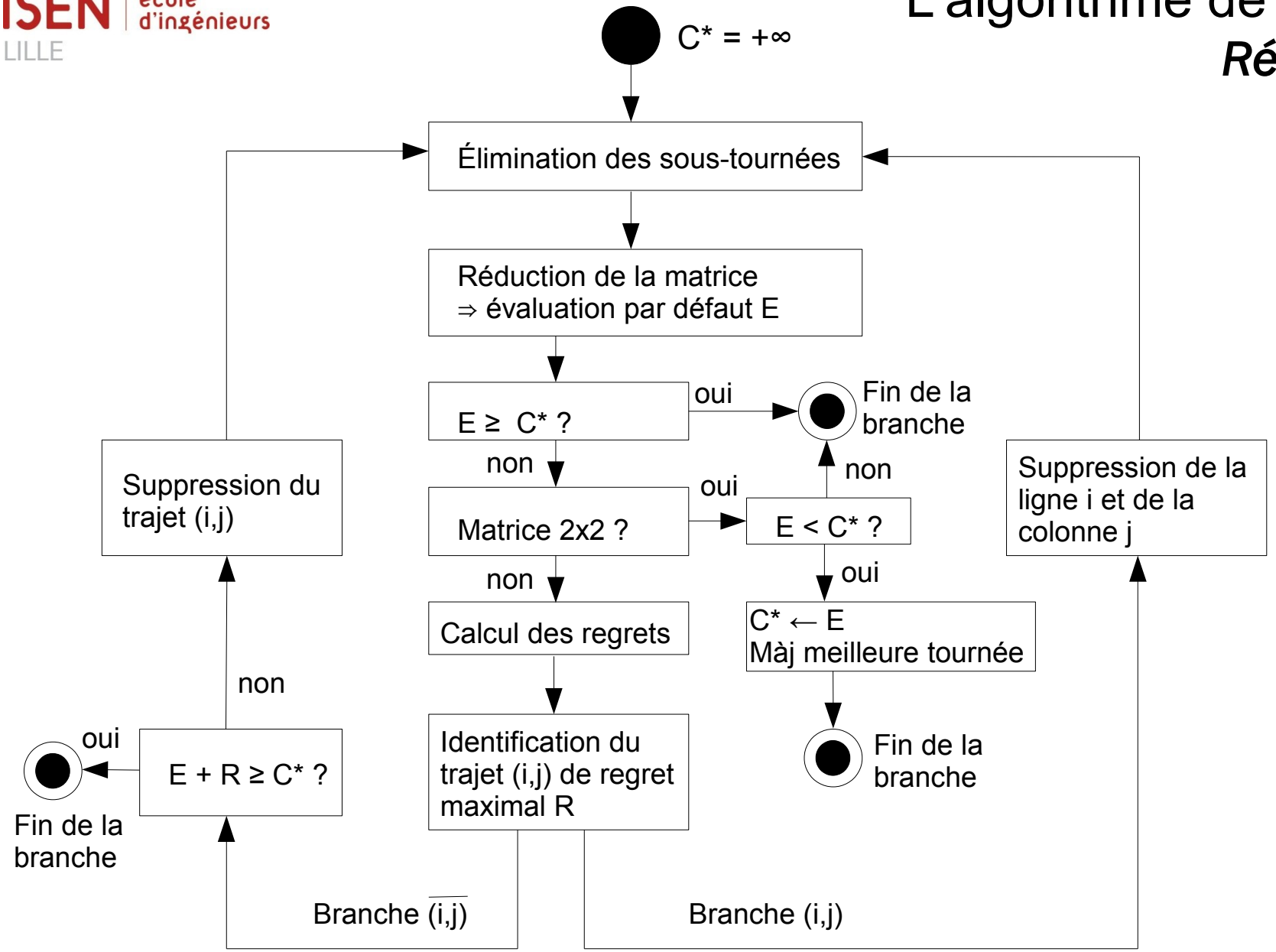
nœud visité

ISEN | école d'ingénieurs
LILLE

Résumé

ISEN école d'ingénieurs LILLE

L'algorithme de Little
Résumé



ISEN | école d'ingénieurs
LILLE

Exercice d'application

Application

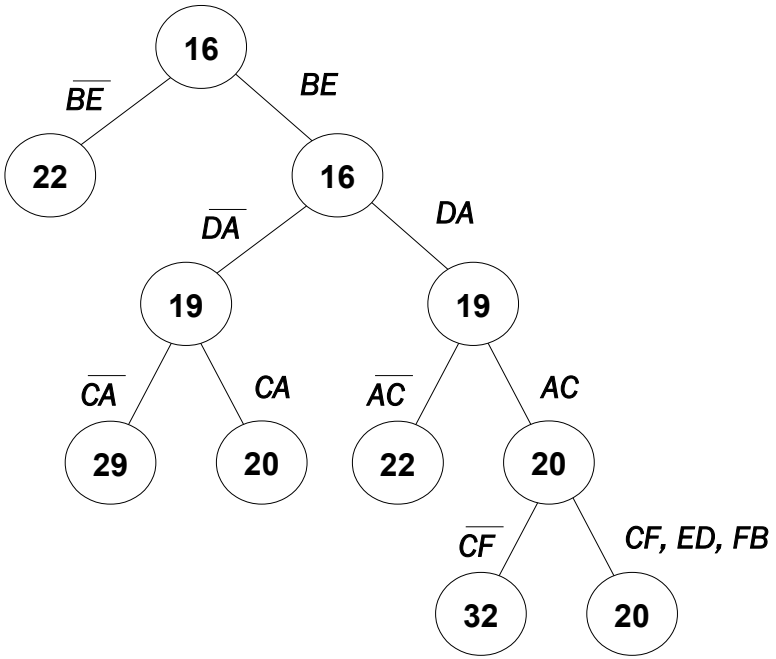
Appliquer l'algorithme de Little pour résoudre problème de voyageur de commerce suivant:

- *6 villes: A,B, C, D, E, F*
- *matrice de coûts:*

	A	B	C	D	E	F
A		1	7	3	14	2
B	3		6	9	1	24
C	6	14		3	7	3
D	2	3	5		9	11
E	15	7	11	2		4
F	20	5	13	4	18	

Solution partielle

Arbre de recherche



Tournée optimale au départ de A:

AC-CF-FB-BE-ED-DA (coût: 20)

ISEN | école d'ingénieurs
LILLE

Exercice d'approfondissement

ISEN école d'ingénieurs
LILLE

Étude de complexité

Le tableau ci-dessous donne le nombre maximal de nœuds $N(n)$ visités par l'algorithme de Little pour $n \geq 3$ villes :

n	$N(n)$
3	8
4	27
5	112
6	565
7	3396
8	23779
9	190240
10	1712169
11	17121700
12	188338711
13	2260064544
14	29380839085
15	411331747204
16	6169976208075
17	98719619329216
18	1678233528596690
19	30208203514740400
20	5739558667800680000

Démontrer par récurrence l'expression de $N(n)$ et vérifier votre résultat sur l'intervalle $[3,20]$ à l'aide du tableau fourni.

Étude de complexité

Réponse :

$N(3) = 8$

Pour $n>3$ **:** $N(n) = (n-1) \cdot (1 + N(n-1))$