

Projet Optimisation Combinatoire

**Hamza BOUKHRISS
Achraf ALLOUL
Ibrahim LABIOD
Oussama NEZAR**

29 Novembre 2020

Description du problème et des instances :

On souhaite élaborer une méthode d'optimisation dédiée à un problème de couverture de cible (Set Covering Problem) .

On Dispose :

- d'un ensemble de M cibles à couvrir
- d'un ensemble de N capteur qui s'active avec un coût C_i

Un capteur peut capter un ensemble de cible , une cible peut être couverte par un à plusieurs capteurs.

Notre objectif sera de couvrir toutes les cibles tout en réduisant le coût d'activation .

Modélisation Linéaire :

Dans cette problématique , on dispose d'un ensemble de M cibles à couvrir , et de N capteurs à activer lier à un coût C , avec le capteur i le cou est C_i . On a une liste de cibles à couvrir qu'on nomme A , avec la cible j , A_j est égale à 0 si elle n'est pas active , 1 si elle est activée . On crée un liste des capteurs couvrant la cible j qu'on nomme V .

Le Modèle Linéaire que nous avons établi est :

$$\text{Minimiser } Z = \sum_i A_i \times C_i$$

Les contraintes :

→ $V_{i,j} = 1$ Si la cible est couverte par la capteur , sinon $V_{i,j} = 0$

→ $A_i = 1$ Si le capteur i est activé

→ C_i le coût d'activation du capteur i

→ N le nombre de capteur activables

→ M le Nombre de cibles

→ $\forall j , \sum_i V_{j,i} \times A_i > 0$

→ $\forall i,j, M > 0, N > 0, C_i \geq 0, V_{i,j} \in \{0, 1\} , A_i \in \{0, 1\}$

Solveur Linéaire GLPK :

On a implémenté une fonctionnalité qui permet de traduire l'instance de base donnée par le prof , et de générer un fichier .lp pour but de pouvoir déterminer la solution optimale pour chaque instance via le solveur linéaire GLPK .

On obtient la traduction des instantes sous format .lp :

```
Minimize
z: 1 x1 + 1 x2 + 1 x3 + 1 x4 + 1 x5 + 1 x6 + 1 x7 + 1 x8 + 1 x9 + 1 x10 + 1 x11 + 1 x12 + 2 x13 + 2 x14 + 2 x15 + 2 x16 + 2 x17 + 2 x18 + 2 x19 + 2 x20 + 2 x21 + 2 x22
Subject To
c1: x91 + x214 + x230 + x289 + x351 + x416 + x488 + x491 + x518 + x567 + x720 + x721 + x735 + x753 + x768 + x928 + x990 >= 1
c2: x22 + x47 + x99 + x192 + x299 + x322 + x340 + x500 + x619 + x628 + x640 + x663 + x709 + x736 + x796 + x844 + x930 + x970 >= 1
c3: x2 + x37 + x56 + x115 + x151 + x160 + x178 + x182 + x218 + x228 + x345 + x388 + x587 + x588 + x607 + x614 + x629 + x635 + x636 + x673 + x757 + x793 + x800 + x849 +
c4: x2 + x93 + x125 + x328 + x334 + x337 + x391 + x402 + x559 + x697 + x703 + x733 + x781 + x790 + x909 >= 1
c5: x18 + x81 + x85 + x116 + x165 + x261 + x277 + x280 + x290 + x331 + x338 + x353 + x400 + x506 + x540 + x557 + x590 + x653 + x712 + x715 + x802 + x884 + x978 >= 1
c6: x14 + x51 + x78 + x200 + x236 + x317 + x332 + x469 + x495 + x578 + x701 + x712 + x790 + x822 + x830 + x839 + x988 >= 1
c7: x78 + x129 + x435 + x459 + x491 + x512 + x548 + x551 + x576 + x640 + x643 + x648 + x652 + x813 + x916 + x948 + x954 + x983 + x999 >= 1
c8: x120 + x150 + x276 + x278 + x379 + x389 + x437 + x493 + x507 + x537 + x568 + x608 + x621 + x710 + x770 + x908 + x935 >= 1
c9: x53 + x59 + x136 + x139 + x178 + x214 + x258 + x288 + x307 + x328 + x476 + x632 + x655 + x735 + x758 + x793 + x880 + x886 + x949 >= 1
c10: x3 + x6 + x64 + x101 + x104 + x144 + x180 + x276 + x402 + x422 + x538 + x600 + x635 + x638 + x694 + x731 + x889 + x942 >= 1
c11: x66 + x118 + x255 + x278 + x434 + x463 + x480 + x555 + x576 + x634 + x669 + x674 + x688 + x751 + x883 + x964 >= 1
c12: x3 + x24 + x29 + x45 + x60 + x74 + x117 + x136 + x162 + x232 + x318 + x371 + x400 + x421 + x433 + x460 + x612 + x659 + x660 + x779 + x800 + x835 >= 1
c13: x14 + x136 + x385 + x389 + x426 + x546 + x702 + x778 + x789 + x831 + x970 >= 1
c14: x68 + x99 + x205 + x275 + x364 + x423 + x543 + x559 + x569 + x609 + x638 + x642 + x693 + x836 + x858 >= 1
c15: x49 + x127 + x149 + x181 + x185 + x228 + x332 + x375 + x426 + x436 + x437 + x486 + x514 + x570 + x594 + x601 + x808 + x814 + x818 + x827 + x874 + x903 + x960 + x97
c16: x86 + x242 + x354 + x365 + x405 + x418 + x428 + x449 + x547 + x565 + x598 + x652 + x682 + x684 + x780 + x829 + x852 + x865 >= 1
c17: x6 + x17 + x33 + x112 + x172 + x180 + x202 + x225 + x297 + x298 + x317 + x390 + x451 + x526 + x528 + x547 + x595 + x654 + x680 + x709 + x743 + x801 + x804 + x891 +
c18: x1 + x80 + x98 + x125 + x146 + x162 + x328 + x394 + x414 + x445 + x460 + x501 + x589 + x622 + x681 + x691 + x807 + x841 + x845 >= 1
c19: x15 + x97 + x112 + x175 + x253 + x324 + x421 + x454 + x605 + x715 + x746 + x785 + x843 + x989 >= 1
c20: x36 + x125 + x144 + x185 + x195 + x219 + x273 + x303 + x371 + x399 + x400 + x583 + x622 + x625 + x681 + x726 + x969 >= 1
c21: x44 + x109 + x125 + x149 + x187 + x456 + x580 + x673 + x678 + x804 + x897 >= 1
c22: x57 + x80 + x116 + x158 + x172 + x267 + x287 + x294 + x313 + x338 + x390 + x401 + x455 + x476 + x492 + x560 + x629 + x682 + x753 >= 1
c23: x16 + x19 + x63 + x67 + x82 + x85 + x305 + x371 + x429 + x439 + x452 + x560 + x620 + x630 + x631 + x647 + x681 + x752 + x847 + x931 >= 1
c24: x106 + x110 + x129 + x141 + x180 + x198 + x213 + x227 + x244 + x370 + x396 + x397 + x410 + x472 + x487 + x500 + x598 + x629 + x636 + x641 + x690 + x725 + x777 + x8
c25: x73 + x118 + x122 + x148 + x159 + x164 + x183 + x209 + x322 + x365 + x377 + x470 + x471 + x517 + x660 + x667 + x707 + x747 + x855 + x880 + x883 + x931 + x936 + x93
```

On utilise ensuite le solveur linéaire GLPK (sous linux) , et on obtient le résultat suivant:

```
amza@lllllllll:~/Bureau$ glpsol --lp OUT.txt --output solution.txt
GLPSOL: GLPK LP/MIP Solver, v4.65
Parameter(s) specified in the command line:
--lp OUT.txt --output solution.txt
Reading problem data from 'OUT.txt'...
200 rows, 1000 columns, 4009 non-zeros
200 integer variables, all of which are binary
405 lines were read
GLPK Integer Optimizer, v4.65
200 rows, 1000 columns, 4009 non-zeros
200 integer variables, all of which are binary
Preprocessing...
200 rows, 1000 columns, 4009 non-zeros
200 integer variables, all of which are binary
Scaling...
A: min|aij| = 1.000e+00 max|aij| = 1.000e+00 ratio = 1.000e+00
Problem data seem to be well scaled
Constructing initial basis...
Size of triangular part is 200
Solving LP relaxation...
GLPK Simplex Optimizer, v4.65
200 rows, 1000 columns, 4009 non-zeros
0: obj = 0.000000000e+00 inf = 2.000e+02 (200)
46: obj = 1.639000000e+03 inf = 0.000e+00 (0)
* 369: obj = 4.290000000e+02 inf = 0.000e+00 (0) 3
OPTIMAL LP SOLUTION FOUND
Integer optimization begins...
Long-step dual simplex will be used
+ 369: mip = not found yet >= -inf (1; 0)
+ 369: >>>> 4.290000000e+02 >= 4.290000000e+02 0.0% (1; 0)
+ 369: mip = 4.290000000e+02 >= tree is empty 0.0% (0; 1)
INTEGER OPTIMAL SOLUTION FOUND
Time used: 0.1 secs
Memory used: 1.4 Mb (1485694 bytes)
Writing MIP solution to 'solution.txt'...
```

```
amza@lllllllll:~/Bureau$ cat solution.txt
Problem:
Rows: 200
Columns: 1000 (200 integer, 200 binary)
Non-zeros: 4009
Status: INTEGER OPTIMAL
Objective: z = 429 (MINIMUM)
```

	No.	Row name	Activity	Lower bound	Upper bound
	1	c1	1	1	
	2	c2	2	1	
	3	c3	1	1	
	4	c4	1	1	
	5	c5	4	1	
	6	c6	2	1	
	7	c7	2	1	
	8	c8	1	1	
	9	c9	1	1	
	10	c10	3	1	
	11	c11	1	1	
	12	c12	3	1	
	13	c13	1	1	
	14	c14	1	1	
	15	c15	1	1	
	16	c16	1	1	
	17	c17	2	1	
	18	c18	3	1	
	19	c19	1	1	
	20	c20	1	1	
	21	c21	1	1	
	22	c22	1	1	
	23	c23	3	1	
	24	c24	1	1	
	25	c25	1	1	
	26	c26	2	1	
	27	c27	2	1	
	28	c28	1	1	
	29	c29	1	1	
	30	c30	1	1	

Algorithme Glouton :

Le premier algorithme implémenté pour trouver la solution la plus optimal sur le problème de couverture de cible est l'algorithme glouton .

Heuristique choisie :

1. On choisit le capteur avec le meilleur ratio de capteur non-couvert divisé par le coût.
2. Une fois le bon capteur trouvé , on l'allume et on déclare toutes les cibles déjà couvertes .
3. On refait la même manipulation jusqu'à ce que toutes les cibles soient couvertes .

Algorithme Glouton Améliorer :

Pour réaliser une nouvelle amélioration à l'algorithme glouton on utilise une matrice, qui représente tous les capteur allumer et chaque capteur, avec une somme de capteur couvrant la même cible.

[illegible]

le capteur couvrant une cibles couvertes par un seule et même capteur , ce capteur est dis essentielle puisque sans sa capteur on ne pourra jamais couvrir toute les cibles , ensuite on fais un test par rapport a chaque capteur si on attache ce capteur toute les cible seront couverte ou pas .

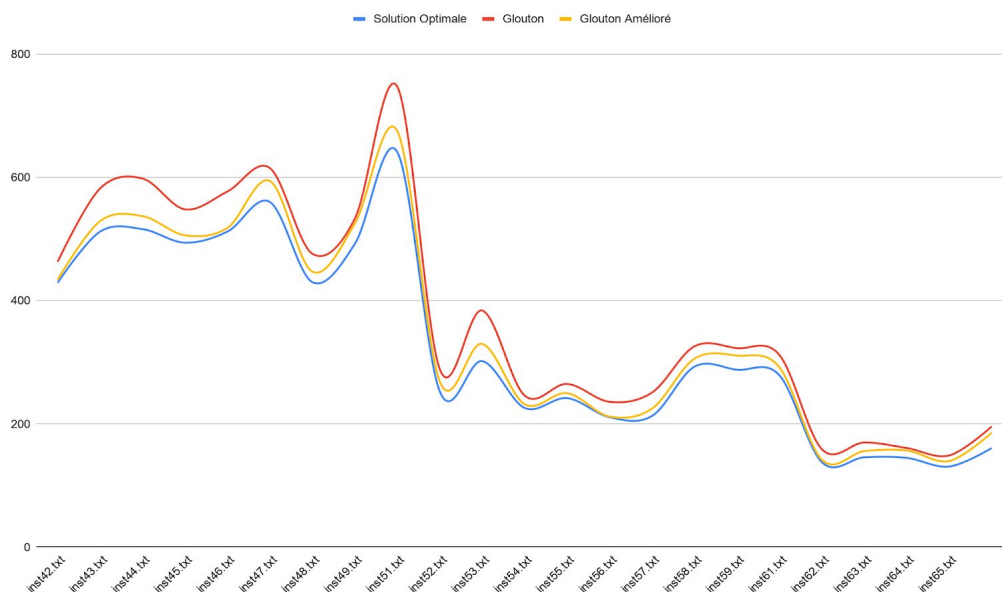
On refais cette manipulation tant qu'on peut plus étendre de capteur .

Analyse des Résultats :

Le tableau ci-dessus représente le score de chaque algorithme implémenté avec une comparaison avec les résultats de GLPK (considérer comme le résultat le plus optimal)

Instance	Solution Optimale	Glouton	Glouton Amélioré	GAP Glouton	GAP Glouton Amélioré
inst41.txt	429	463	434	0,08	0,01
inst42.txt	512	582	529	0,14	0,03
inst43.txt	516	598	537	0,16	0,04
inst44.txt	494	548	506	0,11	0,02
inst45.txt	512	577	518	0,13	0,01
inst46.txt	560	615	594	0,10	0,06
inst47.txt	430	476	447	0,11	0,04
inst48.txt	492	533	525	0,08	0,07
inst49.txt	641	747	675	0,17	0,05
inst51.txt	253	289	269	0,14	0,06
inst52.txt	302	384	330	0,27	0,09
inst53.txt	226	246	232	0,09	0,03
inst54.txt	242	265	250	0,10	0,03
inst55.txt	211	236	212	0,12	0,00
inst56.txt	213	251	225	0,18	0,06
inst57.txt	293	326	306	0,11	0,04
inst58.txt	288	323	311	0,12	0,08
inst59.txt	279	312	292	0,12	0,05
inst61.txt	138	159	142	0,15	0,03
inst62.txt	146	170	156	0,16	0,07
inst63.txt	145	161	157	0,11	0,08
inst64.txt	131	149	140	0,14	0,07
inst65.txt	161	196	186	0,22	0,16

La courbe en dessous montre une comparaison entre les différents algorithmes par rapport à chaque instance proposée par le professeur . On remarque donc que notre solution du glouton améliorée s'approche de plus en plus au résultat optimal GLPK .



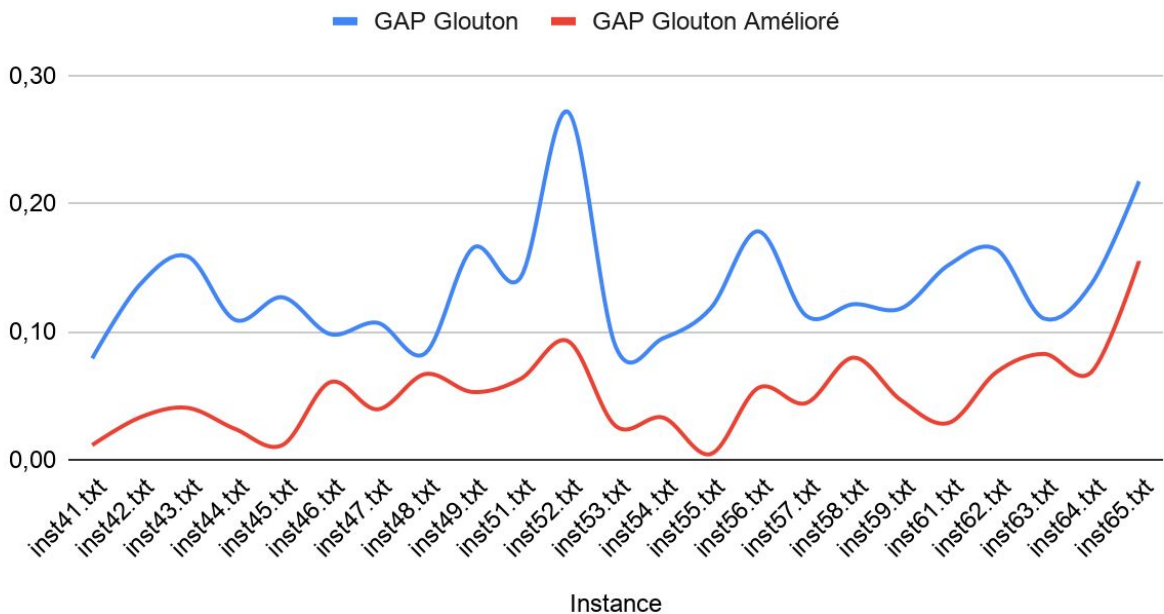
GAP :

le Gap = la différence entre la solution que l'on a trouvé et la solution optimale (GLPK)

$$Gap(S) = \frac{f(S) + f(S^*)}{f(S^*)}$$

avec $f(S)$ la solution optimal et $f(S^)$ la solution trouve*

GAP Glouton et GAP Glouton Amélioré



Le graphique au-dessus représente le GAP de chaque instance en fonction de l'instance , on remarque toujours que notre version améliorée du glouton s'approche encore plus vers la solution optimale (GLPK).

Conclusion :

Le projet était intéressant , mais malheureusement vu notre charge de travail on a pas pu consacrer tous nos moyens sur ce projet , notre groupe se compose majoritairement de 75% d'alternant , il était important de travailler en groupe en même temps pour avancer simultanément sur le projet , Espérons qu'on bien pu vous expliquer notre travailler sur le projet proposer .