

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Électronique et d'Informatique

Département Informatique



Master 1 SII

Projet Intelligence en essaim

Rapport de la partie 2

Résolution du problème SAT

Réalisé par :

- AYAD Abdou 171731057441
- GUENDOUZ Akrem Mohamed 171731070870
- BOUSSEKSOU Yacine 171731078048

19 Juin 2021

Sommaire

1	<i>L'algorithme génétique (GA, genetic algorithm) :.....</i>	<i>3</i>
1.1	Implémentation :	3
1.2	Réglage des paramètres.....	3
1.3	Tests et résultats	4
2	<i>Particle Swarm Optimization (PSO)</i>	<i>5</i>
2.1	Implémentation	5
2.2	Réglage des paramètres.....	6
2.3	Tests et résultats	6
3	<i>Comparaison entre les différentes approches (BFS, DFS, A*, GA, PSO) :.....</i>	<i>7</i>
4	<i>Interface graphique de GA et PSO :</i>	<i>8</i>
4.1	Définition des données.....	8
4.2	Statistiques du déroulement d'un exemple de GA et PSO :	10
5	<i>Conclusion :</i>	<i>11</i>

1 L'algorithme génétique (GA, genetic algorithm) :

est un algorithme de recherche évolutionniste. il est utilisé dans le but de trouver des solutions optimales ou des solutions moins mauvaises pour les problèmes d'optimisations qui n'ont pas de solution précise (solution inconnue). Cette algorithme est basé sur des phénomènes biologiques afin de faire évoluer les populations de solutions de manière progressive(reproduction), et pour cela, l'algorithme génétique repose sur 3 opérations qui sont répétées plusieurs fois tout en gardant la meilleure solution trouvée et ces opérations sont la sélection (choix des individus qui sont plus enclins à obtenir les meilleurs résultats),le croisement ou l'enjambement (division et combinaison des parties de solutions) et la mutation (altération aléatoire des particularités d'un individu).

1.1 Implémentation :

L'exploitation du l'algorithme génétique dans les problèmes SAT s'effectue par la génération d'un ensemble des solutions envisageables initiale avec solutions aléatoires différentes. En bouclant et pour chaque itération, deux individus tirés d'une manière aléatoire sont désignés pour la reproduction. Cette dernière est composée du croisement autrement dit enjambement qui est effectué uniquement si un taux arbitraire est inférieur à une valeur fixe (définie comme paramètre du l'AG). Ce traitement consiste à diviser les solutions des individus précédents en liant la 1ère moitié avec la 2ème moitié de chaque individu. Puis, une mutation des deux résultants est faite, de même que pour le croisement, un taux doit être petit qu'une valeur fixée préalablement. Cette opération est définie par l'inversion d'autant de littéraux que le pourcentage de la mutation (en entrée) par rapport au nombre total des variables. Enfin, il remplace l'ensemble de début par les solutions modifiées si sont meilleures .

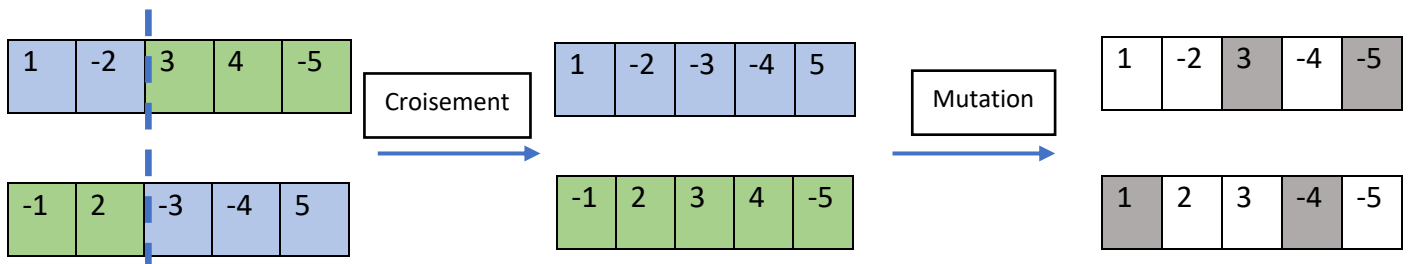


Figure qui montre le processus de reproduction (croisement et mutation) de deux individus

1.2 Réglage des paramètres

L'efficacité de toute méthode d'optimisation ne dépend pas seulement de son algorithme, mais aussi de ses hyperparamètres. Ces derniers, choisis sagement, permettent d'exploiter les possibilités de la technique concernée au maximum.

Pour cela, un ajustement des paramètres du GA est fait dans le but de trouver la meilleure combinaison des valeurs d'entrée. La procédure consiste à varier un seul paramètre et fixer les autres à chaque fois en faisant appel à des boucles imbriquées. ainsi, tous les résultats sont sauvegardés et analysés pour retrouver la meilleure configuration qui satisfait le plus des clauses dans un temps minimal

1.3 Tests et résultats

Comme pour les méthodes exactes, "uf75-325" est utilisé ; cinq instances avec "*nombre d'essais = 5*" et "*temps par essai = 5 secondes*". Pour les hyperparamètres, ils sont ajustés sur les meilleurs valeurs trouvés lors de leur réglage :

- Taille de la population : 40
- Taux de croisement : 100%
- Taux de mutation : 36%
- Nombre des itérations : 3500

Ci-dessous, un tableau qui résume la globalité des tests.

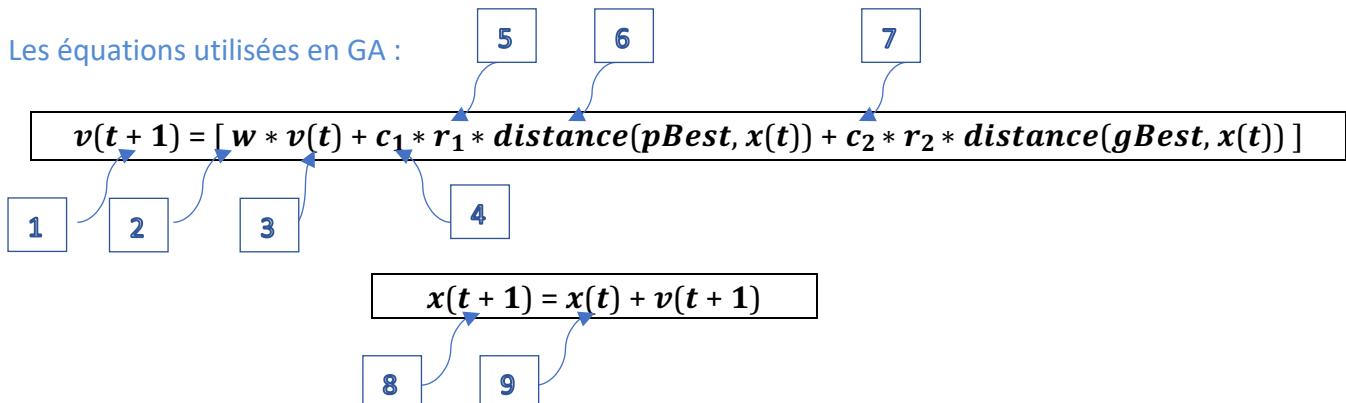
Numéro de l'instance	Nombre d'essais	Temps par essai (sec)	Taux moyen de satisfiabilité	Temps moyen d'exécution en secondes
uf75-01.cnf	5	5	97.969223 %	0.143
uuf75-010.cnf	5	5	98.4615402 %	0.094
uf75-020.cnf	5	5	98.4000015 %	0.11
uuf75-030.cnf	5	5	98.5230713 %	0.125
uf75-060.cnf	5	5	98.1538544 %	0.137
La moyenne	5	5	98.3015381 %	0.121

2 Particle Swarm Optimization (PSO)

L'algorithme de l'optimisation par essaims particulaires (Particle Swarm Optimization, PSO) s'inspire du comportement des individus sociaux dans un groupe en général, et le déplacement d'une clique des oiseaux en particulier. Il débute par la création d'un ensemble de particules aux positions aléatoires en sauvegardant cette dernière comme le meilleur emplacement visité. Par la suite, ces positions du groupe initial sont modifiées en jouant sur la vitesse calculée à l'aide d'une fonction mathématique de chaque particule en actualisant leur emplacement ainsi que la meilleure position globale, le but étant d'explorer l'espace des solutions potentielles. Ces actions sont réitérées un nombre fini de fois.

2.1 Implémentation

L'adaptation du PSO au SAT est établie en créant une liste des individus avec solutions aléatoires en étant les meilleures localement (Particle's Best Position, pBest) pour chacun, et en actualisant le meilleur global (Global Best Position, gBest). Après, une boucle est exécutée autant de fois que la valeur de "maximum itérations" (paramètre en entrée) en changeant/améliorant à chaque fois pBest/gBest, et cela en ajustant la vitesse de chaque particule qui sera utilisée pour modifier la solution courante en inversant autant de bits que la valeur de la nouvelle vitesse.



1 : Nouvelle vitesse

2 : Paramètre

3 : Ancienne vitesse

4 : Paramètre

5 : Paramètre (nombre aléatoire entre 0 et 1)

6 : Distance hamiltonienne

7 : Paramètre

8 : Nouvelle position (nouvelle solution)

9 : Ancienne position (ancienne solution)

2.2 Réglage des paramètres

Comme pour le GA, un ajustement des paramètres est fait dans le but de trouver la meilleure combinaison des hyperparamètres en entrée du PSO.

2.3 Tests et résultats

Similairement, les expérimentations sont faites sur "uf75-325" (5 instances, nombre d'essais = 5 et temps par essai = 5 secondes). Pour les hyperparamètres, ils sont ajustés sur la meilleure combinaison de valeurs trouvés dans la phase du réglage :

- Nombre de particules : 40
- Première constante (c_1) : 60
- Deuxième constante (c_2) : 30
- w : 40
- Nombre des itérations : 2500

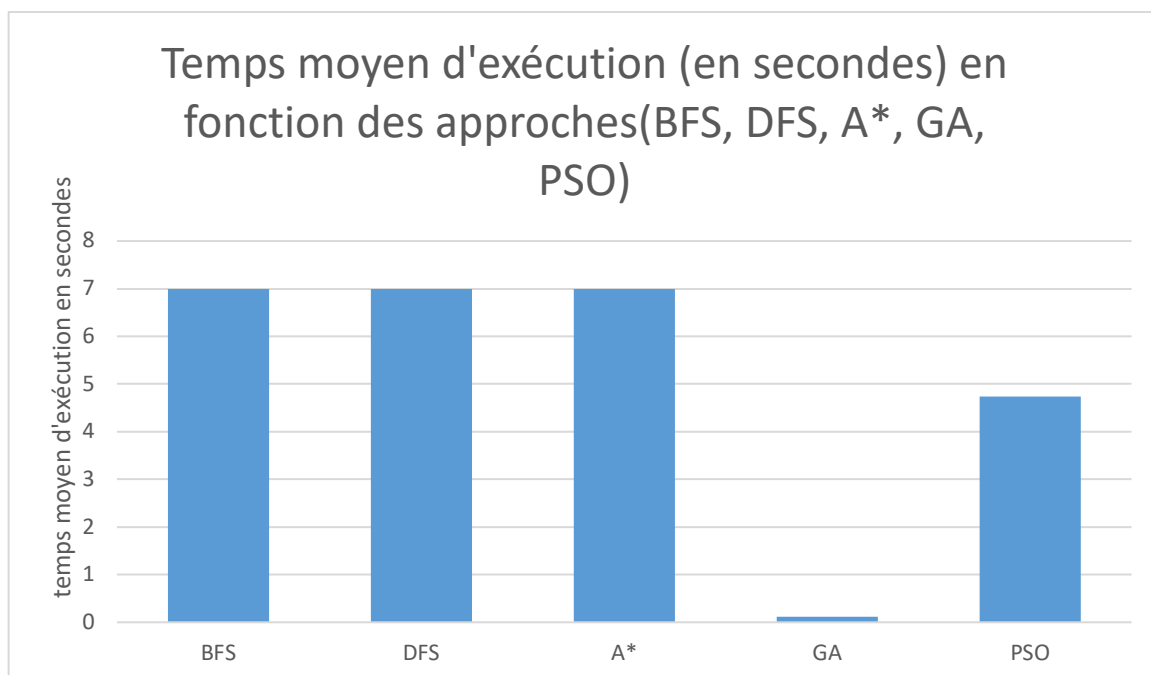
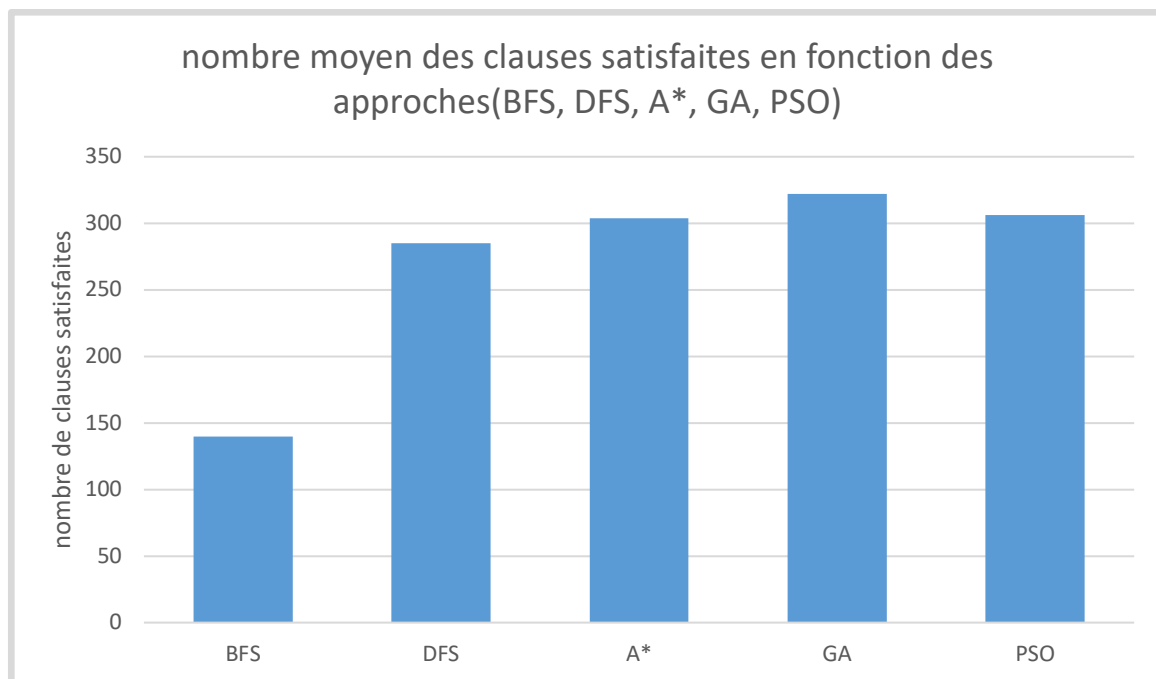
Le tableau suivant résume les tests et leurs résultats.

Numéro de l'instance	Nombre d'essais	Temps par essai (sec)	Taux moyen de satisfiabilité	Temps moyen d'exécution en secondes
uf75-01.cnf	5	5	94.2769241 %	4.621
uuf75-010.cnf	5	5	94.030777 %	4.595
uf75-020.cnf	5	5	93.6615372 %	4.603
uuf75-030.cnf	5	5	93.5384598 %	4.868
uf75-060.cnf	5	5	94.030777 %	5
La moyenne	5	5	93.907695 %	4.7374

3 Comparaison entre les différentes approches (BFS, DFS, A*, GA, PSO) :

La comparaison est faite selon deux critères, "nombre moyen de clauses satisfaites" et "temps moyen d'exécution".

Les graphes statistiques associés ci-dessous.



En analysant les deux histogrammes précédents on constate que le DFS est plus performant que le BFS en terme du nombre de clauses satisfaites, avec le même temps d'exécution. Notons aussi une légère évolution de satisfaction des clauses de A* contre la meilleure recherche aveugle. Ce résultat logique a été attendu vu l'utilisation d'une simple fonction heuristique.

Pour les approches méta-heuristiques, le GA est largement plus performant que le PSO. En effet, une augmentation des clauses satisfaites est remarquée pour GA par rapport à PSO. En ce qui concerne l'autre critère, le temps, est bien-évidemment en faveur de GA, le résultat est obtenu en un temps de 0,121 secondes en moyenne (écart de 4,61 secondes avec PSO).

Enfin, la comparaison des recherches exactes avec celles méta-heuristiques est claire, ces dernières sont tout simplement meilleures que les autres, que ce soit en "nombre moyen de clauses satisfaites" ou bien "temps moyen d'exécution" (écart de 4.883 secondes).

4 Interface graphique de GA et PSO :

4.1 Définition des données

Comme le montre l'interface ci-dessous, les mêmes composantes graphiques que celles des méthodes heuristiques sont maintenues, à la différence des champs de saisi issus pour l'ajustement des hypres paramètres de la configuration d'entrée pour le GA.

Solveur SAT

Configuration des données d'entrée | Résultats | Statistiques

Importer une instance

Algorithme Génétique (GA)

Nombre de tentati... 5

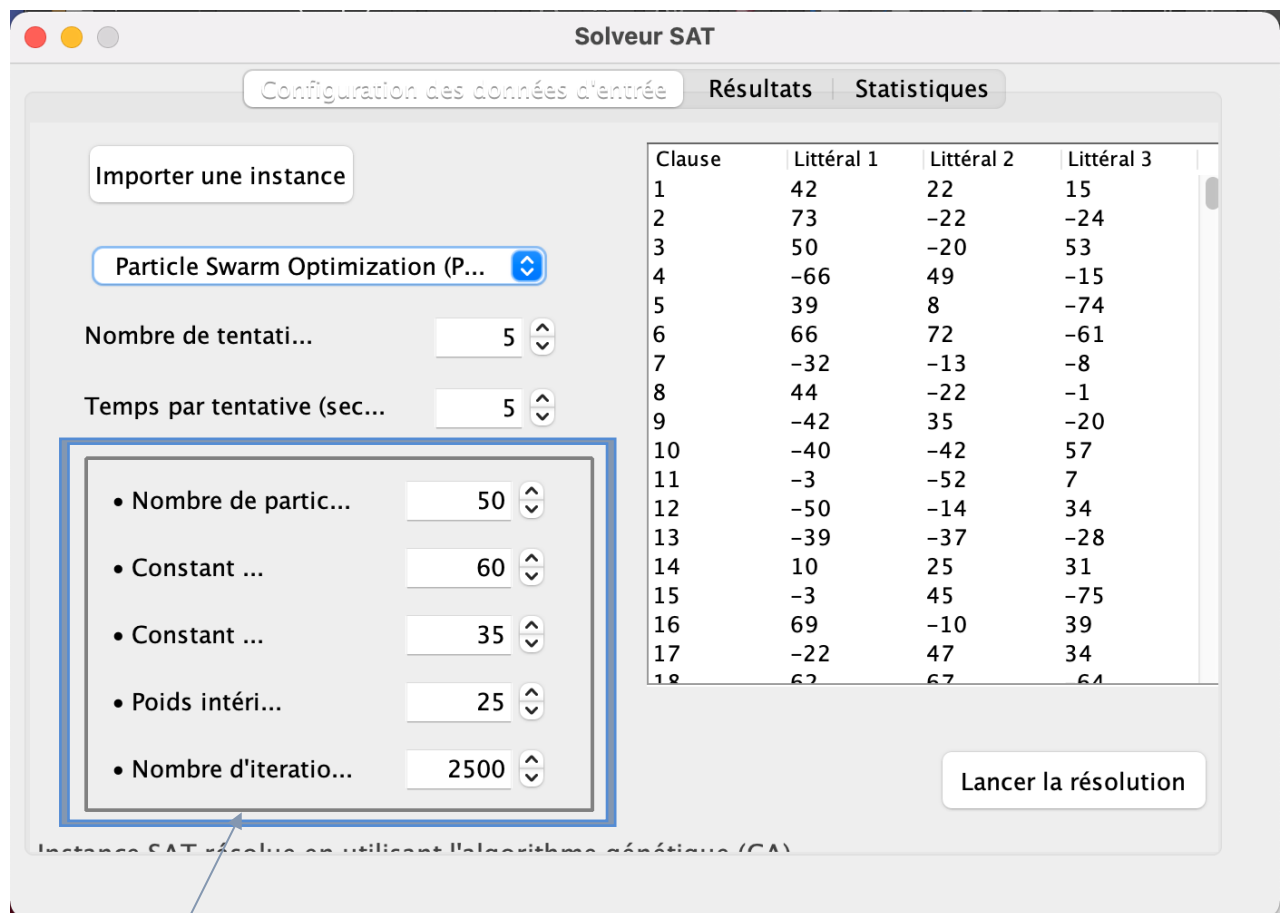
Temps par tentative (sec...) 5

• Taille de la popu... 100
 • Taux de crois... (%) 70
 • Taux de mu... (%) 20
 • Nombre d'iteratio... 3000

Clause	Littéral 1	Littéral 2	Littéral 3
1	42	22	15
2	73	-22	-24
3	50	-20	53
4	-66	49	-15
5	39	8	-74
6	66	72	-61
7	-32	-13	-8
8	44	-22	-1
9	-42	35	-20
10	-40	-42	57
11	-3	-52	7
12	-50	-14	34
13	-39	-37	-28
14	10	25	31
15	-3	45	-75
16	69	-10	39
17	-22	47	34
18	62	67	64

Lancer la résolution

Instance SAT résolue en utilisant l'algorithme génétique (GA)

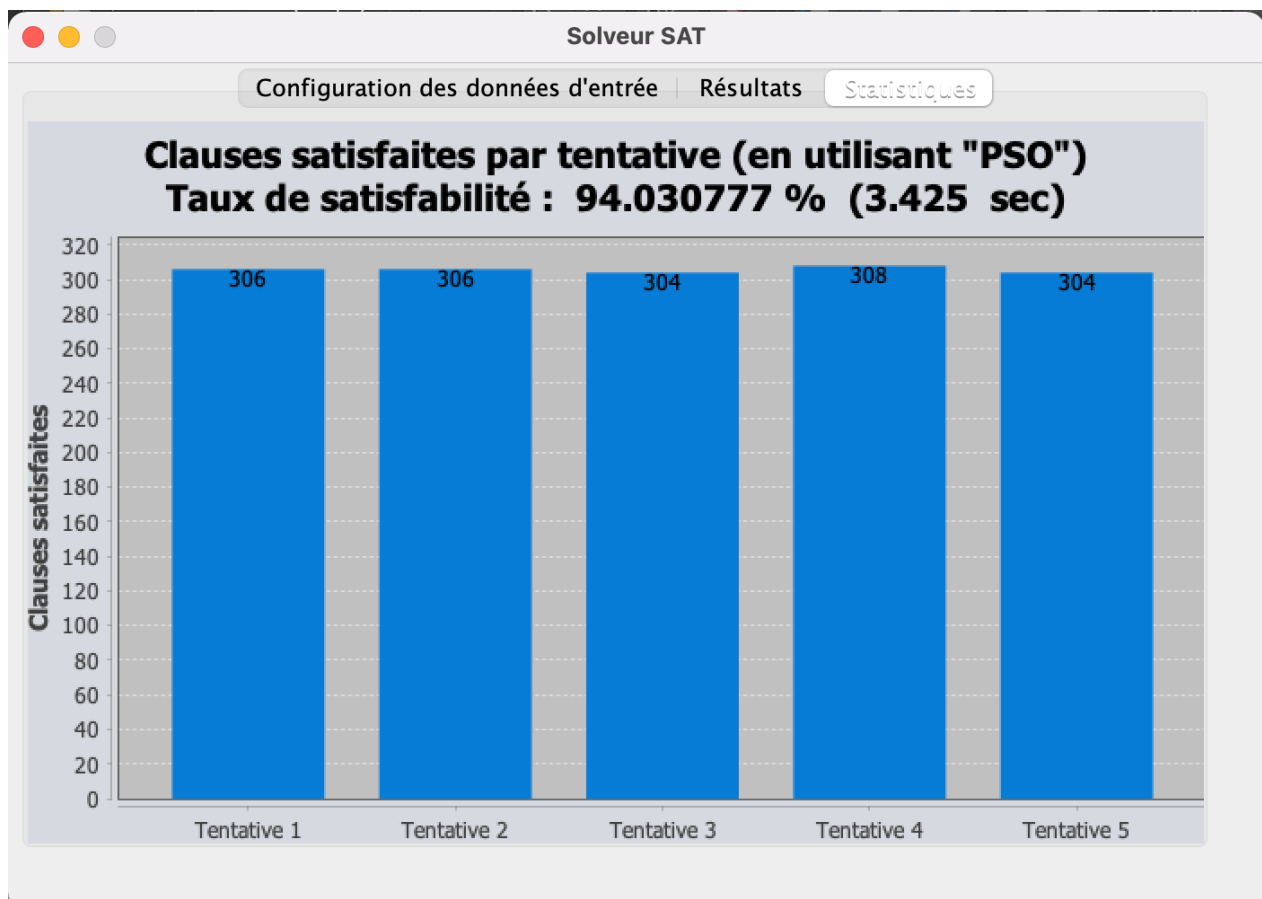
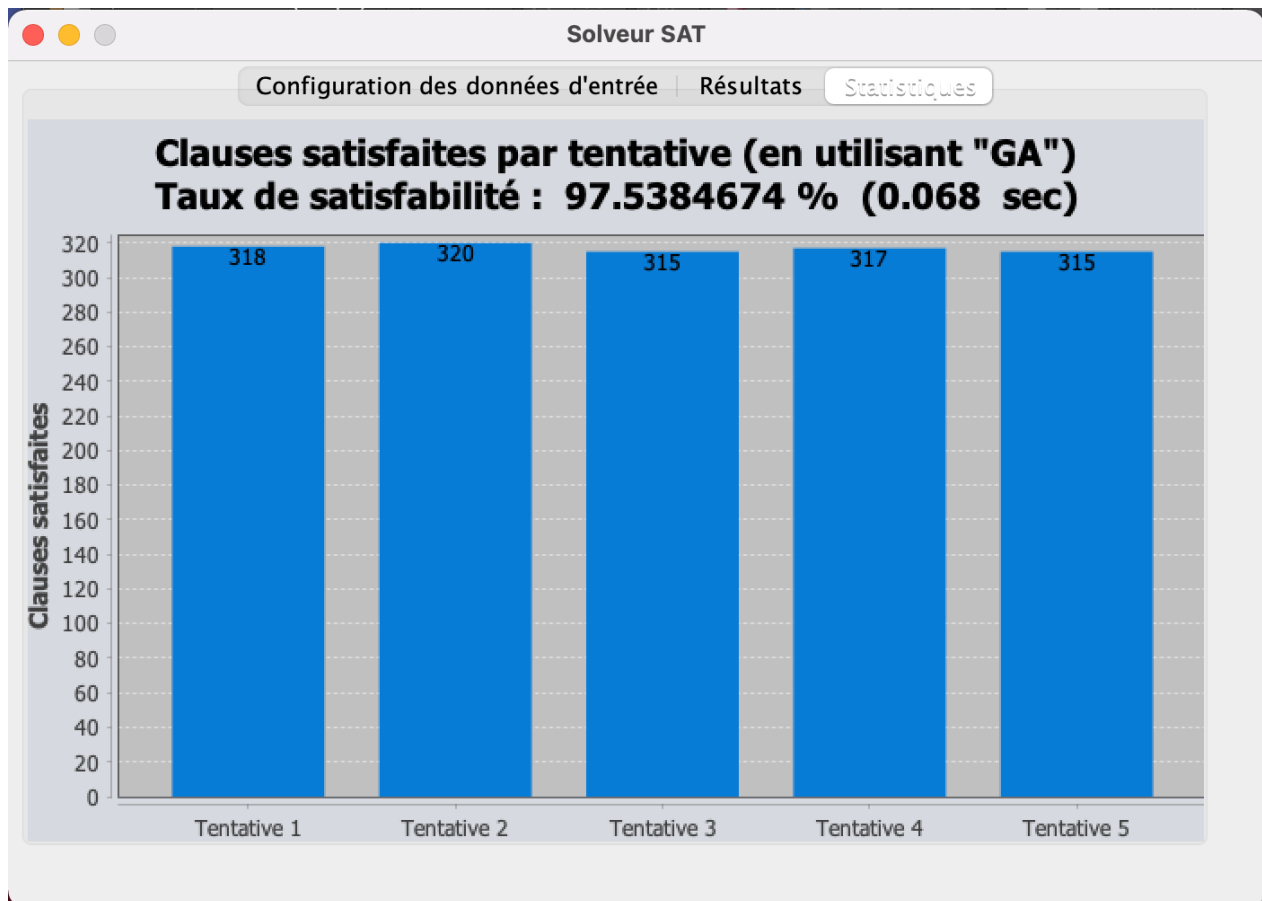


2

1 : Hyper paramètres de l'algorithme génétique GA (configuration d'entrée).

2 : Hyper paramètres de PSO.

4.2 Statistiques du déroulement d'un exemple de GA et PSO :



5 Conclusion :

Ce projet nous a permis d'approfondir nos connaissances en se familiarisant avec des approches sophistiquées pour résoudre des problèmes complexes. Ces derniers, souvent NP-Complets, n'étant pas solvables avec les techniques classiques. Pour cela, les méthodes méta-heuristiques sont utilisées cherchant à optimiser la résolution du problème donné en termes des ressources utilisées, temps d'exécution et de la qualité de la solution.