

Ajouter legende sur la figure output  
La figure output a tendance a ralentir l'optimization

# MOPSO (Multi-Objective Particle Swarm Optimization) sous Matlab

Judicaël AUBRY

Centre de Recherche en Ingénierie de l'ESTACA, Laval, France

`judicael.aubry@estaca.fr`

25 septembre 2013

## Introduction

Ce document est un mode d'emploi pour une implémentation particulière d'un algorithme d'optimisation multi-objectifs basé sur la métaheuristique des essaims particuliers.

Je considère comme pré-requis la notion d'optimisation multi-objectifs et toutes les notions relatives à la définition d'un problème d'optimisation (i.e. fonction objectifs, fonctions contraintes, paramètres d'optimisation etc.).

Je considère aussi que le lecteur sera allé voir en quoi consiste la méthode des essaims particuliers et notamment la terminologie qui y est associée.

Je m'attacherai plus dans ce document à expliquer les points de détails à la bonne utilisation de ce programme Matlab.

## MOPSO.m

Le fichier MOPSO.m contient la fonction principale de l'algorithme, c'est à dire la boucle *while* qui s'effectue *N\_iterations* fois. Il contient aussi toutes les fonctions auxquelles fait appel la boucle principale, notamment la fonction de tri pour extraire le front de Pareto par exemple.

L'algorithme est implémentée sous la forme d'une fonction Matlab :

$[Front\_Pareto, Front\_Pareto\_Objectifs] = \text{MOPSO}(options)$

**Arguments :** *options* qui est une structure avec différents champs permettant de configurer l'algorithme. Nous reviendrons sur les champs de l'argument *options* dans la suite.

**Valeurs de sortie :** deux tableaux *Front\_Pareto* et *Front\_Pareto\_Objectifs* qui constituent les résultats attendus de l'algorithme.

*Front\_Pareto* est un tableau de taille  $N\_parametres \times N\_archive$  qui contient donc tous les paramètres des solutions se trouvant sur le front de Pareto. Les particules sont triées selon le sens croissant de la première fonction objectif.

*Front\_Pareto\_objectifs* est un tableau de taille  $N\_objectifs \times N\_archive$  qui contient donc les valeurs des fonctions objectifs des solutions se trouvant sur le front de Pareto. Les particules étant toujours triées selon le sens croissant de la première fonction objectif.

## Explication détaillée de la structure *options*

**options.AlgParams** contient les principaux paramètres de l'algorithme

**options.AlgParams.N\_particules** valeur du nombre de particules. Correspond au nombre d'évaluations des fonctions objectifs à chaque itération.

valeur typique : de 30 à 100 particules par variable a été pour moi un bon compromis. Sachez l'adapter à votre temps de calcul et à vos résultats obtenus

**options.AlgParams.N\_iterations** Nombre d'itérations à effectuer avant arrêt de l'algorithme. Une possibilité de relance à partir d'un arrêt est possible.

valeur typique : La moitié du nombre de particule. Si l'algorithme n'évolue plus, il n'y a pas de raison d'insister. **Conseil :** Relancer plusieurs fois l'optimisation pour le même problème. Puis faites une fusion des résultats obtenus. Cela permet d'éliminer les mauvaises convergences.

**options.AlgParams.N\_variables** Nombre de variables d'optimisation du problème. Peut-être en pratique déterminé à partir du tableau *Domaine* qui contient les valeurs minimales et maximales que peuvent prendre les variables d'optimisation.

**options.AlgParams.N\_archive** A chaque itération, les solutions optimales (au sens du front de Pareto) sont sauvegardées. Elles sont fusionnées dans une archive des solutions optimales trouvées précédemment. Cette archive contient donc l'ensemble des solutions optimales trouvées depuis le lancement de l'algorithme. Pour éviter l'explosion de la taille de cette archive, une stratégie de limitation de taille a été programmée. Ce paramètre correspond donc à la taille limite que l'on souhaite donner à cette archive. Il est

important de noter que plus sa taille est importante, plus l'actualisation et la fusion prendra du temps à chaque itération (complexité en  $N \cdot \log(N)$ ).

**options.StraParams** contient les paramètres relatifs à la métaheuristique des essaims particuliers.

**options.StraParams.Accel\_memoire** Accélération cognitive : tendance qu'une particule aura à aller vers sa meilleure position en mémoire à l'itération suivante.

min, max ?

Valeur typique : 0.7 à 1

**options.StraParams.Accel\_guide** Accélération sociale : tendance qu'une particule aura à aller vers la meilleure particule de l'essaim à l'itération suivante. En multi-objectif, il n'y a pas une mais plusieurs meilleures particules. Une stratégie a donc été définie dans la fonction *choisir\_guides*.

min, max ?

Valeur typique : 0.7 à 1

**options.StraParams.Inertie\_debut et Inertie\_fin** tendance qu'une particule aura à suivre sa trajectoire dans l'espace des paramètres à l'itération suivante.

min, max ?

Valeur typique : 0.8 à 1.2

**options.StraParams.Proba\_mut** pour rajouter un caractère aléatoire à l'algorithme qui serait sinon entièrement déterministe en dehors du tirage initial des positions des particules, une certaine proportion des particules sont déplacées aléatoirement autour de leurs positions.

min, max ?

Valeur typique : 0.01 (ou 1%)

**options.StraParams.Fact\_constrict** facteur qui permet de diminuer au fil des itérations les valeurs précédemment définies. L'intérêt est de permettre, une fois l'exploration du début terminée, d'intensifier la recherche locale sur le front de Pareto.

typical,  
min, max ?

**options.Objectif** contient les informations relatives à la fonction objectif

**options.Objectif.fonction** Handle Matlab de la fonction objectif. Nous allons préciser dans la suite quels sont les arguments et les valeurs de retour qu'il faut prévoir pour cette fonction.

**options.Objectif.Domaine** Tableau de taille  $N\_variables \times 3$  qui contient pour toutes les variables du problème, la valeur minimale et maximale ainsi que, en 3ème colonne, 0 si la variable est de type continue, ou une valeur non nulle correspondant au pas de discrétisation de la variable si elle est de type discrète (ex : nombre de paires de pôles  $\rightarrow 1$ )

**options.Sauvegarde** contient les informations relatives à la sauvegarde des résultats.

**options.Sauvegarde.etat** true ou false : si on souhaite ou pas sauvegarder le résultat

**options.Sauvegarde.fichier** nom que l'on souhaite donner au fichier de sauvegarde.

**options.Initialisation** contient les informations relatives à l'initialisation de l'algorithme à partir d'un fichier de sauvegarde précédent. Utile pour reprendre une optimisation non terminée ou lorsque l'on s'est rendu compte que le nombre d'itération n'a pas été suffisant.

**options.Initialisation.etat** 0 = initialisation aléatoire, 1 = initialisation à partir du fichier renseigné, 2 = Initialisation uniquement du front de Pareto sauvegardé dans le fichier, les particules sont tirées aléatoirement dans l'espace des paramètres, 3 = Initialisation uniquement du dernier essaim sauvegardé, le front de Pareto n'est pas rechargé.

**options.Initialisation.fichier** nom du fichier à partir duquel on souhaite effectuer l'initialisation

**options.Affichage.etat** true si l'on souhaite lancer la fonction *affichage* (fonction à définir en fonction de ses besoins se situant à la fin du fichier mopso.m) à chaque itération et avoir une image de l'évolution du Front de Pareto.

## Présentation des différentes variables mises en jeu dans le programme

**Domaine** tableau de taille  $N_{variables} \times 3$

contient pour toutes les variables du problème, la valeur minimale et maximale ainsi que, en 3ème colonne, 0 si la variable est de type continue, ou une valeur non nulle correspondant au pas de discrétisation de la variable si elle est de type discrète (ex : nombre de paires de pôles  $\rightarrow 1$ )

**Essaim** tableau de taille  $N_{variables} \times N_{particules}$

contient la position actuelle des particules dans l'espace des paramètres.

**Objectifs** tableau de taille  $N_{objectifs} \times N_{particules}$

contient les valeurs actuelles des fonctions objectifs des particules de l'essaim

**Contraintes** tableau de taille *ça dépend du nombre de contraintes*  $\times N_{particules}$

contient les valeurs actuelles des fonctions contraintes des particules de l'essaim

**Divers** tableau de taille *ça dépend de vous*  $\times N_{particules}$

contient les valeurs actuelles de résultats renvoyés par la fonction objectif mais qui ne sont ni des objectifs ni des contraintes mais pouvant être utiles pour une analyse en post-traitement. (ex objectif minimisation de la masse totale d'un moteur, valeurs dans divers masses des différentes matières premières : cuivre fer aimants...)

**Vitesses** tableau de taille  $N_{variables} \times N_{particules}$

contient le dernier déplacement effectué par chaque particule dans l'espace des paramètres (peu utile)

?

**Memoires\_Variables** tableau de taille  $N\_variables \times N\_particules$

Le préfixe Mémoires est relatif à la meilleure position obtenue par chaque particule depuis le début des itérations.

**Memoires\_Objectifs** tableau de taille  $N\_objectifs \times N\_particules$

**Memoires\_Contraintes** tableau de taille *ça dépend du nombre de contraintes*  $\times N\_particules$

**Front\_Pareto\_Variables** tableau de taille  $N\_variables \times N\_archive$

Le préfixe Front\_Pareto est relatif à toutes les solutions archivées, c'est à dire optimales au sens de Pareto

**Front\_Pareto\_Objectifs** tableau de taille  $N\_objectifs \times N\_archive$

**Front\_Pareto\_Contraintes** tableau de taille *ça dépend du nombre de contraintes*  $\times N\_particules$

**Front\_Pareto\_Divers** tableau de taille *ça dépend de vous*  $\times N\_particules$

## Comment construire la fonction objectif

La fonction objectif que vous devez définir doit respecter certaines règles. Elle doit avoir le typage suivant :

```
function [Objectifs,Contraintes,Divers]=Fonction_objectif(Essaim)
```

Elle prend donc en paramètres l'essaim, qui est en fait un tableau de variables d'optimisation. Il faut donc à l'intérieur de cette fonction évaluer chacune de ces solutions. Le plus simple est donc de faire une boucle sur  $N\_particules$ .

Elle doit renvoyer trois tableaux :

**Objectifs** les valeurs des fonctions objectifs à **minimiser**. Si vous souhaitez maximiser des fonctions (ex :  $\eta$ ), je vous conseille de renseigner l'opposé de ce que vous voulez maximiser (ex :  $-\eta$ ).

**Contraintes** les valeurs des contraintes qui doivent être négatives si la contrainte est respectée et positive si elle ne l'est pas. Je conseille de définir vos contraintes de la manière suivante :

*Contrainte température = Température - ~~Température limite~~ max*

Il est possible au premier tirage de la population et pendant les premières itérations, qu'aucune solution "viable" ne soit trouvée. Une minimisation d'une somme des contraintes normalisée est alors effectuée jusqu'à obtenir au moins une solution viable. Le décompte des itérations ne débute qu'à partir du moment où une solution viable est trouvée. L'algorithme a le droit à 40 itérations pour trouver une solution, au delà il s'arrête.

**Divers** peut être un tableau vide. Si vous souhaitez analyser certaines valeurs en post traitement dans le fichier de sauvegarde, mais qui n'interviennent pas explicitement dans les objectifs ou les contraintes, vous pouvez les mettre ici.