



Mini Projet

PSO

Practical swarm optimization

Présenté par : Azami youssef

Plan

Nos points de discussion clés



Remarque: Nous avons illustrer le fonctionnement de ce principe en utilisant un exemple afin de pouvoir toucher le volet pratique de notre algorithme

■ Introduction

■ Voyageur de Commerce

■ Principe de fonctionnement

■ Reseaux de neurone

■ Domaine D'application

■ Demonstration



Introduction



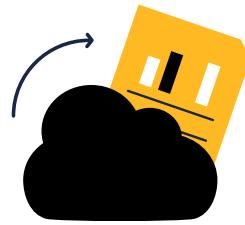
Définition

La méthode PSO est une méta-heuristique qui permet de trouver l'optimum d'une fonction en un temps de traitement raisonnable



Historique

modèle développé par Craig Reynolds à la fin des années 1980, Une autre source d'inspiration, revendiquée par les auteurs, James Kennedy et Russel Eberhart, est la socio-psychologie.



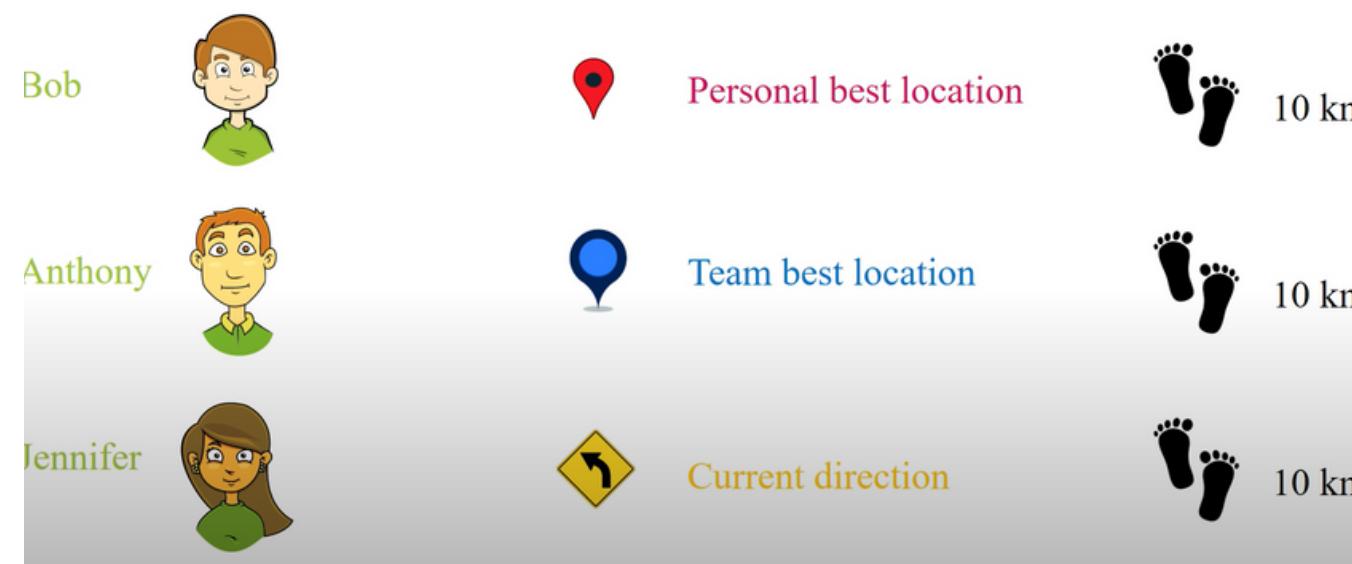
Origine

Cet algorithme s'inspire à l'origine du monde du vivant,Cette méthode d'optimisation se base sur la collaboration des individus entre eux. Elle a d'ailleurs des similarités avec les algorithmes de colonies de fourmis.



principe de fonctionnement

PSO search strategy



PSO search strategy



PRINCIPE GENERAL

Soit un ensemble d'archéologues qui cherche à trouver un emplacement spécifique donc il faut décider de ce séparer pour agrandir la zone de recherche en se déplaçant en utilisant le principe de practical swarm optimisation

REGLES DE DÉPLACEMENT

Chaque archéologue doit informer les autres s'il a un changement de Gbest ainsi les points de départ et les directions sont choisis aléatoirement et chacun doit mémoriser son Pbest location et Gbest



principe de fonctionnement

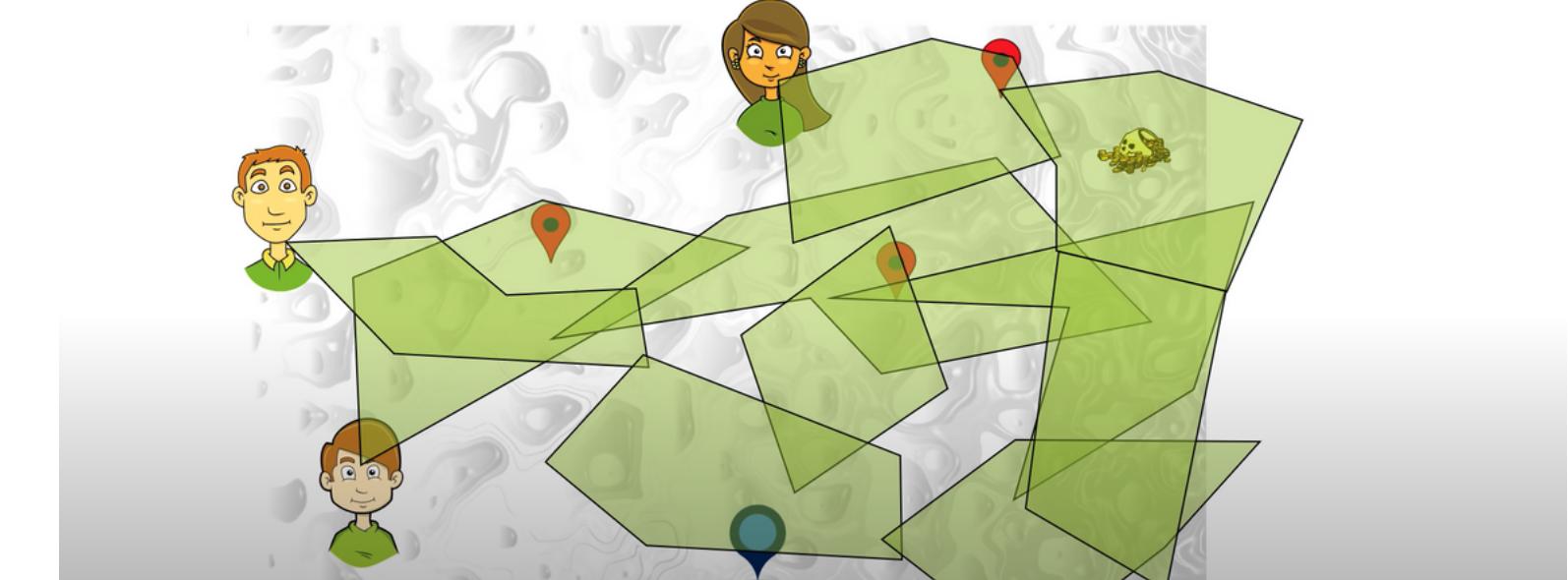
PSO search strategy



$$2 \times r \times 10 \text{ km}$$

$r \text{ in } [0,1]$

PSO search strategy

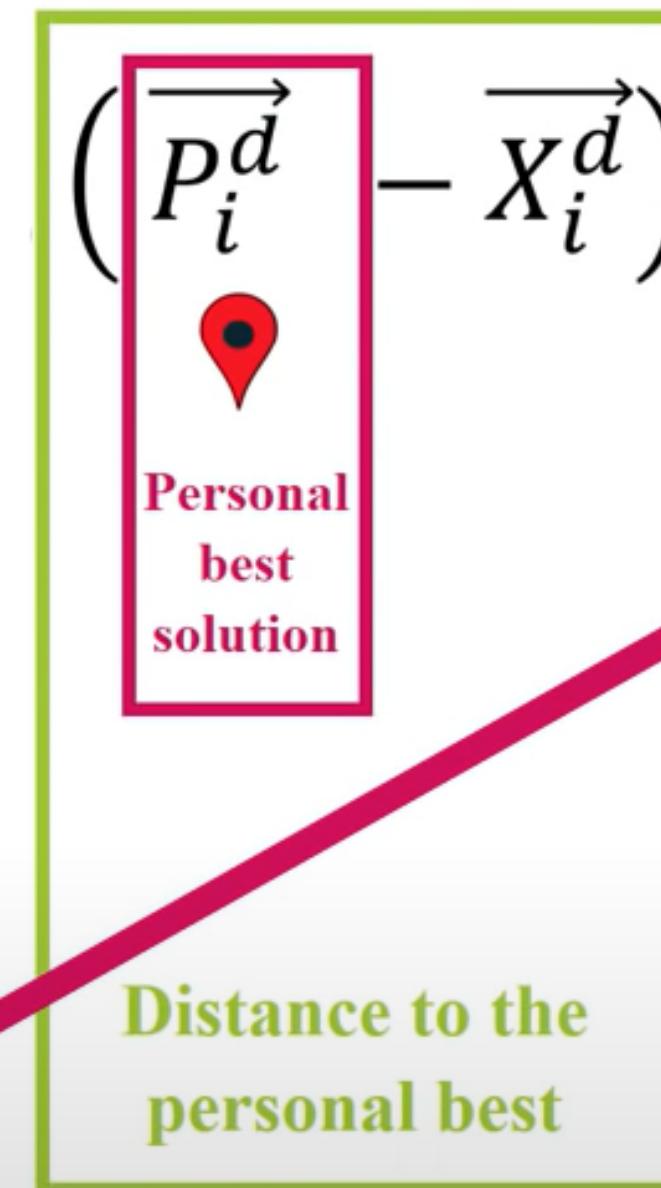
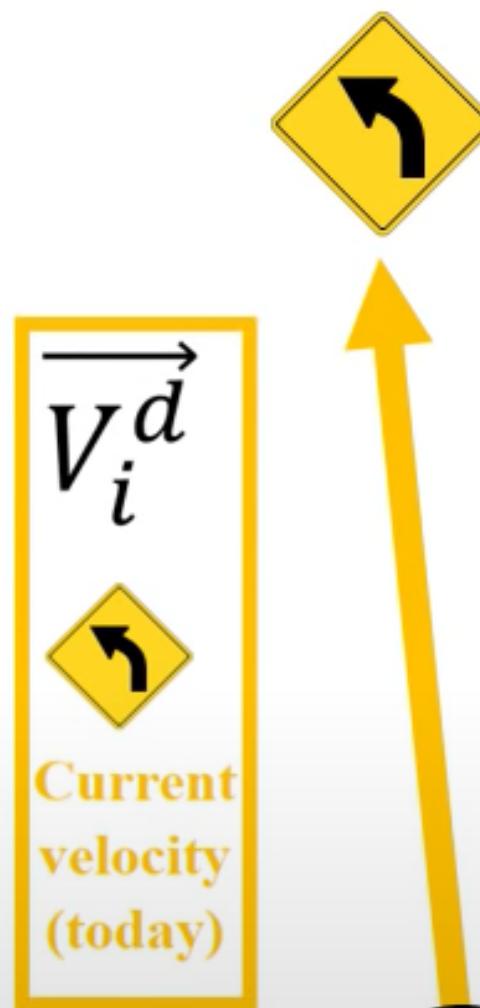


PRINCIPE DE FONCTIONNEMENT

La distance à parcourir chaque jour va varier en utilisant un facteur "r" une valeur comprise entre 0 et 1
le point de départ de chaque jour sera le point d'arrêt la nuit qui précède ainsi tous les particules effectue la même opération ce qui va se finaliser par un recouvrement totale d'un emplacement de recherche dans une durée minimal

PSO search strategy

Current direction



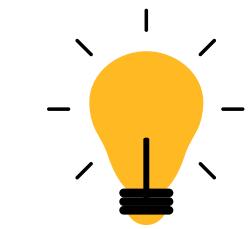
$$(\vec{G}^d - \vec{X}_i^d)$$

Global
best
solution

Distance to the global best



Personal best s



Mis en ouvre

Vi c'est la vitesse qui est en relation avec la vitesse et la direction de déplacement et en utilisant des information vers le Gbes et Pbest nous pouvant savoir la prochaine position est par cumule d'itération le minimum globale



Global best sol

$$P_i^{t+1} = P_i^t + V_i^{t+1}$$

$$V_i^{t+1} = wV_i^t + c_1 r_1 (P_{best(i)}^t - P_i^t) + c_2 r_2 (P_{bestglobal}^t - P_i^t)$$

Inertia



Inertia

Permet à chaque particule de se déplacer dans la même direction avec la même vitesse afin de contrôler les déplacements dans une région de best positions

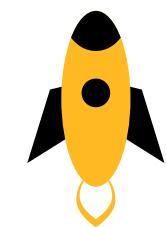
Cognitive (Personal)



Cognitive

C'est l'influence Personnelle sur le groupe cette influence permet d'améliorer la position actuelle de particule individuellement

Social (Global)

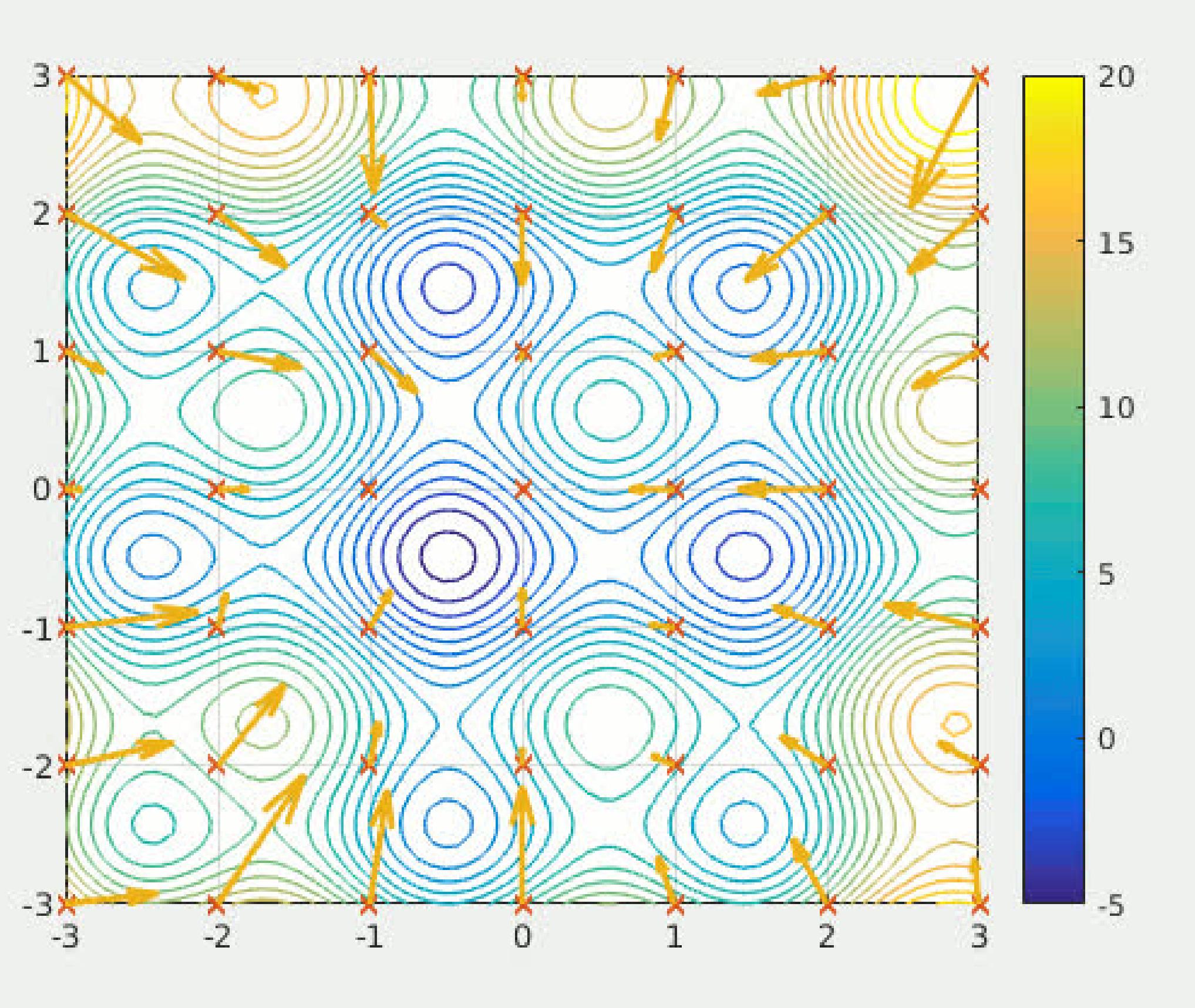


Social

Permet au particule le suivre le meilleur chemin en se référant au Best globale

PSO

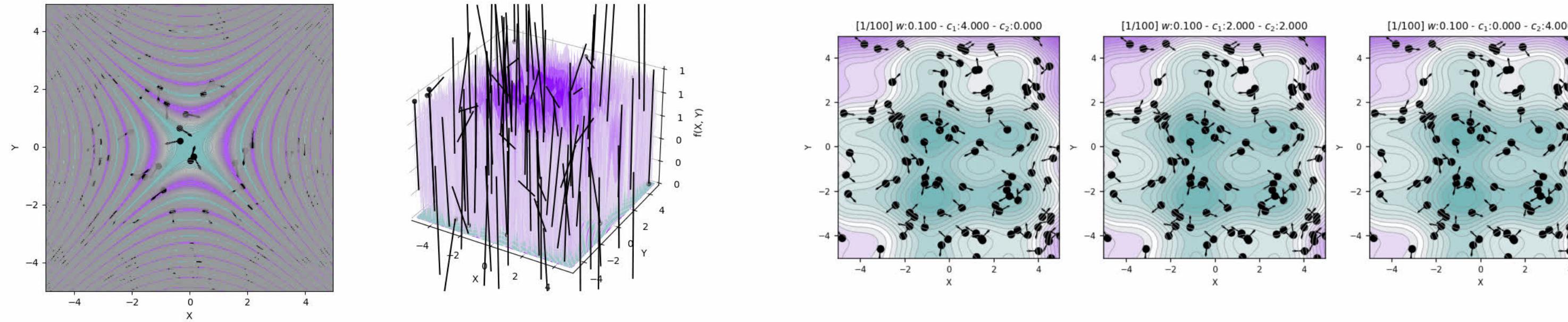
Explore l'espace de recherche grâce à des essais successifs de positions de boids, leurs mouvements étant générés par des équations simples



Intuitivement, les actions des différents boids de l'essai permettent simultanément d'explorer l'espace de recherche et d'exploiter les zones les plus prometteuses. Un grand nombre de variantes de cet algorithme ont été développées et sont utilisés dans divers cadres d'application



Variation des paramètre c1,c2 et W



INTERIA NULL

une interia null provoque une malle organisation des paticule qui entraîne une mal repartition dans l'espace de recherche

VARIATION DE C1 ET C2

Dans le premier cas nous avons choisi un c2 null ceci et traduit par un mouvement limité des particule en se basant sur le facteur personnele et dans le troisieme cas nous avon choisi un c1 null ce qui a influence le nombre d'iterateration pour arivé a un minimum globale



Mini Projet

Domaine d'application

Domaine d'application de PSO

Problème

Lennard-Jones atomic cluster

Gear train design

Frequency modulation sound parameter identification

Spread spectrum radar poly-phase code design

Compression spring design

Perm function

Mobile network design

Pressure Vessel design

Welded Beam design

Speed Reducer design

Problème de contraintes 1

Problème de contraintes 2



Mini Projet



Domaine d'application

Pso est utilisé dans mainte problème très sophistiqué comme atomique clusture atomique pour les mutation et le design des ligne téléphonique ainsi que le design des train chifters



Pouvoir de PSO

Pso se met en compétition avec des grand algorithme d'optimisation et algorithme de résolution de problèmes comme les algorithme de Christofides



Mini Projet

Voyageur de Commerce

Voyageur de Commerce



Voyageur de commerce

DÉFINITION

Problème du commis voyageur, est un problème d'optimisation qui consiste à déterminer, étant donné une liste de villes et les distances entre toutes les paires de villes², le plus court circuit qui passe par chaque ville une et une seule fois.

DESCRIPTION DU PROBLÈME

étant donné n points (des « villes ») et les distances séparant chaque point, trouver un chemin de longueur totale minimale qui passe exactement une fois par chaque point et revienne au point de départ

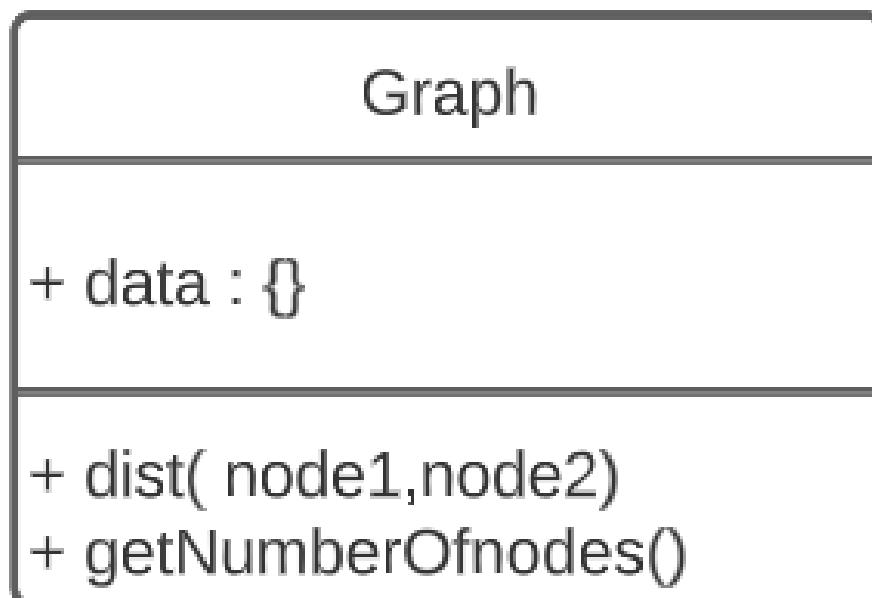
Les Classes utilisés



GRAPH



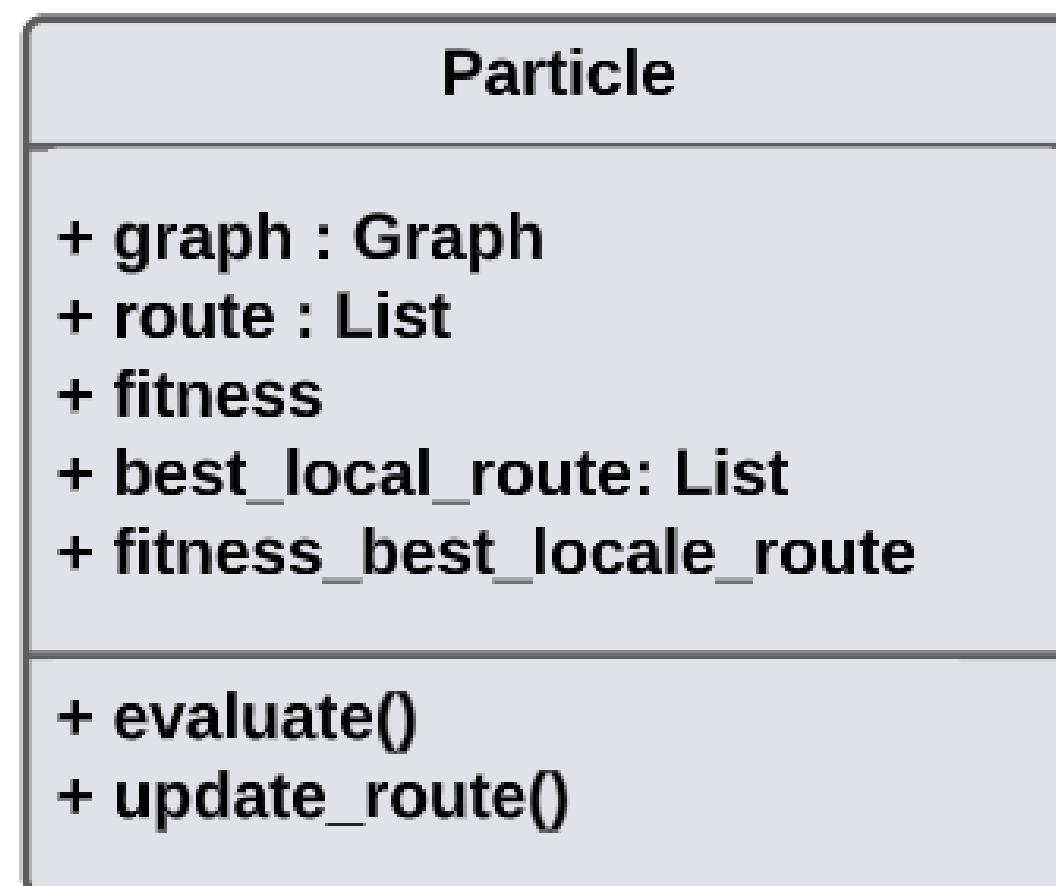
une instance est un graphe complet G avec un ensemble de sommets, un ensemble d'arêtes et une fonction de coût sur les arcs. Le problème est de trouver le plus court cycle hamiltonien dans le graphe G.



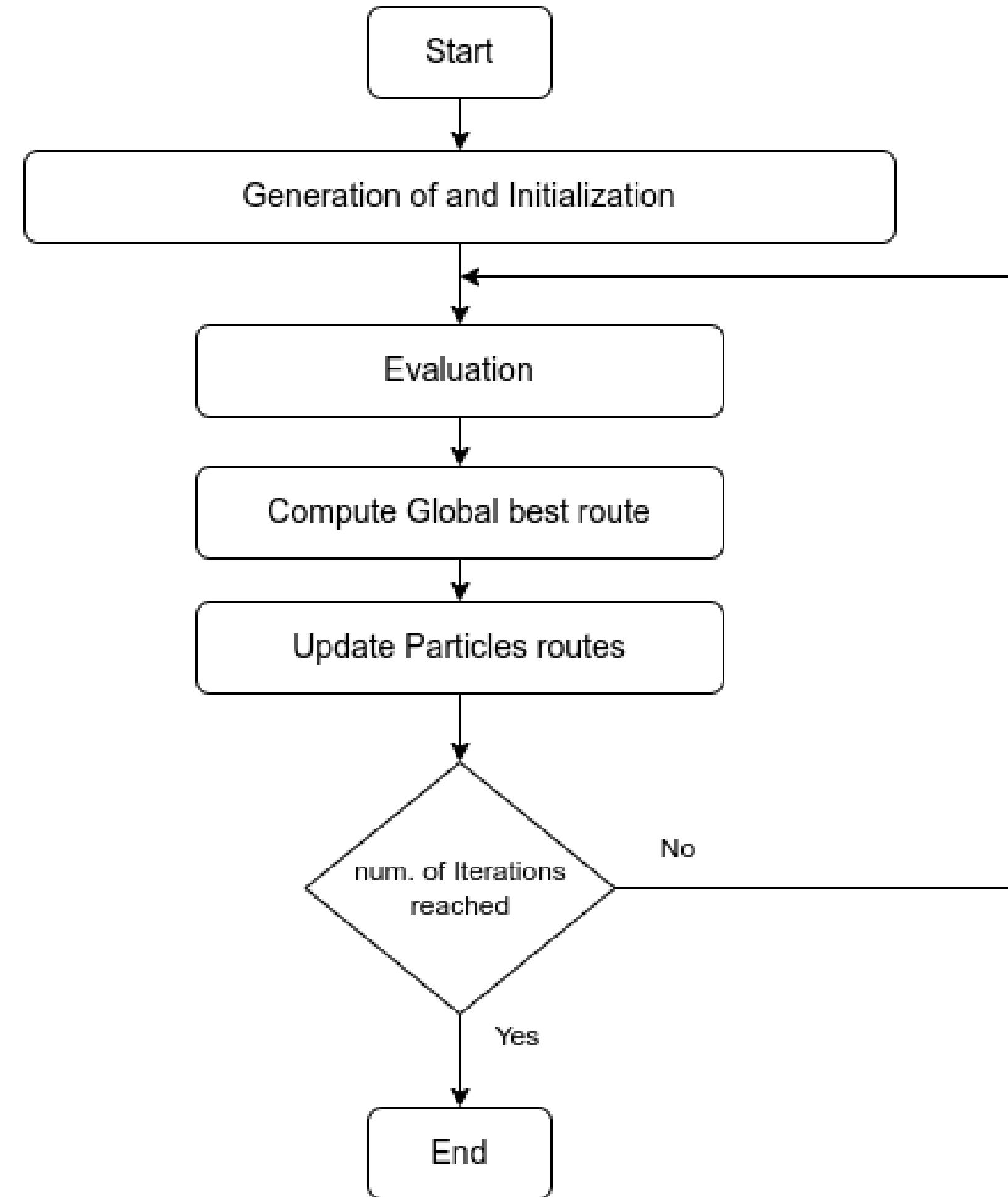
PARICULE



une particule est définie par une instance d'une graph une fitness et un best local route et les méthodes evaluate qui permet de calculer la fitness ou le meilleur chemin et update_route qui permet d'optimiser le chemin



Processus de développement



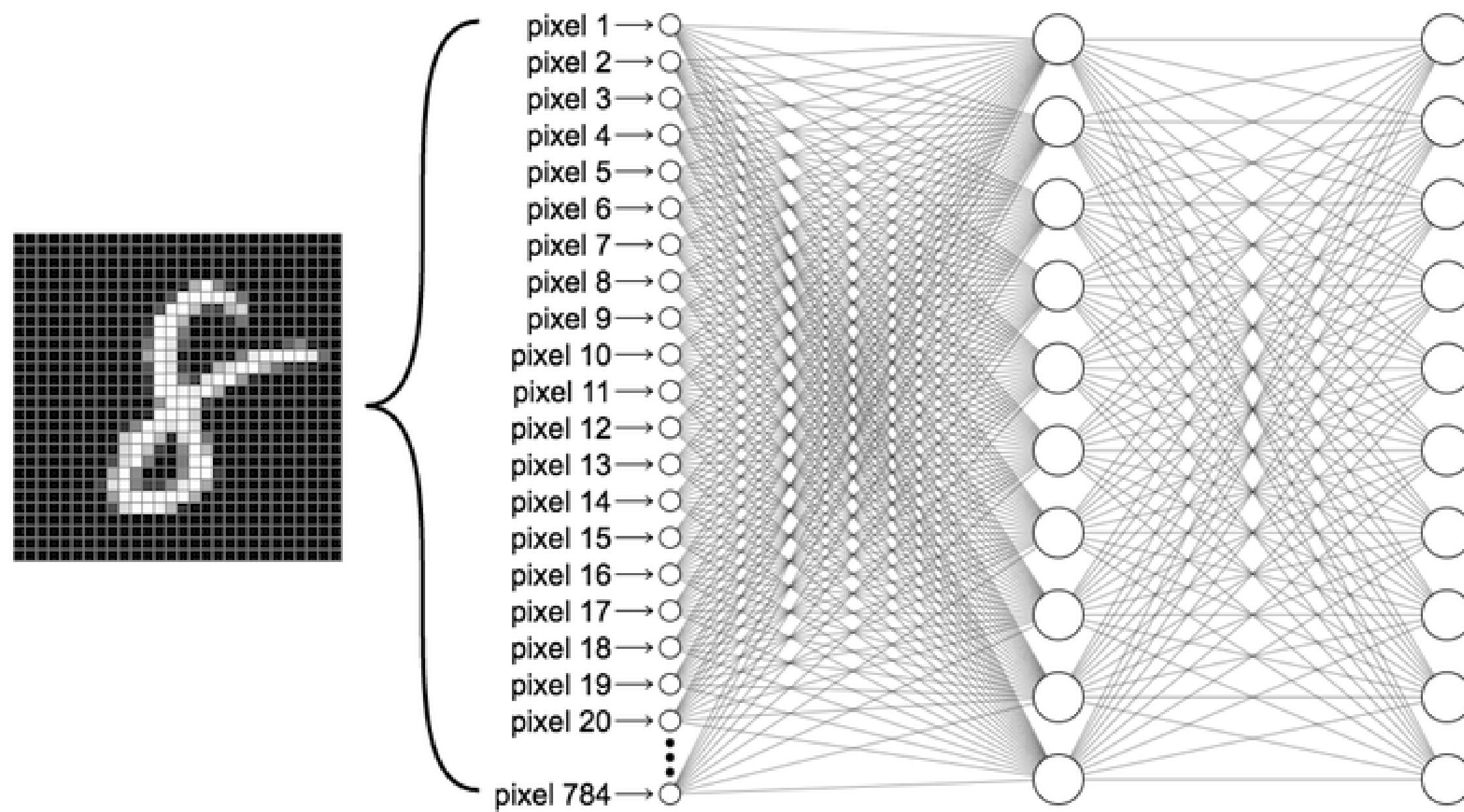


Mini Projet

Réseau de Neuron

Réseau de Neuron

Notre Réseau de Neuron (MNIST)



Les Classes utilisés



PARICULE



NEURAL_NETWORK



Particle

```
+ input_layer_size  
+ hidden_layer_size  
+ output_layer_size  
+ fitness  
+ W1  
+ W2  
+ velocity  
+ fitness_local_best  
+ local_best_W1  
+ local_best_W2
```

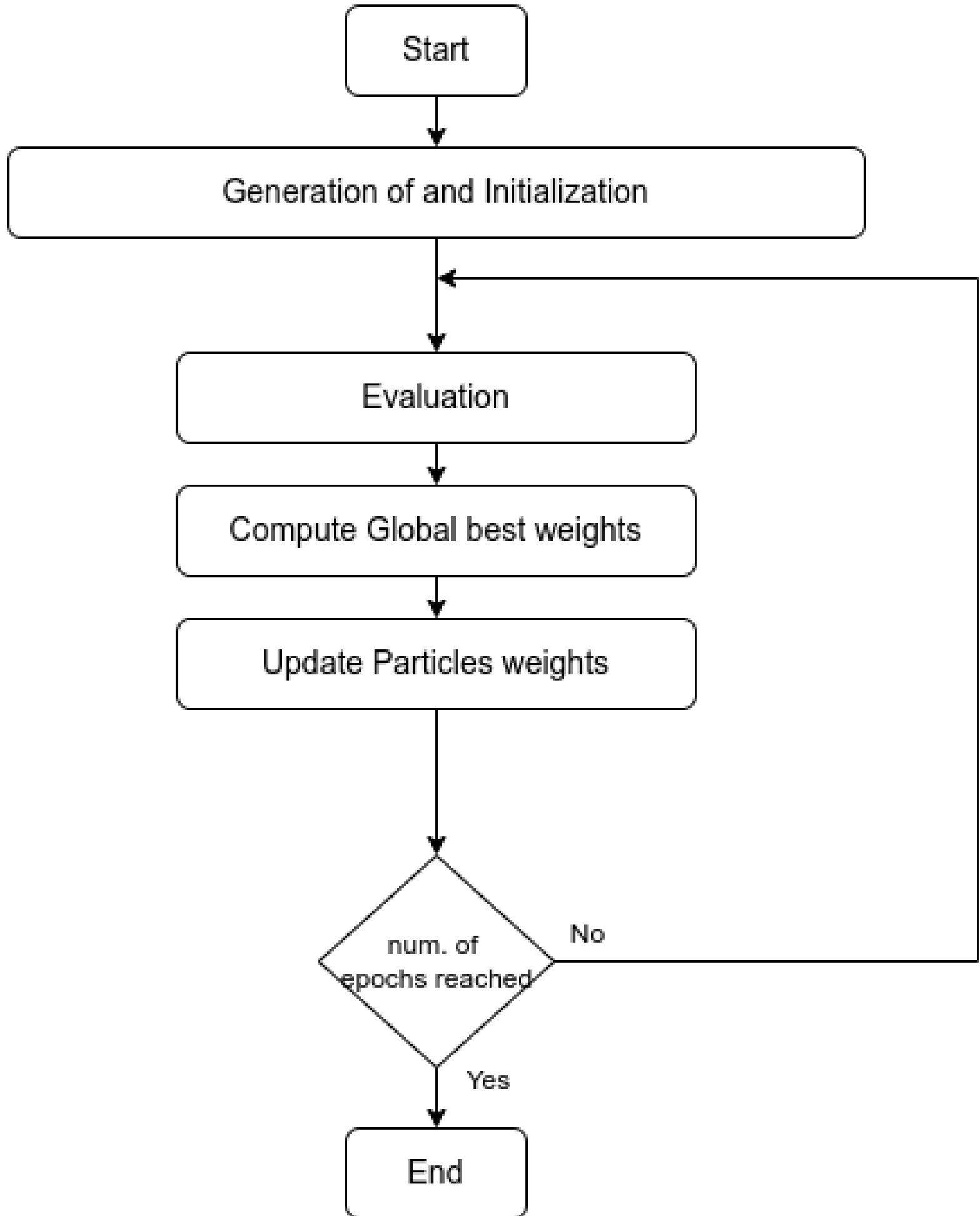
```
+ evaluate( x, test_list)  
+ forward_pass( inputs, targets)  
+ update_weights( g_W1, d_W2, w, c1, c2)  
- accuracy( x)
```

Neural_Network

```
+ input_layer_size  
+ hidden_layer_size  
+ output_layer_size  
+ epochs  
+ learning_rate  
+ particles_number  
+ swarm : Particle[]  
+ fitness_global_best  
+ global_best_W1  
+ global_best_W2
```

```
+ train( train_list, test_list, w, c1, c2)
```

Processus de développement





Mini Projeto

Demonstration

Demonstration



Mini Projet

Merci !



N'hésitez pas à nous Poser des
question!!!!

