# Projet PSO MAM3 2020

MALTESE Salomé
THIMON Saphia-Lise
LI Hanwen
DENIS Hansi

Polytech Nice-Sophia

Le 29 Mai 2020

# Introduction et Motivation

- L'optimisation consiste à minimiser (ou maximiser) une fonction. Avec le modèle PSO, on cherche à minimiser une fonction en prenant un essaim de particules qui se déplacent. Leur déplacement est influencé par leur postion mais également celle des autres particules. Le déplacement de chacune d'elle dépend de son "particle best" qui va permettre de se déplacer vers un minimum mais plutôt local puisque celui-ci concerne qu'une particule, et du "global best" qui permet l'intéraction et facilite ainsi la convergence vers un minimum global.

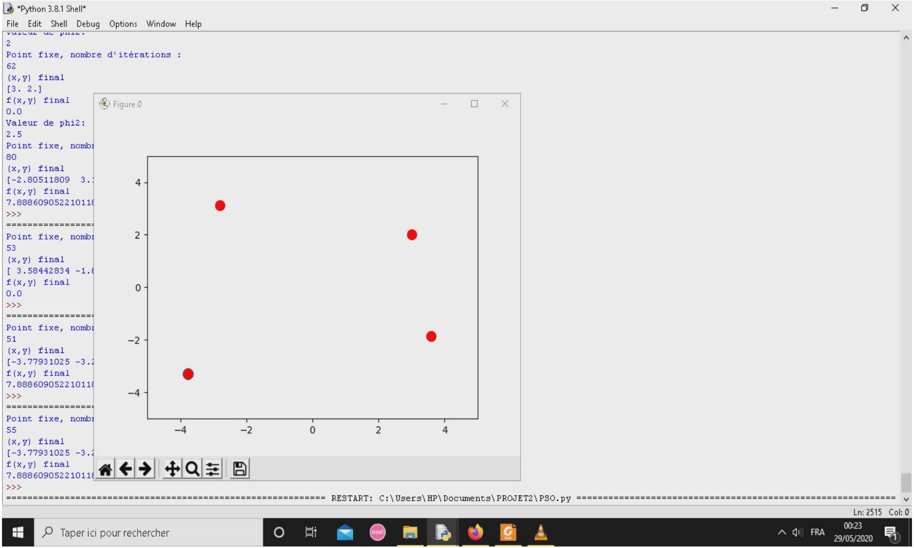
# Algorithme suivi

**for** *chaque particule i* **do**
  **for** *chaque dimension d* **do**
    Initialision aléatoire de la position des $x_i$ dans leur domaine de définition
  **end**
**end**
**while** *nombre maximum d'itérations non atteint* **do**
  mise à jour personal best et global best;
  **for** *chaque particule i* **do**
    calcul de la vitesse puis mise à jour de la position;
  **end**
  **if** *position précédente est la même que l'actuelle* **then**
    **Result:** Minimum global, nombre d'itérations
  **end**
**end**
**Result:** f(x,y) final, nombre d'itérations max

# Tests de validation (Benchmarks de référence)

Nous avions à étudier la fonction d'Himmelblau. Avec des paramètres qui sont $population = 100, \phi 1 = 0.1, \phi 2 = 1.2, w = 0.1$. On a systématiquement une convergence vers l'un des minimums de la fonction en entre 50 et 60 itérations. Exemple de convergence des particules vers l'un des minimums:

# Exemple Himmelblau



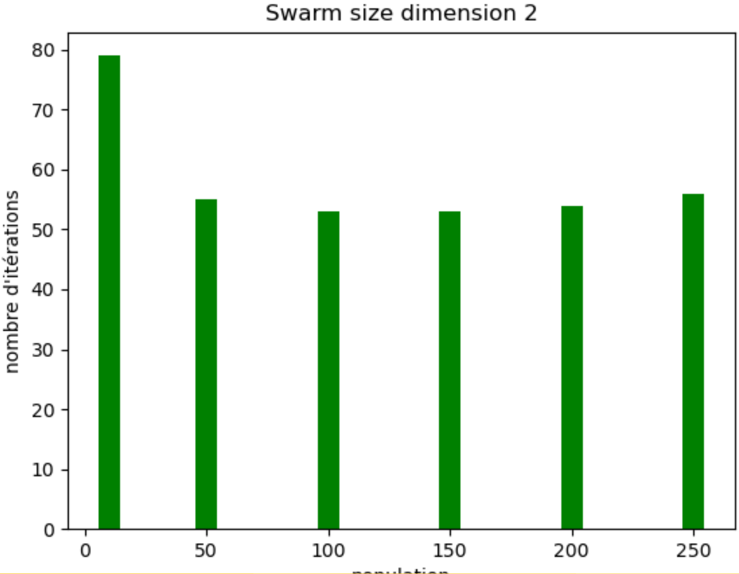
# Tests de validation (Benchmarks de référence)

Dans cet exemple, le paramètre d'interaction sociale est bien plus important que celui d'action individuelle. de se fait il suffit qu l'une des particules soit proche d'un optimum pour que toutes les autres s'agglutinent autour d'elle et que la méthode converge. De plus on peut observer lorsu'on lance la simulation qu'il semble que les particules convergent en bien moins d'itérations que le nombre affiché à la fin. Cela s'explique par le fait que le critère d'arrêt est l'immobilité totale de toutes les particules or on peut avoir après une convergence global des petits mouvements résiduels dûs à l'inertie.

# Étude de l'influence de la taille de l'essaim

Pour étudier l'effet de la taille de l'essaim, nous avons tester l'algorithme avec différentes tailles de population : 10,50,100,150,200,250. Nous avons ensuite afficher le nombre d'itérations en fonction de la taille de l'essaim.

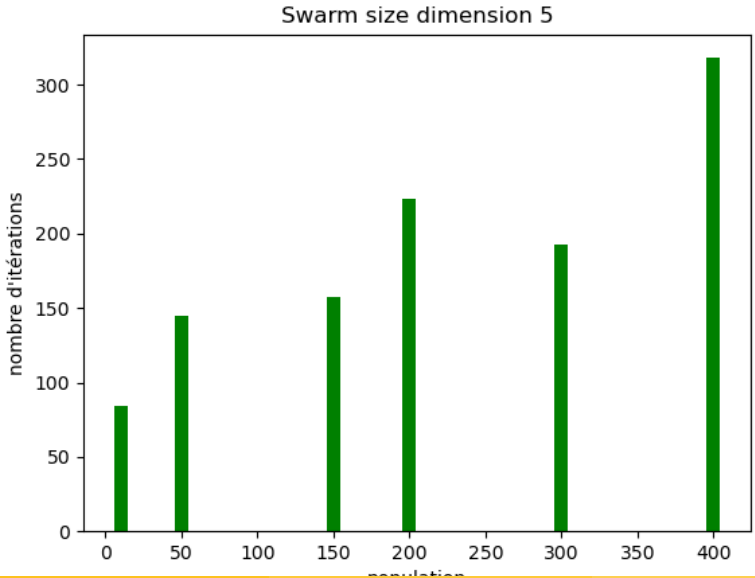
# Étude de l'influence de la taille de l'essaim



# Étude de l'influence de la taille de l'essaim

La taille de l'essaim a une influence modérée sur la vitesse de convergence vers un minimu. On observe que pour que la recherche de l'optimum soit efficace, il est nécéssaire que l'essaim ait une certaine taille . Ici en 2d on voit qu'à partir de 50 particules on peut voir une réduction du nombre d'itérations. Au delà, la vitesse de convergence ne bouge presque pas. Pour n=10, la méthode met un peu plus de temps à converger. Il semble donc qu'il y une taille optimale pour l'utilisation de la méthode, suffisamment grande pour que la méthode converge le plus rapidement possible, pas trop grande pour que l'algorithme tourne au plus vite. Dans notre exemple ci-dessus, n=100 semble être le bon compromis.

# Étude de l'influence de la taille de l'essaim



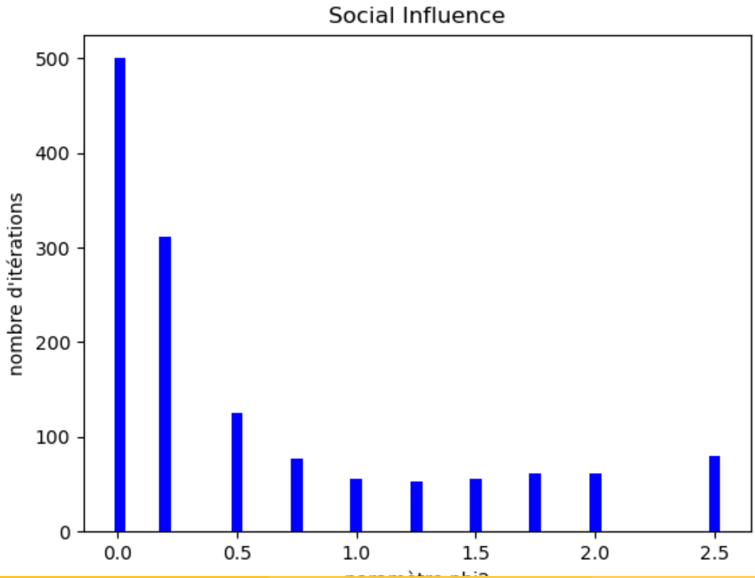
# Étude de l'influence de la taille de l'essaim

Ici en dimension 5 on peut observer sensiblement le même phénomène, le nombre optimal semble être aux alentours de 100. On ne doit pas se laisser duper par la valeur en 10 qui correspond est faible car on est arrivé à un état immuable sans atteindre l'optimum.

# Étude de l'influence sociale

Il faut étudier la partie $\phi 2 U_2^t(gb^t - x_i^t)$ de la formule du calcul de la vitesse.
Pour cela, nous avons tester notre algorithme avec plusieurs $\phi 2$ et nous avons observer le nombre d'itérations pour voir si ça converge ou non et si oui, à quelle vitesse selon $\phi 2$.
Nous avons fait un diagramme qui montre le nombre d'itérations en fonction de $\phi 2$. Nous avons tester pour les valeurs suivantes : 0.01,0.2,0.5,0.75,1.0,1.25,1.5,1.75, 2 et 2.5. Notre étude a été réalisée en 2d sur la fonction de Himmelblau.

# Étude de l'influence sociale



# Étude de l'influence sociale

- Pour $\phi 2 = 0.01$, il y a 500 itérations, ce qui correspond à notre nombre d'itérations maximum. La boucle s'est arrêtée mais la méthode n'a pas convergé. En mettant $\phi 2$ à une valeur quasiment nulle, nous enlevons l'intéraction entre les particules. Chaque particule ne sait pas où est le global best, et se déplacement seulement en fonction de sa position et du particle best. Les particules ne se déplacent donc que très peu, elles restent dans une zone autour de leurs positions de départ.
- Pour $\phi 2 = 2.5$, on remarque que la méthode mets plus de temps à converger. Il faut itérations, alors que pour les valeurs entre 1 et 2 de $\phi 2$, le nombre d'itérations ne dépasse pas 60. Ici, les particules vont plus se déplacer vers le global best, même si celui-ci n'est pas proche de l'optimum.

# Étude de l'influence sociale

- Pour les autres valeur de $\phi 2$, la méthode converge plus ou moins à la même vitesse : 310 itérations pour $\phi 2 = 0.2$, 150 itérations pour $\phi 2 = 0.5$, 50 itérations pour $\phi 2 = 1$

Finalement, on observe que les particules ont besoin de savoir où est le global best, donc d'échanger des informations sur les positions de chacunes pour que la méthode puisse converger de manière optimale.

# Conclusion

Pour conclure nous pouvons dire que le Modèle PSO se montre très efficace de part son applicabilité en dimension quelconque et se démarque des méthodes que l'on a vu en cours. Premièrement elle fait appel à l'aléatoire, ce qui nous a surpris au début. En second lieu son fonctionnement basé sur l'interaction entre les particules inspiré du monde animal est intriguant. On voit alors que la nature est bien faite.

# Annexe

Code python
Algorithme modèle