
NOUVEAUX ALGORITHMES HYBRIDES POUR L'OPTIMISATION GLOBALE

II.1. INTRODUCTION

Ce chapitre constitue l'une des contributions principales de ce travail de thèse. Il présente une nouvelle approche d'hybridation pour construire des algorithmes d'optimisation plus efficaces et plus robustes.

L'idée de cette approche est la suivante : on laisse un AE diriger le processus global de minimisation, en faisant évoluer des générations successives d'individus. Mais, lors de chaque évaluation de génération, des individus particuliers, dits maîtres, sont judicieusement sélectionnés pour y calculer le coût exact et son gradient. Ceci suppose bien sûr que l'on puisse disposer de routines de calcul du coût et du gradient, dont on se sert avec parcimonie. Ces calculs, coûteux dans le contexte habituel de la mécanique numérique, sont faits pour un faible nombre d'individus – maîtres – et servent à fabriquer une approximation qui sera utilisée pour évaluer l'ensemble des individus restants de la génération courante.

L'utilisation d'approximation de fonctions objectifs n'est évidemment pas une nouvelle idée, les surfaces de réponse ou les réseaux neuronaux, largement répandus en optimisation, en sont une bonne illustration. L'originalité de notre contribution vient du fait que l'on utilise le gradient pour enrichir l'approximation et des techniques de classification – agglomérat habituelles en traitement statistique de données, pour identifier les meilleurs individus maîtres, pour lesquels le coût exact et le gradient sont calculés.

Le fruit de la nouvelle approche hybride proposée consiste en deux algorithmes hybrides que nous appelons respectivement *Algorithme Génétique avec Métamodèle utilisant le Gradient et basé sur l'Agglomération (AG-MGA)* ou *basé sur l'interpolation de Litzka-Orkisz (AG-MGO)*.

II.2. ALGORITHME GENETIQUE AVEC METAMODELE UTILISANT LE GRADIENT ET BASE SUR L'AGGLOMERATION (AG-MGA)

Comme SE-Meta, les deux algorithmes "l'AG-MGA" et "l'AG-MGO" utilisent aussi l'organigramme d'un AE canonique. La seule modification apportée réside dans l'étape d'évaluation des individus de la population, comme montrée sur la *Figure 1.1*.



Figure 1.1. Organigramme d'un algorithme évolutionnaire avec Métamodèle

A chaque génération, au lieu d'évaluer exactement les individus de la population, l'AG-MGA et l'AG-MGO les approximent en utilisant des Méta-modèles. Ce sont ces Méta-modèles qui entraînent la différence entre ces deux algorithmes.

Dans cette section, nous expliquons d'abord le fonctionnement du Méta-modèle de l'AG-MGA, qui est basé sur une méthode d'agglomération et une interpolation discontinue d'ordre 1 donnée.

Considérons une population qui évolue suivant un AE (cette étude utilise celui nommé Pikaïa [Charbonneau 2002]). Pour faciliter la compréhension, nous allons présenter le méta-modèle à travers l'exemple d'un problème d'optimisation à 2 paramètres (μ_1, μ_2). A une génération g , supposons qu'on a une population P_g de N_{ind} individus distribués dans l'espace de recherche comme présentés sur la Figure 1.2.

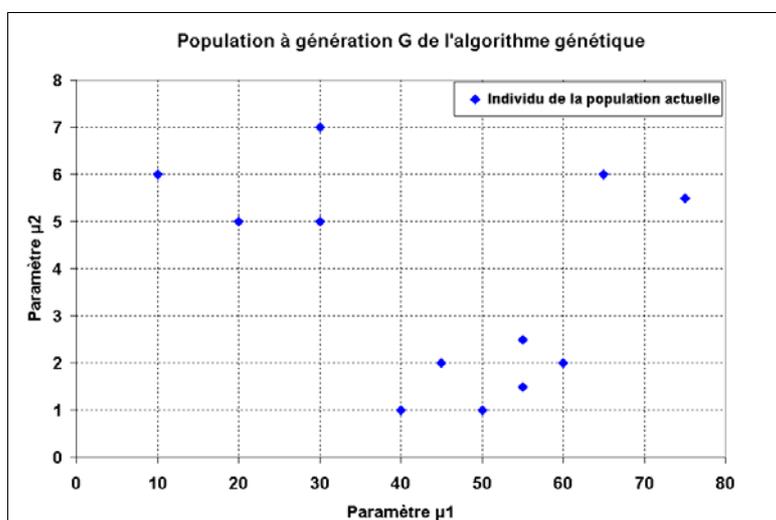


Figure 1.2. Population exemplaire dans l'espace de recherche à 2 paramètres à la génération g

Ce méta-modèle consiste en trois étapes principales :

- ↳ D'abord, il applique un algorithme pour regrouper la population courante en un nombre prédéfini $nbpeva$ de sous populations (agglomérats) contenant chacune les "voisins les plus proches entre eux". Le nombre $nbpeva$ représente le nombre d'évaluations exactes autorisées à chaque génération. Un algorithme très simple et largement répandu pour réaliser une telle partition est celui des K-moyennes :

- 1) choisir $nbpeva$ points initiaux distincts quelconques dans la population P_g comme points maîtres initiaux

$$x_M^k \in P_g \text{ pour } k = 1, \dots, nbpeva$$

- 2) initialiser les agglomérats (ou les sous populations) $C_k \subset P_g$ en associant à chaque point x_M^k les points restants de la population qui sont

plus proches (en distance euclidienne) de lui que de tous les autres points maîtres x_M^j , $j \neq k$

3) calculer les nouveaux barycentres de tous les agglomérats :

$$\text{pour } k = 1, \dots, nbpeva \quad x_{M, \text{nouveau}}^k = \frac{1}{C_k} \sum_{j=1}^{c_k} x_j$$

où c_k est le nombre d'individus appartenants à l'agglomérat C_k

4) recommencer les étapes 2 et 3 jusqu'à la stationnarité.

La *Figure 1.3* montre un exemple du regroupement de la population précédente en 3 agglomérats.

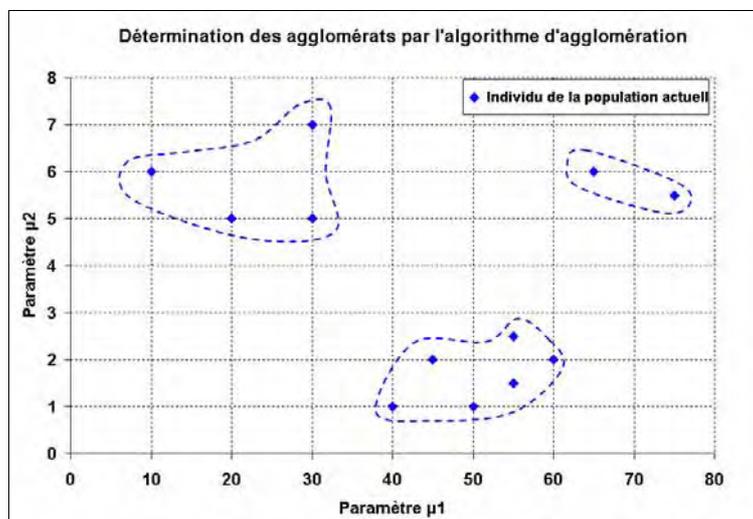


Figure 1.3. Regroupement des individus dans les agglomérats

- ↳ Ensuite, la valeur de la fonction coût Φ et de son gradient sont calculés exactement aux points maîtres, les centres de gravité de ces agglomérats (les points ronds en rose sur la *Figure 1.4*). Il faut noter que ces points ne sont pas des individus de la population considérée.
- ↳ Enfin, la valeur de tous les individus de la population est approximée par une interpolation linéaire locale obtenue (voir *Figure 1.5* pour une représentation 1D) selon :

1) Pour chaque individu i , identifier l'agglomérat C_j auquel il appartient

2) Faire l'approximation P_I d'ordre 1 en X_j suivante :

$$\tilde{\Phi}(X_i) \approx \Phi(X_j) + \nabla\Phi(X_j)(X_i - X_j) \quad (2.1)$$

Cette approximation P1 étant différente pour chaque agglomérat, elle est globalement discontinue sur l'ensemble du domaine.

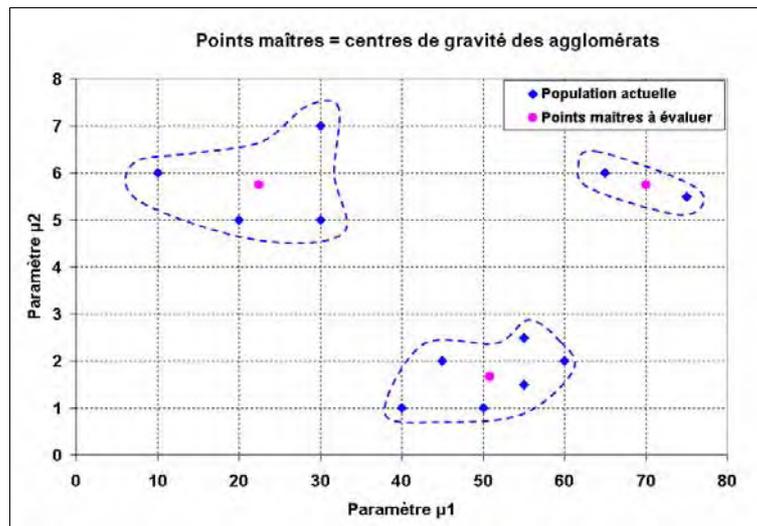


Figure 1.4. Points maîtres et leurs agglomérats

Une fois tous les individus évalués, l'AE se charge de faire évoluer la génération courante vers la suivante.

Avec ce méta-modèle, quelque soit la taille de la population, la fonction coût n'est évaluée qu'en un nombre fixe et réduit de points. Cela nous permet donc de travailler avec des populations de très grandes tailles à des coûts maîtrisés et ainsi de mieux explorer l'espace de recherche.

II.3. ALGORITHME GENETIQUE AVEC METAMODELE UTILISANT LE GRADIENT ET BASE SUR L'INTERPOLATION DE LISZKA-ORKISZ (AG-MGO)

L'algorithme AG-MGA que nous venons de décrire présente deux inconvénients: une interpolation discontinue et un méta-modèle sans mémoire. En effet,

- Il utilise une interpolation discontinue d'un agglomérat à l'autre. Ainsi, un individu situé à mi-chemin entre deux points maîtres peut prendre deux valeurs très différentes. Cependant, durant la convergence de l'AE, de tels individus ont de moins en moins de chance d'exister car les individus s'agglutinent autour des barycentres. La *Figure 1.5* présente cet inconvénient pour une interpolation en 1D : le problème est qu'il y a un saut sur l'évaluation de la fonction coût qui pourrait perturber la convergence de l'AE et donc de la méthode (en tous cas, la ralentir).

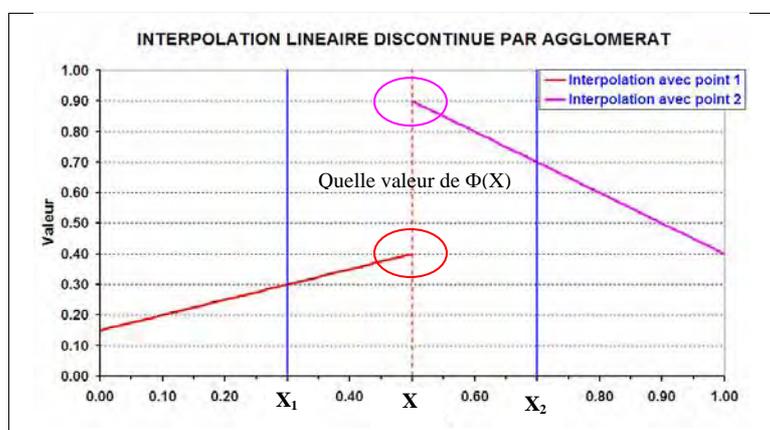


Figure 1.5. Interpolation linéaire discontinue par agglomérat : quelle valeur le point prend-t-il à mi-chemin ?

- Le second inconvénient vient du fait que l'AG-MGA ne sert pas des calculs des générations précédentes. Ces informations peuvent être précieuses pour améliorer la qualité des approximations dans le cas où la population ne déplace pas de manière importante entre chaque génération. Lorsque par exemple, d'une génération à l'autre, les agglomérats sont disjoints, les barycentres de la génération précédente (les seuls évalués exactement) ne sont plus pertinents pour une bonne évaluation P1 locale à l'intérieur des nouveaux agglomérats.

Pour pallier à ces deux défauts, nous avons introduit l'approche *Algorithme Génétique avec Métamodèle utilisant le Gradient basé sur l'interpolation de Liszka-Orkisz (AG-MGO)* avec un méta-modèle plus proche des surfaces de réponses. Au lieu d'effacer les informations des générations précédentes, il réutilise toutes les informations disponibles pour construire une approximation plus précise et plus fiable de la valeur de la fonction coût.

Supposons donc qu'à la génération courante, on dispose de N_{bpcalc} points maîtres, notés X_k^A – les anciens, dont le critère et le gradient exacts ont déjà été calculés aux générations précédentes (à la génération 0, $N_{bpcalc} = 0$).

Nous décidons ensuite du nombre N_{bpeva} d'évaluations exactes que nous nous permettons à l'étape courante. Les individus sélectionnés, selon une stratégie que nous décrirons plus loin, seront notés X_j^N – les nouveaux.

Nous allons d'abord décrire comment utiliser les $N_M = N_{bpeva} + N_{bpcalc}$ individus maîtres pour réaliser l'interpolation, puis comment choisir un placement optimal des nouveaux points X_j^N .

II.3.1. Interpolation de Liszka-Orkisz.

La méthode d'interpolation de Liszka-Orkisz est utilisée pour construire le méta-modèle. Dans cette section, on ne distingue pas les nouveaux points maîtres des anciens, car les approximations ne sont faites qu'une fois que tous les nouveaux points ont été évalués.

On possède $N_M = N_{bpcalc} + N_{bpeva}$ points maîtres en lesquels la fonction coût et son gradient sont évalués :
$$\begin{cases} \Phi(X_j)_{j=1\dots N_M} \\ \nabla\Phi(X_j)_{j=1\dots N_M} \end{cases}$$

Supposons que l'individu i de la population est approximé au point j maîtres par :

$$\Phi(X_i) \approx \Phi(X_j) + \nabla\Phi(X_j)\Delta X_{ij} \quad \forall X_j, j = 1, \dots, N_M \quad (2.2)$$

où
$$\Delta X_{ij} = X_i - X_j$$

La méthode de Liszka-Orkisz [Liszka et al. 1980] consiste en la recherche de la valeur de l'approximation $\hat{\Phi}(X_i)$ de la fonction coût en X_i , qui minimise l'erreur moyenne quadratique $E(\hat{\Phi}(X_i))$ des approximations linéaires effectuées en tous les points maîtres. Elle s'écrit :

$$E(\hat{\Phi}(X_i)) = \frac{1}{2} \sum_{j=1}^{N_M} \left(\frac{\Phi(X_i) - \Phi(X_j) - \nabla\Phi(X_j)\Delta X_{ij}}{(\Delta X_{ij})^2} \right)^2 \quad (2.3)$$

Par un calcul simple, nous trouvons que :

$$\hat{\Phi}(X_i) = \frac{\sum_{j=1}^{N_M} \frac{\Phi(X_j) + \nabla\Phi(X_j)\Delta X_{ij}}{(\Delta X_{ij})^4}}{\sum_{j=1}^{N_M} \frac{1}{(\Delta X_{ij})^4}} \quad (2.4)$$

Cette approximation est continue. Elle ressemble aux moindres carrés mobiles, si ce n'est qu'ici on considère tous les points sans utiliser une fonction de pondération qui s'annule en dehors d'un voisinage du point. On remarque toutefois le poids en $\frac{1}{(\Delta X_{ij})^4}$ qui remplace.

La Figure 1.6 en présente un exemple en 1D. On n'a plus de problème comme présenté dans la Figure 1.5. La précision augmente avec le nombre de points maîtres pris en compte, ce qui permet d'envisager un enrichissement progressif de méta-modèle au cours des générations, obtenant ainsi un algorithme à mémoire.

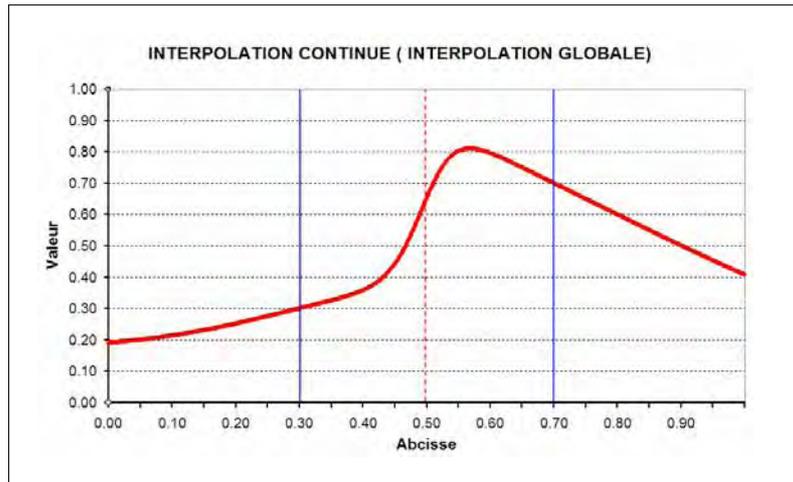


Figure 1.6. Interpolation continue basée sur la méthode de Liszka-Orkisz

Dans ce qui suit, nous nous intéressons au placement "au mieux" des nouveaux points maîtres, en tenant compte de l'existence des anciens points maîtres déjà évalués.

II.3.2. Placement optimal des nouveaux points maîtres

Durant l'évolution de la population, les N_{bpeva} nouveaux points maîtres à évaluer exactement sont sélectionnés de façon à minimiser l'erreur d'approximation en tous les individus de la population :

$$Erreur = \sum_{i=1}^{N_{bind}} (\Phi(X_i) - \hat{\Phi}(X_i))^2 \quad (2.5)$$

où N_{bind} est le nombre d'individus de la population et $\hat{\Phi}(X_i)$ est calculé par l'équation (2.4), nous avons :

$$Erreur = \sum_{i=1}^{N_{bind}} \left(\Phi(X_i) - \frac{\sum_{j=1}^{N_M} \Phi(X_j) + \nabla \Phi(X_j) \Delta X_{ij}}{\sum_{j=1}^{N_M} \frac{1}{(\Delta X_{ij})^4}} \right)^2 \quad (2.6)$$

$$\text{ou encore} \quad Erreur = \sum_{i=1}^{N_{bind}} \left(\frac{\sum_{j=1}^{N_M} \frac{\Phi(X_i) - (\Phi(X_j) + \nabla\Phi(X_j)\Delta X_{ij})}{(\Delta X_{ij})^4}}{\sum_{j=1}^{N_M} \frac{1}{(\Delta X_{ij})^4}} \right)^2$$

En supposant que le Hessien est borné et défini positif. On a :

$$\Phi(X_i) - \Phi(X_j) + \nabla\Phi(X_j)\Delta X_{ij} \cong \Delta X_{ij} \nabla^2\Phi(X_j)\Delta X_{ij} \leq C \|\Delta X_{ij}\|^2 \quad (2.7)$$

$$\text{d'où} \quad Erreur \cong \sum_{i=1}^{N_{bind}} \left(\frac{\sum_{j=1}^{N_M} \frac{\Delta X_{ij} \nabla^2\Phi(X_j)\Delta X_{ij}}{(\Delta X_{ij})^4}}{\sum_{j=1}^{N_M} \frac{1}{(\Delta X_{ij})^4}} \right)^2 \leq C \sum_{i=1}^{N_{bind}} \left(\frac{\sum_{j=1}^{N_M} \frac{1}{(\Delta X_{ij})^2}}{\sum_{j=1}^{N_M} \frac{1}{(\Delta X_{ij})^4}} \right)^2 \quad (2.8)$$

où C est la borne de $\|\nabla^2\Phi(X)\|$.

La minimisation de l'erreur d'approximation est maintenant approchée par la minimisation de la fonction suivante :

$$\Pi(X_j) = \sum_{i=1}^{N_{bind}} \left(\frac{\sum_{j=1}^{N_M} \frac{1}{(\Delta X_{ij})^2}}{\sum_{j=1}^{N_M} \frac{1}{(\Delta X_{ij})^4}} \right)^2 \quad (2.9)$$

Comme $N_M = N_{bpcalc} + N_{bpeva}$, la fonction à minimiser qui tient en compte de l'existence des points maîtres déjà évalués prend la forme:

$$\Pi(X_j) = \sum_{i=1}^{N_{bind}} \left(\frac{\sum_{k=1}^{N_{bpcalc}} \frac{1}{(\Delta X_{ik})^2} + \sum_{j=1}^{N_{bpeva}} \frac{1}{(\Delta X_{ij})^2}}{\sum_{k=1}^{N_{bpcalc}} \frac{1}{(\Delta X_{ik})^4} + \sum_{j=1}^{N_{bpeva}} \frac{1}{(\Delta X_{ij})^4}} \right)^2 \quad (2.10)$$

La minimisation de cette fonction nous amène donc à un sous problème d'optimisation qui est assez complexe à résoudre. Le temps d'une évaluation de cette fonction étant négligeable, nous utilisons un AG pour résoudre ce sous problème. Un autre AG est donc intégré dans le programme pour ce problème de minimisation. En pratique, nous utilisons l'AG de Carroll [Carroll 1997], et les nouveaux points à évaluer sont identifiés facilement. Les figures suivantes nous montrent des exemples en 2D de dispositions des nouveaux points à évaluer exactement :

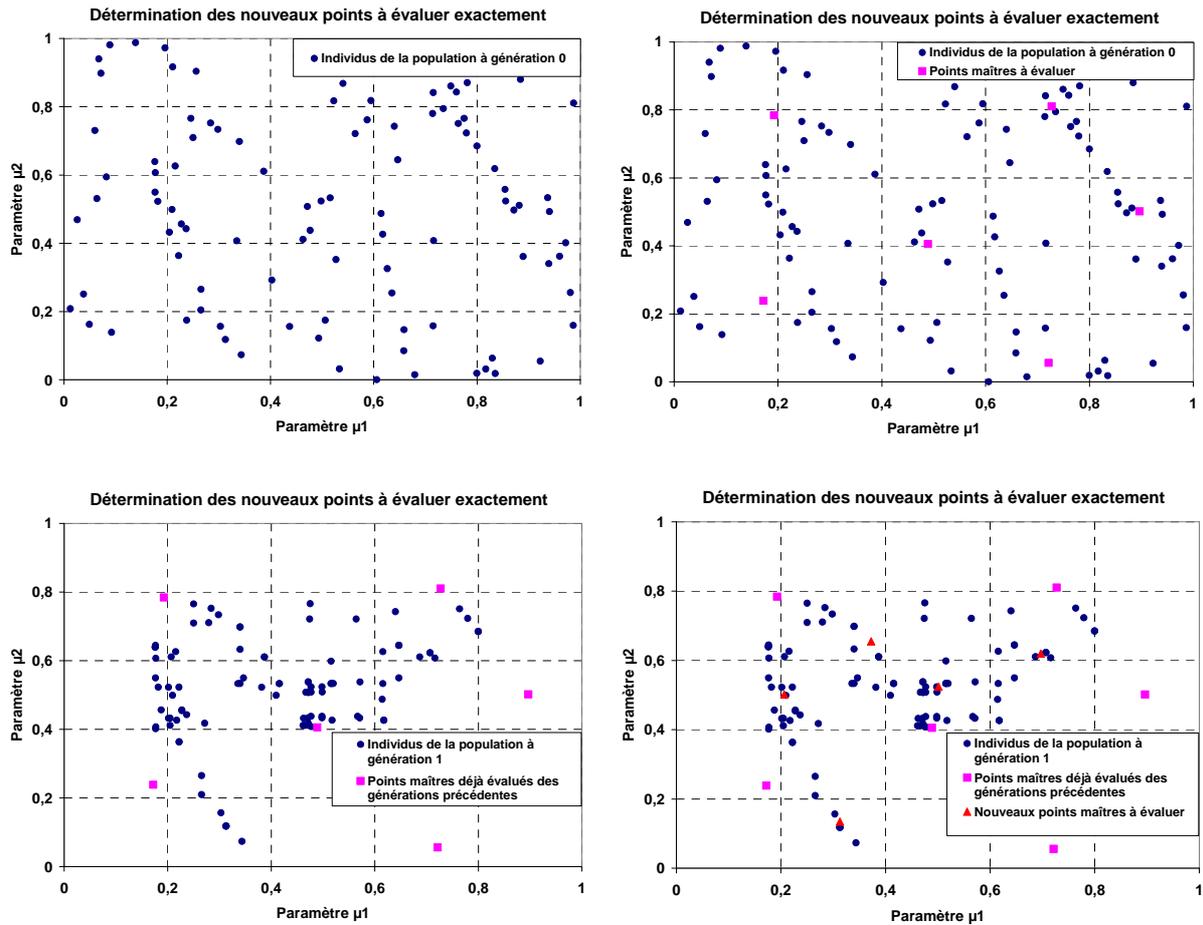


Figure 1.7. Disposition des individus, des anciens points maîtres et des nouveaux points maîtres.

II.4. CONCLUSION

Nous avons présenté, dans ce chapitre, une approche d'optimisation hybride, dont le principe est piloter l'optimisation par un algorithme global, généralement gourmand en évaluations, et de tirer profit de la disponibilité des gradients pour fournir à ce dernier des approximations très peu coûteuses de la fonction coût. La contribution originale de ce travail concerne le choix des points dits maîtres pour lesquels le coût exact et le gradient sont calculés. On obtient ainsi une méthode d'optimisation globale dont on maîtrise à l'avance le coût en calculs.

Cette méthode a été implémentée avec un algorithme génétique comme pilote, de deux façons: la première, sans mémoire, utilise une approximation discontinue. La seconde, avec mémoire des évaluations exactes précédentes, utilise l'approximation continue de Liszka-Orkisz. Nous allons donc les appliquer au forgeage.