



SORBONNE UNIVERSITÉ  
FACULTÉ DES SCIENCES ET INGÉNIERIE

MASTER D'INFORMATIQUE

4I201 - RÉOLUTION DE PROBLÈMES

**Rapport du projet :**  
**Métaheuristiques pour la résolution du**  
**problème de l'arbre de Steiner de poids**  
**minimum**

*B. Thanh Luong, 3504859*  
*Hans Thirunavukarasu, 3605592*

Encadrant :  
M. Thibaut Lust

mai 2017

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Algorithme génétique</b>	<b>2</b>
2.1	Population initiale . . . . .	2
2.2	Heuristique du plus court chemin . . . . .	3
2.3	Heuristique de l'arbre couvrant minimum . . . . .	3
2.4	Diversification la population initiale par randomiser des heuristiques de construction . . . . .	3
<b>3</b>	<b>Recherche locale</b>	<b>3</b>
<b>4</b>	<b>Analyse des résultats</b>	<b>6</b>
<b>5</b>	<b>Conclusion</b>	<b>8</b>

# 1 Introduction

Le problème de l'arbre de Steiner de poids minimum est NP-difficile. Étant donné un graphe connexe  $G = (V, E)$  et un ensemble  $T$  des sommets dits terminaux, nous allons chercher un ensemble des sommets non terminaux qui relie des sommets terminaux créant un arbre couvrant de poids minimum.

L'objectif de ce projet est d'utiliser 3 approches qui résolvent le problème en temps polynomial : un algorithme génétique, une heuristique de construction et une recherche locale.

L'algorithme génétique a été conçu de manière générale et aléatoire afin d'avoir une population variée des individus différents. Grâce à la variation de la population et aux différentes méthodes de génération d'une nouvelle population à chaque étape, la convergence vers un optimum local est rapide.

Nous allons utiliser deux heuristiques qui sont le plus court chemin et l'arbre couvrant minimum pour améliorer l'approche de problème.

Enfin, la recherche locale sera utilisée pour avoir une perspective globale de problème. Nous testerons nos algorithmes sur des instances des ensembles B, C, D, et E disponibles à <http://steinlib.zib.de/testset.php>.

## 2 Algorithme génétique

### 2.1 Population initiale

N'importe quel individu (réalisable ou non réalisable) est codé par un vecteur binaire pour chaque sommet non-terminal, prenant 1 si le sommet est présent dans le graphe (ou la forêt dans le cas non réalisable) et 0 sinon.

La première phase consiste à construire une population aléatoire dont les individus ayant chaque bit une probabilité entre 0.2 et 0.5 d'être pris.

Ensuite, nous allons construire des générations suivantes à partir de la population initiale. Nous proposons 2 types de sélection des parents de la population courante ainsi que 2 stratégies de remplacement de population.

**Sélection des parents :** Parmi des individus, le moyen simple est de choisir celui qui a le meilleur score. Nous proposons une deuxième façon c'est pour chaque individu, nous le prenons comme parent si un tirage aléatoire est inférieur au ratio  $\frac{\text{best score}}{\text{son score}}$ .

**Production des enfants :** S'il y a un unique parent choisi, pour chaque bit, une probabilité entre 0.01 et 0.04 de le changer de 0 à 1 et inversement. Sinon nous faisons un opérateur de croisement, pour chaque pair de parents, nous utilisons le croisement au milieu pour reproduire deux enfants.

**Remplacement de population :** Le remplacement générationnel est d'écraser complètement la population courante par la nouvelle. Le remplacement élitiste est la sélection des meilleurs individus parmi les parents et les enfants avec une probabilité comme la sélection des parents.

Après avoir testé toutes les combinaisons des processus possibles, nous constatons que la convergence vers le minimum des croisements est très lente et aussi loin de l'optimum du problème. Nous allons rajouter dans la population initiale 2 individus spéciales qui sont les résultats des heuristiques du plus court chemin et de l'arbre couvrant minimum. Ces 2 individus sont pour but d'accélérer la convergence.

## 2.2 Heuristique du plus court chemin

Cette heuristique donne une des meilleurs approches dans la population initiale. Elle consiste à la construction un graphe complet des sommets terminaux. Si deux sommets terminaux ne sont pas connecter le coût de l'arête connectant deux sommets est égale au plus court chemin entre ces deux dans le graphe initiale. Une fois le graphe complet est fait, nous remplaçons toutes les arêtes "virtuelles" par le chemin réel, puis construire un arbre couvrant minimum tous les sommets du graphe et enfin enlever toutes les feuilles redondantes (i.e. les sommets non-terminaux de degré 1).

Cette heuristique est  $2 - \frac{2}{|T|} - \text{OPT}$ .

## 2.3 Heuristique de l'arbre couvrant minimum

Cette heuristique est moins efficace que celle du plus court chemin mais donne aussi une solution assez proche de l'optimum. Nous construisons l'arbre couvrant minimum de problème, puis la récursions est faite jusqu'à ce que l'on ne puisse plus enlever les feuilles redondantes.

## 2.4 Diversification la population initiale par randomiser des heuristiques de construction

La population aléatoire n'est pas efficace pour converger vers l'optimum. On converge très souvent vers celui du plus court chemin.

De ce fait, les individus aléatoires seront remplacés par ceux des deux heuristiques proposées en perturbant le poids des arêtes. C'est-à-dire, une nouvelle instance de l'arbre sera crée en changeant le poids associé aux arêtes de -0.2 à -0.05 ou de 0.05 à 0.2. Quand nous obtenons le résultat des heuristiques, nous reviendrons vers le problème initial des nouvelles instance.

## 3 Recherche locale

En partant de la population initiale générée par les heuristiques la randomisation des heuristiques des constructions, nous allons faire une recherche locale suivante :

---

**Algorithme 1 : Recherche locale**

---

**Données :**  $I$  : Meilleur individu de la population initiale

**Résultat :** Résultat de la recherche locale

```
1 faire
2   Voisinage  $\leftarrow []$ 
3   pour arête  $A \in \text{non-terminaux}$  faire
4     si  $A$  prise dans  $I$  alors
5       si  $\text{degré}(A) = 1$  alors
6         Nouveau voisin =  $I$  avec  $A$  non prise
7         Voisinage = Voisinage  $\cup \{ \text{Nouveau voisin} \}$ 
8       sinon
9         Nouveau voisin =  $I$  avec  $A$  non prise
10        si Nouveau voisin connexe alors
11          Voisinage = Voisinage  $\cup \{ \text{Nouveau voisin} \}$ 
12        fin
13      fin
14    sinon
15      Nouveau voisin =  $I$  avec  $A$  prise
16      si Nouveau voisin connexe alors
17        Voisinage = Voisinage  $\cup \{ \text{Nouveau voisin} \}$ 
18      fin
19    fin
20  fin
21  si Meilleur candidat du voisinage a un meilleur score que  $I$  alors
22     $I \leftarrow$  Meilleur candidat du voisinage
23    Non convergence
24  sinon
25    retourner  $I$ 
26  fin
27 tant que Non convergence;
```

---



## 4 Analyse des résultats

instance-OPT	Algorithmes	Résultat	Variance	Écart-type	Temps d'exécution
B/b01.stp -	PCC	82			0.00437
	CM	89			0.00092
	BP-E	82.0	0.0	0.0	0.69497
	BP-G	82.0	0.0	0.0	2.0908
	MP-E	82.0	0.0	0.0	0.93037
	MP-G	82.0	0.0	0.0	0.53656
	RAN	82			1.08031
	LOC	82			0.2593
B/b02.stp -	PCC	90			0.00579
	CM	96			0.00099
	BP-E	88.8	0.36	0.6	1.68771
	BP-G	87.9	0.49	0.7	2.42792
	MP-E	87.6	2.84	1.68523	1.38892
	MP-G	88.6	0.84	0.91652	0.66034
	RAN	83			1.153
	LOC	83			0.44437
B/b03.stp -	PCC	140			0.02426
	CM	145			0.00079
	BP-E	139.0	0.0	0.0	0.17844
	BP-G	139.0	0.0	0.0	0.15985
	MP-E	138.0	0.0	0.0	1.09945
	MP-G	138.0	0.0	0.0	1.70942
	RAN	138			2.55592
	LOC	138			0.44391
B/b04.stp -	PCC	64			0.00516
	CM	68			0.00114
	BP-E	62.0	0.0	0.0	0.29161
	BP-G	61.1	1.89	1.37477	2.23011
	MP-E	59.9	1.29	1.13578	2.3175
	MP-G	60.0	1.8	1.34164	0.90594
	RAN	62			1.1598
	LOC	59			0.44534
B/b05.stp -	PCC	62			0.00801
	CM	72			0.00117
	BP-E	61.0	0.0	0.0	0.32952
	BP-G	61.9	0.69	0.83066	2.48462
	MP-E	61.0	0.0	0.0	1.76089
	MP-G	61.0	0.0	0.0	0.99294
	RAN	61			1.99881
	LOC	61			0.56503
B/b06.stp -	PCC	128			0.03404
	CM	130			0.00169
	BP-E	127.0	0.0	0.0	0.61864
	BP-G	126.1	1.89	1.37477	2.99528
	MP-E	123.6	2.04	1.42829	3.8873
	MP-G	123.7	2.61	1.61555	1.72668
	RAN	124			2.71111
	LOC	124			0.72595

instance-OPT	Algorithme	Résultat	Variance	Écart-type	Temps d'exécution
B/b07.stp -	PCC	111			0.00907
	CM	112			0.0015
	BP-E	111.0	0.0	0.0	0.23455
	BP-G	111.3	0.21	0.45826	8.30198
	MP-E	111.0	0.0	0.0	2.11361
	MP-G	111.0	0.0	0.0	1.20931
	RAN	111			2.27941
	LOC	111			0.78642
B/b08.stp -	PCC	104			0.01734
	CM	107			0.00136
	BP-E	104.0	0.0	0.0	0.25968
	BP-G	104.3	0.81	0.9	0.92731
	MP-E	104.0	0.0	0.0	2.45666
	MP-G	104.0	0.0	0.0	1.22263
	RAN	104			1.75624
	LOC	104			0.70403
B/b09.stp -	PCC	224			0.07396
	CM	226			0.0011
	BP-E	221.0	0.0	0.0	0.48518
	BP-G	220.4	0.24	0.4899	9.89917
	MP-E	220.3	0.41	0.64031	6.02025
	MP-G	220.4	0.44	0.66332	2.57152
	RAN	220			5.29973
	LOC	220			1.30634
B/b10.stp -	PCC	98			0.0132
	CM	114			0.00165
	BP-E	93.7	0.81	0.9	4.48824
	BP-G	93.6	16.84	4.10366	40.86327
	MP-E	90.5	3.45	1.85742	5.24673
	MP-G	91.2	2.76	1.66132	3.08325
	RAN	88			4.73751
	LOC	86			4.66053
B/b11.stp -	PCC	92			0.02981
	CM	121			0.00174
	BP-E	92.0	0.0	0.0	0.99987
	BP-G	94.2	19.56	4.42267	74.77886
	MP-E	90.6	0.84	0.91652	5.85837
	MP-G	90.8	0.96	0.9798	2.3999
	RAN	90			5.26706
	LOC	90			1.92905
B/b12.stp -	PCC	174			0.09599
	CM	183			0.00135
	BP-E	174.0	0.0	0.0	1.10555
	BP-G	174.0	0.0	0.0	0.7528
	MP-E	174.0	0.0	0.0	5.17198
	MP-G	174.0	0.0	0.0	3.30554
	RAN	174			8.91171
	LOC	174			2.52435



instance-OPT	Algorithme	Résultat	Variance	Écart-type	Temps d'exécution
B/b13.stp -	PCC	175			0.02191
	CM	189			0.0022
	BP-E	174.5	1.05	1.0247	6.5285
	BP-G	174.3	2.41	1.55242	8.99241
	MP-E	173.2	1.16	1.07703	4.36672
	MP-G	173.0	1.4	1.18322	2.93404
	RAN	174			6.20703
	LOC	171			4.43862
B/b14.stp -	PCC	238			0.0446
	CM	255			0.0021
	BP-E	237.0	0.0	0.0	0.4777
	BP-G	244.0	12.8	3.57771	59.70153
	MP-E	236.3	0.21	0.45826	5.00365
	MP-G	236.1	0.29	0.53852	3.02972
	RAN	236			5.9153
	LOC	236			3.38697
B/b15.stp -	PCC	322			0.16348
	CM	333			0.00163
	BP-E	318.9	0.09	0.3	7.5786
	BP-G	319.1	2.89	1.7	11.94463
	MP-E	318.6	0.24	0.4899	6.57827
	MP-G	318.8	0.16	0.4	3.48763
	RAN	318			8.49575
	LOC	318			4.17085
B/b16.stp -	PCC	137			0.02842
	CM	177			0.00218
	BP-E	135.5	1.45	1.20416	76.48287
	BP-G	138.6	83.84	9.15642	36.63659
	MP-E	135.4	0.84	0.91652	7.6889
	MP-G	135.5	1.45	1.20416	4.14082
	RAN	132			11.28039
	LOC	134			4.54774
B/b17.stp -	PCC	134			0.06228
	CM	137			0.00214
	BP-E	134.0	0.0	0.0	1.02891
	BP-G	133.6	0.64	0.8	572.66436
	MP-E	133.2	0.56	0.74833	9.58948
	MP-G	132.5	0.65	0.80623	5.89732
	RAN	133			21.46101
	LOC	131			8.94396
B/b18.stp -	PCC	223			0.24142
	CM	229			0.00183
	BP-E	221.9	0.09	0.3	18.83366
	BP-G	221.8	0.36	0.6	55.83069
	MP-E	220.5	0.45	0.67082	20.86467
	MP-G	220.4	0.64	0.8	9.32316
	RAN	219			18.15929
	LOC	218			14.64771

## 5 Conclusion