Algorithmes hybrides pour l'optimisation globale

Application en forgeage

Abderrahmane Habbal* — Lionel Fourment — Tien Tho Do****

** Lab. Mathématiques*
Université de Nice - Sophia-Antipolis
Parc Valrose, F-06108 Nice cedex
habbal@unice.fr

*** CEMEF, Ecole des Mines de Paris*
BP 207, F-06904 Sophia-Antipolis cedex
{lionel.fourment, tientho.do}@ensmp.fr

RÉSUMÉ. *On introduit deux algorithmes hybrides d'optimisation évolutionnaire basés sur le principe d'approximation du critère en se servant d'un nombre limité d'évaluations exactes de la fonction et de son gradient. Le premier algorithme utilise un ansatz discontinu avec classification de la génération courante. Le second utilise l'interpolation de Liszka-Orkisz et garde mémoire des évaluations exactes des générations précédentes. Ces deux méthodes sont appliquées à un problème 3D d'optimisation de forme en forgeage, avec un critère combinant l'énergie totale de forgeage et une fonction de mesure de défaut. On présente des résultats numériques qui illustrent l'efficacité de ces algorithmes.*

ABSTRACT. *We introduce two evolutionnary hybrid optimizers, based on surrogate models which use a limited prescribed number of exact evaluations of the criterion and its gradient. The first algorithm uses a discontinuous ansatz with a clustering technique. The second one uses a Liszka-Orkisz interpolation scheme, and keeps memory of the exactly evaluated individuals of previous generations. These two methods are applied to a 3D forging shape optimization problem. The considered objective combines the total energy cost and a defect criterion. We present numerical results which illustrate the efficiency of the developped algorithms.*

MOTS-CLÉS : *algorithme génétique, stratégie d'évolution, classification de données, Lisza-Orkisz, forgeage, optimisation de forme, état adjoint.*

KEYWORDS: *genetic algorithm, evolution strategy, clustering, Lisza-Orkisz, forging process, shape optimization, adjoint method.*

DOI:10.3166/REMN.17.303-322

1. Introduction

Les algorithmes d'optimisation qui peuvent se passer du gradient sont attrayants, en particulier parce qu'ils donnent du sens à la formule *un solveur = un optimiseur*, qui illustre le fait qu'une fois que l'on a construit un solveur, on peut faire de l'optimisation sans autre investissement supplémentaire que de mettre en boucle le minimiseur et le solveur.

Ces algorithmes appartiennent pour l'essentiel à deux classes. La première est celle des méthodes *à la Powell* (Powell, 1964) datant des années 1960, dont on pourra trouver une revue récente par exemple dans (Conn *et al.*, 1997). Ces méthodes ne prétendent pas convenir à l'optimisation globale. La seconde classe est celle des algorithmes évolutionnaires, en particulier les algorithmes génétiques ou les stratégies d'évolution. Ces algorithmes, d'essence probabiliste, sont conçus *pour* faire de l'optimisation globale. John Holland (1962) en a posé les principes fondateurs, et David Goldberg (1989) a fortement contribué à les vulgariser et à en montrer tout l'intérêt pratique.

Il est maintenant classique de comparer cette dernière classe, les algorithmes évolutionnaires, aux méthodes habituelles de descente par le gradient. Les premières ne se laissent pas piéger dans des minima locaux, mais consomment un trés grand nombre d'évaluations de la fonction coût, alors que les secondes convergent vite, avec peu d'évaluations du coût, mais n'ont aucun moyen d'échapper aux minima locaux s'il y en a, sans compter que le développement, théorique ou informatique, lié au calcul de gradients, est souvent loin d'être trivial.

Pour tenter de concilier les avantages de ces deux approches, l'idée de cet article est la suivante : on laisse un algorithme évolutionnaire 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 trés faible nombre d'individus -maîtres- et servent à fabriquer un ansatz qui sera utilisé pour évaluer l'ensemble des individus restants de la génération courante.

L'utilisation d'ansatz de fonctions objectifs n'est évidement 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'ansatz et des techniques de classification -clustering- habituelles en traitement statistique de données, pour identifier les meilleurs individus maîtres, pour lesquels le coût exact et le gradient sont calculés.

Nous avons implémenté, selon le principe décrit ci-avant, deux approches. La première utilise une approximation locale discontinue, sans mémoire d'une génération à l'autre. La seconde utilise une approximation locale continue et avec mémoire.

Ces deux approches sont testées sur un ensemble de fonctions mathématiques, puis appliquées à un problème d'optimisation de préforme en forgeage.

Sachant que le coût de simulation moyen d'un problème de forgeage est de 24 heures de calcul, et que le nombre de simulations visées pour une optimisation doit être inférieur à la cinquantaine, nous avons là un bon argument *contre* l'utilisation d'optimiseurs trop gourmands en évaluations. De même, la présence d'extrema locaux (Fourment et Ward, 2003) est un bon argument *contre* l'utilisation d'optimiseurs locaux. Le problème de forgeage, domaine d'application représentatif de nombreux problèmes en mécanique numérique, est donc un trés bon candidat pour illustrer l'intérêt des approches hybrides en optimisation.

2. Principe et description de l'approche hybride

Nous ne détaillons pas le principe de fonctionnement des algorithmes évolutionnaires ; à titre d'indication, on pourra se référer au livre (Bäck *et al.*, 2000). Nous notons désormais *AE* un de ceux-ci.

Une variable d'optimisation s'appelle individu. Un ensemble d'individus s'appelle une population, et un *AE* fait évoluer une population initiale de génération en génération, selon des règles données, théoriquement jusqu'à stationnarité de ce processus.

Dans ce contexte, le principe de base de l'approche hybride que nous développons est sommairement le suivant :

1) on se donne une génération numéro (n), formée de N_{ind} individus,

2) on fabrique N_M individus maîtres, dont on calcule le critère exact et le gradient du critère,

3) on calcule pour les N_{ind} individus un critère approximatif, de coût de calcul beaucoup plus faible,

4) l'*AE* fait évoluer la population évaluée avec ce critère approximatif, de la génération (n) à $(n+1)$, puis retourne à l'étape 2, jusqu'à convergence.

Remarquons que les individus maîtres ne sont pas nécessairement préexistants dans la population courante, mais sont fabriqués selon un algorithme qui sera précisé pour les deux cas d'implémentation.

On se donne une fonction $J : K \subset \mathbb{R}^N \rightarrow \mathbb{R}$ que l'on cherche à minimiser sur l'ensemble $K = [a_i, b_i]^N$ (contraintes de bornes). Les contraintes autres que celles-ci sont supposées intégrées dans la formulation du critère J, éventuellement *via* une pénalisation.

On supposera que J est de classe $\mathcal{C}^2(K)$ et que son hessien est borné sur K, *i.e.* $|| < \nabla^2 J(\mathbf{u})\,\mathbf{v}, \mathbf{v} > || \leq \alpha ||\mathbf{v}||^2$ pour tout $\mathbf{u} \in K$ et tout $\mathbf{v} \in \mathbb{R}^N$ (le produit scalaire et la norme sont ici ceux euclidiens de $\mathbb{R}^N$).

On suppose qu'à une génération donnée, l'algorithme évolutionnaire utilisé (*AE*) propose une population $\mathcal{P}$ formée de N_{ind} individus : $\mathcal{P} = \{ \mathbf{x}^{(k)}, 1 \leq k \leq N_{ind} \} \subset K$, chaque individu $\mathbf{x}^{(k)} \in \mathbb{R}^N$ étant noté au besoin $\mathbf{x}^{(k)} = (x_i^{(k)})_{1 \leq i \leq N}$.

2.1. *Première implémentation : AG-MC, basée sur le clustering et sur une approximation* $P1$ *discontinue*

L'utilisateur décide du nombre N_{eval} d'évaluations *exactes* du critère J et de son gradient qu'il peut se permettre. Le principe de la méthode est alors de considérer la population $\mathcal{P}$ comme formée de N_{eval} classes, ou clusters (agglomérats). Il s'agit d'une partition de $\mathcal{P}$ en sous-ensembles contenant chacun les *voisins les plus proches entre eux*. Un algorithme trés simple et largement répandu pour réaliser une telle partition est celui des *K-moyennes* par exemple :

1) on choisit N_M points initiaux distincts quelconques,

$$\mathbf{x}_M^{(i)} \in \mathcal{P} \text{ pour } 1 \leq i \leq N_M;$$

2) on affecte à chaque $\mathbf{x}_M^{(i)}$ les points restants qui sont plus proches (en distance euclidienne) de lui que de tout autre $\mathbf{x}_M^{(j)},\ j \neq i$,

on aura ainsi initialisé les clusters $C_i \subset \mathcal{P}$,

3) on calcule les nouveaux barycentres :

$$\mathbf{x}_M^{(i)} = \frac{1}{c_i} \sum_{j=1}^{j=c_i} \mathbf{x}_j \qquad \textit{pour tout } \mathbf{x}_j \in C_i \qquad [1]$$

où C_i est le cluster numéro i, contenant c_i éléments,

4) on recommence les étapes 2 et 3 jusqu'à stationnarité,

Une fois réalisée la classification, on calcule le critère et le gradient *exacts* pour les N_M barycentres $\mathbf{x}_M^{(i)} \in C_i$, appelés individus maîtres, puis on évalue le reste de la population $\mathcal{P}$ selon l'algorithme :

1) pour chaque $\mathbf{x} \in \mathcal{P}$, identifier le cluster C_i auquel il appartient,

2) faire l'approximation $P1$ discontinue, $\tilde{J}(\mathbf{x})$ de $J(\mathbf{x})$ suivante :

$$\tilde{J}(\mathbf{x}) = J(\mathbf{x}_M^{(i)}) + \nabla J(\mathbf{x}_M^{(i)}).(\mathbf{x} - \mathbf{x}_M^{(i)}) \qquad [2]$$

Une fois tous les individus évalués, l'algorithme *AE* est ensuite en charge de faire évoluer la génération courante vers la génération suivante.

Sous l'hypothèse que le hessien, que nous avons supposé borné, varie peu au sein de chaque cluster, hypothèse somme toute assez raisonnable, on peut montrer aisèment

que parmi tous les points $\mathbf{x}^*$ de $\mathcal{P}$, potentiellement individus maîtres, celui qui minimise l'erreur d'approximation $|J(\mathbf{x}) - \tilde{J}(\mathbf{x})|$ sur C_i est le barycentre. En effet, si l'on note

$$E(\mathbf{x}^*) = \frac{1}{2} \sum_{\mathbf{x} \in C_i} |J(\mathbf{x}) - \tilde{J}(\mathbf{x})| \quad \text{avec } \tilde{J}(\mathbf{x}) = J(\mathbf{x}^*) + \nabla J(\mathbf{x}^*).(\mathbf{x} - \mathbf{x}^*),$$

alors un développement de Taylor de $J(\mathbf{x})$ à l'ordre deux montre que

$$E(\mathbf{x}^*) = \sum_{\mathbf{x} \in C_i} < \nabla^2 J(\mathbf{z})(\mathbf{x} - \mathbf{x}^*), \mathbf{x} - \mathbf{x}^* > \text{ avec } \mathbf{z} \in C_i.$$

Grâce à l'hypothèse sur le hessien, ce dernier terme est équivalent à

$$\sum_{\mathbf{x} \in C_i} < \mathbf{x} - \mathbf{x}^*, \mathbf{x} - \mathbf{x}^* >$$

dont le minimum définit précisèment le barycentre $\mathbf{x}_M^{(i)}$ de C_i.

Par ailleurs, comme l'erreur d'évaluation commise en considérant $\tilde{J}$ au lieu de J sur C_i est $J(\mathbf{x}) - \tilde{J}(\mathbf{x}) =< \nabla^2 J(\mathbf{z})(\mathbf{x} - \mathbf{x}_M^{(i)}), \mathbf{x} - \mathbf{x}_M^{(i)} >$ pour un certain $\mathbf{z} \in C_i$, on voit que, dans les zones convexes de J, lieux où vivent les minima, on a $J(\mathbf{x}) > \tilde{J}(\mathbf{x})$, ce qui donne des valeurs artificiellement basses au critère et permet ainsi d'augmenter la probabilité des individus des zones convexes de J d'être sélectionnés. Le même raisonnement permet de noter que les zones concaves, lieux où vivent les maxima de J, sont elles surévaluées, pénalisant ainsi leur sélection. L'approximation de Taylor à l'ordre un sur les clusters agit donc comme un filtre privilégiant les zones convexes de J, y concentrant les clusters au cours de l'évolution des générations.

Tests sur des fonctions mathématiques

Le premier exemple est celui de la fonction $f_1(x) = \sin(2x)\exp(2-x)$. Elle a un minimum global et plusieurs minima locaux.

La population initiale est la même pour l'*AE* (un algorithme génétique) que pour la méthode hybride, composée de 50 individus. Nous avons utilisé 6 clusters, de sorte qu'à la 3^e génération, le nombre total d'évaluations pour la méthode hybride est de 18 alors qu'il vaut 150 pour l'*AE* , voir figures 1 et 2.

L'algorithme hybride a trouvé l'extremum global alors que l'*AE* a encore 2 candidats possibles : sa population est concentrée autour de la valeur de cet extremum, alors que celle de l'*AE* est plus distribuée, et répartie autour de 2 extrema. Cet exemple met en relief les propriétés de filtrage de l'algorithme hybride.

Le second exemple est en 2D : $f_2(x,y) = \sin(x-y)\exp(2-y)$. Cette fonction a un minimum global (environ -22026) au point $x = (-3.28740, -8.0)$. Pour une comparaison consistante des deux approches, nous avons utilisé les paramètres suivants :

– pour la méthode hybride : 120 individus, 5 clusters, 6 générations (numérotées de 0 à 5), pour un total de 30 évaluations,

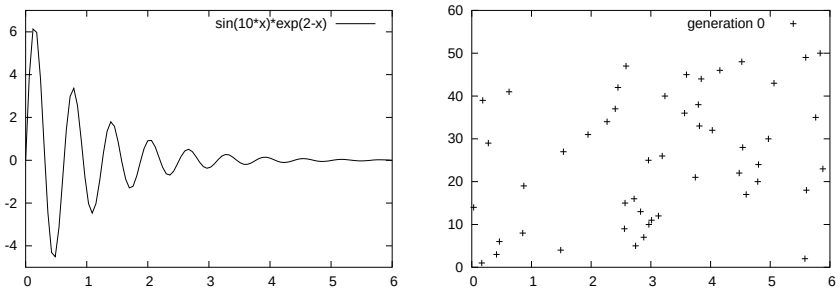


Figure 1. *a) Représentation graphique de la fonction f_1. b) Première génération de l'AE - abscisses (axe horizontal) vs. index des individus (axe vertical)*

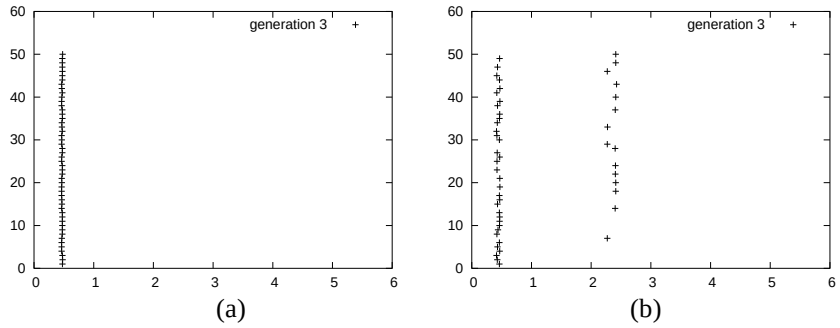


Figure 2. *Génération* 3. *a) Approche hybride. b) L'AE seul*

– pour l'*AE* : 30 individus, ce qui revient à 30 évaluations par génération.

A la génération 5, l'*AE* utilise 180 évaluations pour atteindre une approximation acceptable de la solution optimale exacte, alors que la méthode hybride utilise seulement 30 évaluations pour atteindre un résultat de qualité comparable, voir figure 4.

Le dernier exemple est la fonction $f_3(x,y) = \sin(x)\cos(y)$ de la figure 5, qui a 6 minima globaux, de valeur -1, aux points $(-3\pi/2;\pi)$, $(-3\pi/2;-\pi)$, $(-\pi/2;0)$, $(\pi/2;\pi)$, $(\pi/2;-\pi)$, et $(3\pi/2;0)$. Nous avons utilisé 3 clusters seulement.

Les résultats présentés figure 6 montrent l'avantage de l'approche hybride sur un *AE* classique. En particulier, la méthode hybride a détecté 3 minima globaux en seulement 30 évaluations là où l'*AE* n'en a détecté que 2 en 180 évaluations.

Les deux premiers exemples présentés dans cette section mettent en évidence la propriété de filtrage de l'approximation discontinue, alors que le troisième montre que, d'une part, celle-ci ne nuit pas à la détection de tous les extrema, mais que d'autre part, le choix du nombre de clusters (3 alors que l'on a 6 minima) borne le nombre de minima qui seront atteints.

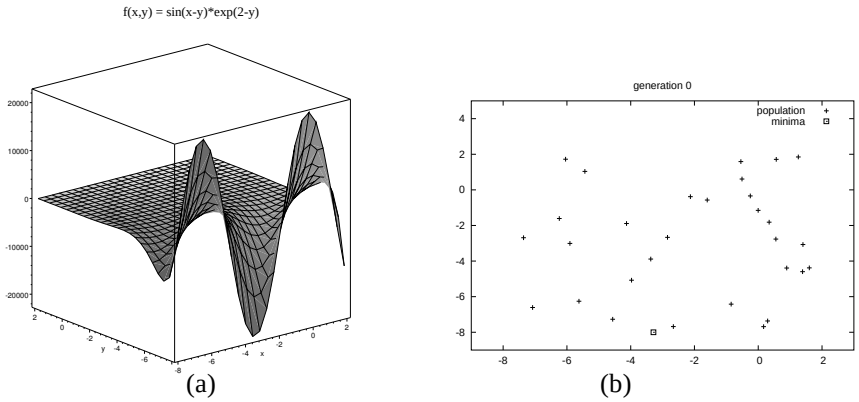


Figure 3. *a) Représentation graphique de la fonction f_2. b) Première génération de l'AE où les individus sont représentés dans l'espace paramétrique*

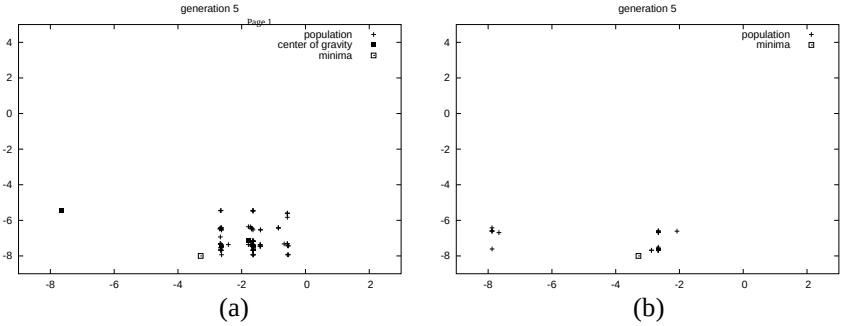
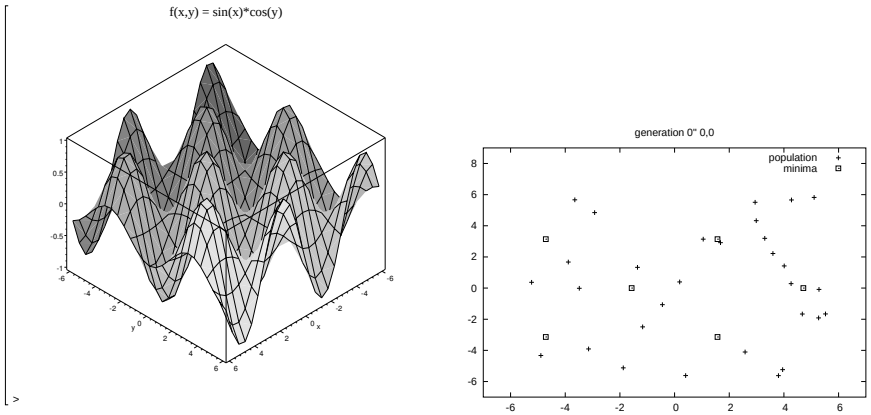
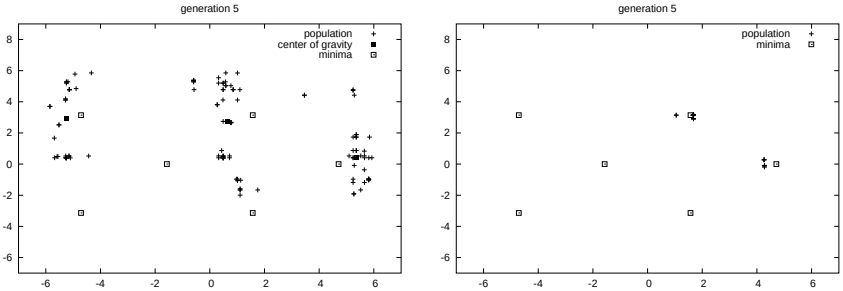


Figure 4. *Génération* 5. *a) Hybride. b) L'AE . (Individus représentés dans l'espace paramétrique)*



(a) (b)

Figure 5. *a) Représentation graphique de la fonction f_3. b) Première génération de l'AE (individus représentés dans l'espace paramétrique)*



(a) (b)

Figure 6. *Génération* 5. *a) Hybride. b) L'AE (individus représentés dans l'espace paramétrique)*

2.2. *Seconde implémentation : AG-MLO, basée sur une approximation de Liszka-Orkisz avec localisation optimale*

L'algorithme que nous venons de décrire présente deux inconvénients : discontinu et sans mémoire. En effet,

– on utilise une interpolation discontinue d'un cluster à l'autre. Ainsi, un individu situé à mi-chemin entre deux barycentres peut aléatoirement prendre deux valeurs lar-

gement différentes (voir figure 7). Cependant, durant la convergence des générations, de tels individus ont de moins en moins de chance d'exister (les individus s'agglutinent autour des barycentres),

– le second inconvénient vient de la difficulté de se servir de calculs de générations précèdentes. Lorsque par exemple, d'une génération à l'autre, les clusters sont disjoints, les barycentres de la génération précèdente (les seuls évalués exactement) ne sont plus pertinents pour une bonne évaluation $P1$ locale à l'intérieur des nouveaux clusters.

Pour pallier ces deux défauts, nous avons introduit une seconde méthode, que nous décrivons ci-dessous.

Supposons donc qu'à la génération courante, on dispose de N_A individus maîtres, notés $\mathbf{x}_A^{(i)}$ -les anciens-, dont le critère et le gradient exacts ont *déjà* été calculés précèdement (initialement, $N_A = 0$).

Nous décidons ensuite du nombre N_N d'évaluations exactes que nous nous permettons à l'étape courante. Les individus sélectionnés, selon une stratégie que nous décrivons plus loin, sont notés $\mathbf{x}_N^{(i)}$ -les nouveaux.

Nous allons d'abord décrire comment utiliser les $N_M = N_A + N_N$ individus maîtres pour réaliser l'interpolation, puis comment choisir un placement optimal des nouveaux points $\mathbf{x}_N^{(i)}$. On notera par la suite $d(\mathbf{x}, \mathbf{y})$ la distance euclidienne entre deux points $\mathbf{x}$ et $\mathbf{y}$ de $\mathbb{R}^N$.

2.2.1. *Interpolation de Liszka-Orkisz*

Dans cette section, on ne fait pas la différence entre les nouveaux individus maîtres et les anciens. On supposera donc que l'on a N_M individus, notés $\mathbf{x}_M^{(i)}$ (la lettre M valant indifférement l'une des lettres A ou N) et dont on connaît la valeur exacte du critère et du gradient de celui-ci.

La méthode de Liszka-Orkisz (Liszka T., Orkisz J., 1980) consiste en la recherche de N_{ind} scalaires (J_k), approximations du critère J en $\mathbf{x}^{(k)}$ (individus non maîtres de la génération courante), qui minimisent l'erreur moyenne quadratique suivante :

$$E(J_k) = \sum_{i=1}^{i=N_M} \left[\frac{J_k - \left(J(\mathbf{x}_M^{(i)}) + \nabla J(\mathbf{x}_M^{(i)}).(\mathbf{x}^{(k)} - \mathbf{x}_M^{(i)}) \right)}{d(\mathbf{x}^{(k)}, \mathbf{x}_M^{(i)})^2} \right]^2 \qquad [3]$$

Un calcul simple montre que :

$$J_k = \sum_{i=1}^{i=N_M} \left[\frac{J(\mathbf{x}_M^{(i)}) + \nabla J(\mathbf{x}_M^{(i)}).(\mathbf{x}^{(k)} - \mathbf{x}_M^{(i)})}{d(\mathbf{x}^{(k)}, \mathbf{x}_M^{(i)})^4} \right] \times D(\mathbf{x}^{(k)}) \qquad [4]$$

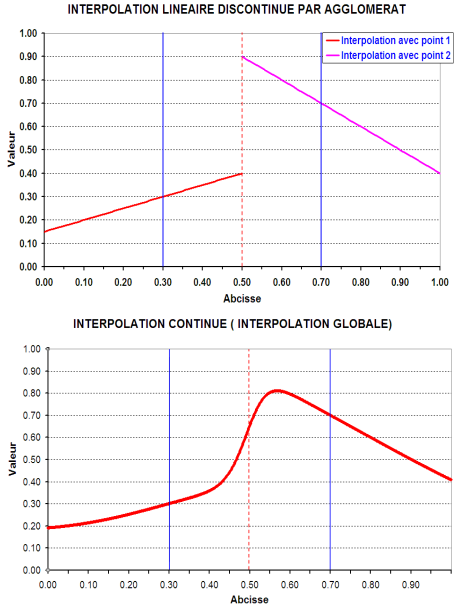


Figure 7. *Illustration du défaut de continuité de l'interpolation par clusters, que ne présente pas la méthode de Liszka-Orkisz*

avec

$$D(\mathbf{x}^{(k)}) = \frac{1}{\sum_{i=1}^{i=N_M} \frac{1}{d(\mathbf{x}^{(k)}, \mathbf{x}_M^{(i)})^4}}$$

Cette approximation, valable pour tout $\mathbf{x}^{(k)}$ non maître, est continue (en considérant le prolongement par continuité $J_k = J(\mathbf{x}_M^{(i)})$ pour les individus maîtres). La précision augmente avec le nombre de points maîtres pris en compte, ce qui permet d'envisager un enrichissement progressif au cours de l'évolution des générations, obtenant ainsi un algorithme à mémoire. Dans ce qui suit, nous nous intéressons au placement « au mieux » des individus maîtres (placement, que par abus de langage, nous qualifions d'optimal).

2.2.2. *Placement optimal des individus maîtres*

Durant l'évolution de la population $\mathcal{P}$, les nouveaux candidats $\mathbf{x}_N^{(i)}$ à être des individus maîtres sont choisis de façon à minimiser l'erreur d'approximation

$$\sum_{k=1}^{k=N_{ind}} |J(\mathbf{x}^{(k)}) - J_k|^2.$$

Or, d'après la formule *[4]*,

$$J(\mathbf{x}^{(k)}) - J_k = \sum_{i=1}^{i=N_M} \left[\frac{J(\mathbf{x}^{(k)}) - \left(J(\mathbf{x}_M^{(i)}) + \nabla J(\mathbf{x}_M^{(i)}).(\mathbf{x}^{(k)} - \mathbf{x}_M^{(i)}) \right)}{d(\mathbf{x}^{(k)}, \mathbf{x}_M^{(i)})^4} \right] \times D(\mathbf{x}^{(k)})$$

$$= \sum_{i=1}^{i=N_M} \left[\frac{< \nabla^2 J(\mathbf{z}^{(i)}).(\mathbf{x}^{(k)} - \mathbf{x}_M^{(i)}), \mathbf{x}^{(k)} - \mathbf{x}_M^{(i)} >}{d(\mathbf{x}^{(k)}, \mathbf{x}_M^{(i)})^4} \right] \times D(\mathbf{x}^{(k)}) \quad [5]$$

pour certains $\mathbf{z}^{(i)}$ dans $\mathcal{P}$.

On supposant -grossièrement- que, au voisinage des individus maîtres, le hessien est borné et a le bon signe, on peut simplifier l'expression *[5]*, et obtenir, en approximation de $\sum_{k=1}^{k=N_{ind}} |J(\mathbf{x}^{(k)}) - J_k|^2$, un critère de placement *indépendant* de la fonctionnelle J, qui s'écrit sous la forme

$$\Pi(\mathbf{x}_N^{(i)}) = \sum_{k=1}^{k=N_{ind}} \left(\frac{\sum_{i=1}^{i=N_A} \frac{1}{d(\mathbf{x}^{(k)}, \mathbf{x}_A^{(i)})^2} + \sum_{i=1}^{i=N_N} \frac{1}{d(\mathbf{x}^{(k)}, \mathbf{x}_N^{(i)})^2}}{\sum_{i=1}^{i=N_A} \frac{1}{d(\mathbf{x}^{(k)}, \mathbf{x}_A^{(i)})^4} + \sum_{i=1}^{i=N_N} \frac{1}{d(\mathbf{x}^{(k)}, \mathbf{x}_N^{(i)})^4}} \right)^2 \quad [6]$$

où, rappelons-le, les N_A anciens points maîtres sont les $\mathbf{x}_A^{(i)}$ et les N_N nouveaux, obtenus par minimisation de Π ci-dessus, sont les $\mathbf{x}_N^{(i)}$.

Par ailleurs, si l'on garde en tête que le processus évolutionnaire a pour vocation de faire converger la population $\mathcal{P}$, et surtout les individus maîtres $\mathbf{x}_M^{(i)}$, vers les minima de J (zones convexes), l'hypothèse sur le « bon comportement » du hessien n'est en fin de compte pas si grossière.

L'efficacité de la méthode hybride AG-MLO a été testée sur plusieurs fonctions mathématiques, essentiellement en 2D ou en 3D. La comparaison des deux approches AG-MC (clustering discontinu) et AG-MLO, sur ces fonctions mathématiques, a toujours donné l'avantage à cette dernière, ce qui confirme l'importance du caractère continu de l'approximation pour une bonne convergence de l'optimisation. Le lecteur pourra se référer à (Tho, 2006) pour une présentation comparative détaillée de ces deux approches sur de nombreux cas-tests académiques.

3. Une application en optimisation de préforme en forgeage

Le forgeage est un processus complexe, instationnaire et fortement non linéaire, dont la simulation complète en 3D peut se révèler trés coûteuse en temps de calcul. C'est donc un bon exemple pour tester des approches hybrides d'optimisation. Pour plus de détails sur le problème d'optimisation en forgeage, nous renvoyons le lecteur

à (Antonio et Dourado, 2002 ; Fourment *et al.*, 1998 ; Sousa et Castro, 2002, Laroussi et Fourment, 2004) pour la partie calcul de gradient.

3.1. *Equations de forgeage*

3.1.1. *Le problème continu*

Nous présentons ci-après un trés court résumé des équations modélisant le problème de forgeage. L'écoulement de matière vérifie les principes fondamentaux de la mécanique des milieux continus, le matériau obéissant à la loi de comportement Norton-Hoff, et les effets d'inertie et de gravité étant négligés. On suppose également que le matériau est rigide, incompressible et homogène.

Les équations sont les suivantes :

$$div(\sigma(\mathbf{v})) = 0, \quad div(\mathbf{v}) = 0, \quad \sigma = s - pI, \quad p = -\frac{1}{3}trace(\sigma) \qquad [7]$$

avec

$$s = 2K\left(\sqrt{3}\dot{\bar{\varepsilon}}\right)^{m-1}\dot{\varepsilon} \quad \text{et} \quad \dot{\bar{\varepsilon}} = \sqrt{\frac{1}{3}\dot{\varepsilon} : \dot{\varepsilon}} \qquad [8]$$

Ici, $\mathbf{v}$ désigne le champs de vitesse, $\dot{\varepsilon}$ le tenseur des vitesses de déformation, σ le tenseur des contraintes, p la pression hydrostatique.

Sur la surface en contact avec l'outillage, notée $\partial\Omega_c$, on impose des conditions de type Signorini et une loi de frottement de Norton :

$$\begin{cases} \left(\mathbf{v} - \mathbf{v}_{outil}\right).n \leq \frac{\delta}{\Delta t} \text{ et } \sigma_n \leq 0 \\ \left[\left(\mathbf{v} - \mathbf{v}_{outil}\right).n - \frac{\delta}{\Delta t}\right]\sigma_n = 0 \\ \tau = -\alpha K \|\Delta\mathbf{v}_t\|^{q-1}\Delta\mathbf{v}_t \end{cases} \qquad [9]$$

où $\mathbf{v}_{outil}$ est la vitesse de l'outil, δ la distance de contact, Δt le pas de temps, τ la scission de frottement, et $\Delta\mathbf{v}_t$ la vitesse relative tangentielle.

A chaque itération en temps, les équations d'équilibre quasi-statique doivent vérifier les conditions d'incompressibilité et de contact. Ces deux conditions sont traitées *via* une formulation mixte. La pression hydrostatique p est alors le multiplicateur de Lagrange de la condition d'incompressibilité, et la pression de contact normal μ_c ($\mu_c \leq 0$) est celle de la condition de contact.

La formulation faible s'écrit donc sous la forme :

$$\left\{\begin{array}{ll} \forall \mathbf{v}^\star, & \int_\Omega 2K\left(\sqrt{3}\dot{\bar{\varepsilon}}\right)^{m-1}\dot{\varepsilon}:\dot{\varepsilon}^\star dw + \int_{\partial\Omega_c}\alpha K\|\Delta\mathbf{v}_t\|^{q-1}\Delta\mathbf{v}_t.\mathbf{v}^\star ds \\ & \qquad -\int_{\partial\Omega_{pc}}\mu_c(\mathbf{v}^\star.n)ds - \int_\Omega p div(\mathbf{v}^\star)dw = 0 \\ \forall \mu_c{}^\star, & \qquad -\int_{\partial\Omega_{pc}}\mu_c{}^\star\left[\left(\mathbf{v}-\mathbf{v}_{outil}\right).n - \frac{\delta}{\Delta t}\right]ds \leq 0, \\ \forall p^\star, & \qquad -\int_\Omega p^\star div(\mathbf{v})dw = 0. \end{array}\right. \quad [10]$$

où nous avons noté $\mathbf{v}^\star$, $p^\star$ et $\mu_c{}^\star$ les fonctions tests correspondant aux champs $\mathbf{v}$, p et μ_c. Le bord $\partial\Omega_{pc}$ désigne la surface de contact potentiel.

3.1.2. *Le problème discrétisé par éléments finis*

Nous avons utilisé un élément fini de type $P1+P1$ en vitesse-pression, stabilisé par l'ajout d'une bulle, afin de satisfaire la condition d'admissibilité LBB.

Sommairement, on note

$$\mathbf{x}_h = \sum_{k=1}^{k=NN}\mathbf{X}^k N^k, \quad \mathbf{v}_h = \sum_{k=1}^{k=NN}\mathbf{V}^k N^k, \quad p_h = \sum_{k=1}^{k=NN} P^k N^k, \quad [11]$$

les approximations $P1$ de géométrie, vitesse et pression, NN étant le nombre total de nœuds du maillage. Par commodité, on note aussi : $\dot{\varepsilon}_h = \sum_{k=1}^{k=NN}\sum_{j=1}^{j=3} B^{jk}\mathbf{V_j}^k$.

A chacun des N incréments de temps du procédé instationnaire, on obtient un système d'équations discrétisées qui s'écrit (on a omis de noter l'incrément en temps) :

$$\left\{\begin{array}{l} \forall k = 1,\ldots,NN, j = 1,2,3 \\ \int_{\Omega_h} 2K\left(\sqrt{3}\dot{\bar{\varepsilon}}_h\right)^{m-1}\dot{\varepsilon}_h : B^{jk}dw_h + \int_{\partial\Omega_{ch}}\alpha K\|\Delta\mathbf{v}_{ht}\|^{q-1}\left(\Delta\mathbf{v}_{ht}.\mathbf{e}_j\right)N^k ds_h \\ +\rho_c\,\mathbf{1}_{\partial\Omega_{pch}}(k)\left[\left(\mathbf{V}^k - \mathbf{V}_{outil}\right).n^k - \frac{\delta^k}{\Delta t}\right]^+ .n_j^k S_k - \int_{\Omega_h} p_h tr(B^{jk})dw_h = 0 \\ \forall k = 1,\ldots,NN, \quad \int_{\Omega_h} N^k div(\mathbf{v}_h)dw_h = 0. \end{array}\right. \quad [12]$$

où ρ_c est le coefficient de pénalisation de la condition de contact, δ^k la distance du nœud k à l'outil, $[x]^+$ désigne la partie positive de x, $\mathbf{e}_j$ la j^e composante de la base canonique et $\mathbf{1}_{\partial\Omega_{pch}}$ la fonction caractéristique de la discrétisée de la surface de contact potentiel.

L'ajout d'une fonction bulle pour enrichir l'interpolation du champ de vitesse (interpolation P1+) permet d'introduire un terme de stabilisation dans les équations. Ce

terme additionel vient modifier la seconde équation de [12] et assure la satisfaction de la condition de LBB. Pour plus de précision, on renvoie le lecteur à (Coupez *et al.*, 2001).

Associé à un schéma d'intégration explicite en temps, on peut résumer le système d'équations non linéaires du procédé de forgeage sous la forme :

$$\forall i = 0, \ldots, N-1 \quad R_i(\mathbf{X_i}, \mathbf{V_i}, P_i) = 0 \quad \mathbf{X_{i+1}} = \mathbf{X_i} + \mathbf{V_i}\Delta t. \qquad [13]$$

3.2. *Optimisation*

Le processus de forgeage dépend naturellement de la géométrie de bon nombre de ses composantes. Parmi ces dernières, l'outil de forge ou le lopin initial (préforme) jouent un rôle particulièrement important. Dans le cas général, la géométrie de la composante que l'on cherche à optimiser est supposée décrite à l'aide d'un jeu de paramètres que nous noterons μ, vecteur qui représente par exemple les sommets du polygone de contrôle si l'on modélise la géométrie par des surfaces de Bézier.

Les équations discrètes de forgeage *[13]* s'écrivent, en soulignant la dépendance vis-à-vis des paramètres de la géométrie cible :

$$\forall i = 0, \ldots, N-1 \quad R_i(\mu; \mathbf{X_i}(\mu), \mathbf{V_i}(\mu), P_i(\mu)) = 0 \quad \mathbf{X_{i+1}}(\mu) = \mathbf{X_i}(\mu) + \mathbf{V_i}(\mu)\Delta t. \quad [14]$$

Le forgeron peut alors définir une fonction coût à minimiser, $\Phi(\mu; \mathbf{X}, \mathbf{V}, P)$, qui est généralement un indicateur de non-qualité (apparition de replis, défaut de remplissage, etc.) Lorsqu'on cherche à minimiser cette fonction avec une méthode de descente, on se heurte à la difficulté classique de calcul direct du gradient due au fait que les solutions $\mathbf{X_i}(\mu), \mathbf{V_i}(\mu), P_i(\mu)$ dépendent implicitement des paramètres géométriques μ. Cette difficulté peut être levée à l'aide de la méthode de l'état adjoint, dont nous rappelons ci-après l'application au problème de forgeage, voir (Laroussi et Fourment, 2004) pour plus de détails.

3.2.1. *Calcul du gradient par état adjoint*

Pour une géométrie μ donnée, on résout les équations discrètes de forgeage [14], qui nous permettent d'évaluer le critère Φ.

Nous posons par définition

$$\phi(\mu) = \Phi(\mu; \mathbf{X_i}(\mu), \mathbf{V_i}(\mu), P_i(\mu)) \qquad [15]$$

On se donne un ensemble de contraintes M pour les variables μ. Le problème est alors le suivant : $\min_{\mu \in M} \phi(\mu)$, pour la résolution duquel nous souhaitons disposer du gradient.

Pour cela, il est trés classique d'introduire le lagrangien Λ, défini par

$$\Lambda(\mu;\lambda;\mathbf{X},\mathbf{V},P) = \Phi(\mu;\mathbf{X},\mathbf{V},P) + \sum_{i=0}^{N-1} \lambda_i R_i(\mathbf{X_i},\mathbf{V_i},P_i). \quad [16]$$

Grâce aux équations *[14]*, on remarque que :

$$\forall\lambda,\ \Lambda(\mu;\lambda;\mathbf{X}(\mu),\mathbf{V}(\mu),P(\mu)) = \Phi(\mu;\mathbf{X}(\mu),\mathbf{V}(\mu),P(\mu)) = \phi(\mu), \quad [17]$$

d'où : $\dfrac{d\Lambda}{d\mu}(\mu;\lambda;\mathbf{X}(\mu),\mathbf{V}(\mu),P(\mu)) = \dfrac{d\phi}{d\mu}(\mu)$ qui est le gradient cherché.

L'essentiel de la méthode de l'état adjoint réside dans un choix intelligent de λ, permettant de calculer le gradient cherché, sans avoir à calculer les dérivées des solutions (appelées variables d'état) par rapport aux variables de forme μ.

La différentiation composée nous donne (en omettant de noter les arguments des fonctions)

$$\begin{aligned}\frac{d\Lambda}{d\mu} = &\left(\frac{\partial\Phi}{\partial\mu} + \sum_{i=0}^{N}\frac{\partial\Phi}{\partial\mathbf{X}_i}\frac{d\mathbf{X}_i}{d\mu} + \sum_{i=0}^{N-1}\frac{\partial\Phi}{\partial\mathbf{V}_i}\frac{d\mathbf{V}_i}{d\mu} + \sum_{i=0}^{N-1}\frac{\partial\Phi}{\partial P_i}\frac{dP_i}{d\mu}\right) \\ &+ \sum_{i=0}^{N-1}\lambda_i^{\,T}\left(\frac{\partial R_i}{\partial\mu} + \frac{\partial R_i}{\partial\mathbf{X}_i}\frac{d\mathbf{X}_i}{d\mu} + \frac{\partial R_i}{\partial\mathbf{V}_i}\frac{d\mathbf{V}_i}{d\mu} + \frac{\partial R_i}{\partial P_i}\frac{dP_i}{d\mu}\right)\end{aligned} \quad [18]$$

Introduisons les variables intermédiaires (Laroussi et Fourment, 2004)

$$\forall i = N-1,\ldots,-1 \quad \Gamma_i = \frac{\partial\Phi}{\partial\mathbf{X}_N} + \sum_{j=i+1}^{j=N-1}\left(\frac{\partial\Phi}{\partial\mathbf{X}_j} + \lambda_j^{\,T}\frac{\partial R_j}{\partial\mathbf{X}_j}\right) \quad [19]$$

et notons $W_i = \begin{pmatrix}\mathbf{V}_i\\ P_i\end{pmatrix}$ et $\overline{\Gamma}_i = \begin{pmatrix}\Gamma_i\\ 0\end{pmatrix}$, les bons candidats λ_i sont alors calculés, en sens retrograde en temps, en résolvant les systèmes linéaires suivants :

$$\forall i = N-1,\ldots,0 \quad \left(\frac{\partial R_i}{\partial W_i}\right)\lambda_i = \left(\frac{\partial\Phi}{\partial W_i} + \Delta t\overline{\Gamma}_i\right)^T \quad [20]$$

Finalement, l'expression du gradient du coût est donnée par :

$$\frac{d\phi}{d\mu}(\mu) = \frac{d\Lambda}{d\mu} = \frac{\partial\Phi}{\partial\mu} + \sum_{i=0}^{N-1}\lambda_i^{\,T}\frac{\partial R_i}{\partial\mu} + \Gamma_{-1}\frac{d\mathbf{X}_0}{d\mu}. \quad [21]$$

3.2.2. *Choix des paramètres de contrôle*

Dans la formule du gradient *[21]*, le terme $\Gamma_{-1}\frac{d\,\mathbf{X}_0}{d\mu}$ représente la sensibilité du coût vis-à-vis de la forme du lopin initial, ou préforme. Les termes $\lambda_i^{\,T}\frac{\partial R_i}{\partial \mu}$ représentent la sensibilité du coût vis-à-vis du modèle mécanique lui-même, donc en particulier vis-à-vis des conditions de contact, pour lesquelles la géométrie de l'outil de forge est déterminante.

La formule du gradient peut être relativement simplifiée, dans l'un des cas :

– optimisation de la forme de l'outil de forge seul :

$$\frac{d\,\mathbf{X}_0}{d\mu}=0,\quad \frac{\partial R_i}{\partial \mu}\neq 0.$$

– optimisation de la préforme seule :

$$\frac{d\,\mathbf{X}_0}{d\mu}\neq 0,\quad \frac{\partial R_i}{\partial \mu}=0.$$

Nous avons choisi le second cas, pour lequel la formule du gradient se réduit à : $\frac{d\phi}{d\mu}(\mu)=\frac{\partial \Phi}{\partial \mu}+\Gamma_{-1}\frac{d\,\mathbf{X}_0}{d\mu}$, ce qui nécessitera tout de même la résolution de toute la suite retrograde en incréments de temps des systèmes linéaires donnés par *[20]*.

3.2.3. *Choix des critères à minimiser*

Nous avons considéré deux types de fonctionnelles objectifs, l'énergie totale Φ_E et une mesure de défaut de repli, Φ_R. Ces critères sont donnés par :

$$\Phi_E=\int_{t_0}^{t_{fin}}\left[\int_{\Omega_t}\sigma:\dot{\varepsilon}dw+\int_{\partial\Omega_{ct}}\tau.\,\mathbf{v}ds\right]dt \qquad [22]$$

$$\Phi_R=\frac{1}{t_{fin}-t_0}\int_{t_0}^{t_{fin}}\left[\frac{1}{S_l}\int_{\partial\Omega_{lt}}\left(\frac{\dot{\bar{\varepsilon}}}{\dot{\bar{\varepsilon}}_m}\right)^{\lambda}ds\right]^{1/\lambda}dt \qquad [23]$$

où Ω_t est le domaine discrétisé au temps t, Ω_{ct} et Ω_{lt} ses surfaces respectivement en contact et libre, et $S_l=\int_{\partial\Omega_{lt}}ds$. Les termes $\dot{\bar{\varepsilon}}$ et $\dot{\bar{\varepsilon}}_m$ sont le taux de déformation équivalente et sa valeur moyenne, et λ est un coefficient d'amplification, choisi suffisament grand pour représenter la fonction maximum, tout en préservant la différentiabilité du critère (dans la pratique, λ est de l'ordre de 10).

3.3. *Résultats numériques d'optimisation*

Nous présentons dans cette section un cas d'application industrielle en optimisation de forgeage d'un engrenage. Une préforme initiale et l'engrenage forgé sont présentés figure 8.

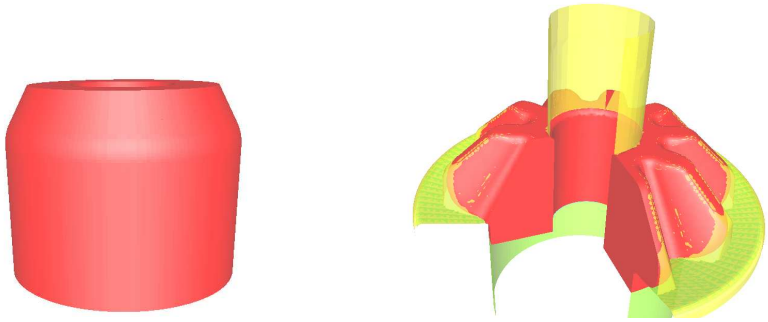


Figure 8. *Préforme initiale et pièce forgée*

La préforme recherchée étant axisymetrique, on se contente de la paramétrisation de la surface de révolution, plus précisèment de son bord comme montré en figure 9. Les calculs de forgeage sont faits en résolvant le problème 3D pour une géométrie représentant le $1/10$ de l'engrenage.

La figure 9 montre également les directions autorisées pour les paramètres. Par ailleurs, la contrainte de volume constant permet d'exprimer un des paramètres en fonction des autres, ce qui évite de gérer cette contrainte au niveau de l'optimiseur, ne gardant plus que les contraintes de bornes du type $a_i \leq \mu_i \leq b_i$.

Nous avons testé plusieurs algorithmes d'optimisation : un algorithme de descente de type BFGS, un de type stratégies d'évolution avec métamodèle SE-Meta, et les deux méthodes présentées dans cet article, utilisant toutes les deux un algorithme génétique pour piloter l'optimisation, et dont l'implémentation AG-MC utilise une interpolation $P1$ discontinue, alors que AG-MLO utilise l'interpolation continue de Liszka-Orkisz.

La fonction coût qui a été optimisée est la moyenne des deux critères d'énergie Φ_E et de défaut de repli Φ_R :

$$\Phi = \frac{1}{2}\frac{\Phi_E}{\Phi_{Eopt}} + \frac{1}{2}\frac{\Phi_R}{\Phi_{Ropt}},$$

la valeur optimale Φ_{Eopt} et Φ_{Ropt} de chaque critère ayant été calculée séparèment. Ce problème a bien plusieurs extrema locaux, la fonction coût combinant des critères antagonistes.

Les formes optimales obtenues sont présentées figure 10, et les résultats d'optimisation résumés dans le tableau 1.

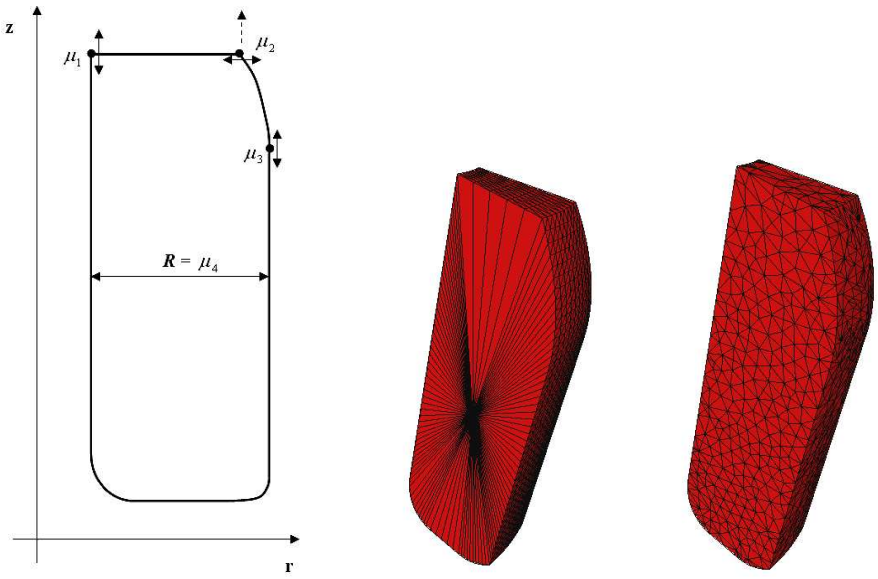


Figure 9. *Parametrisation de la préforme et construction du domaine de calcul 3D.*

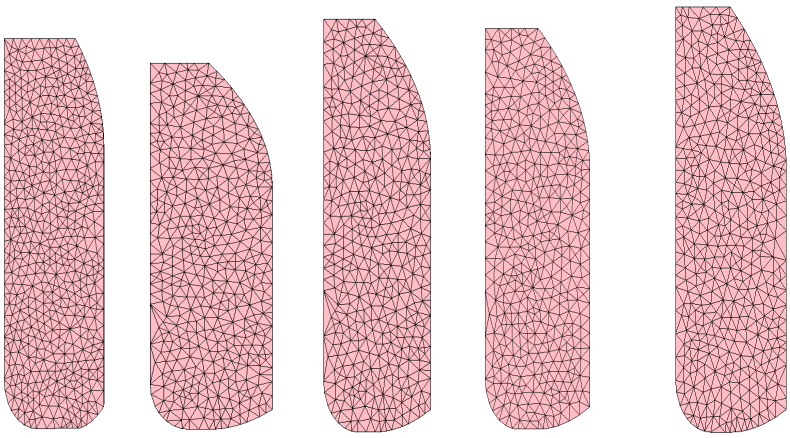


Figure 10. *De gauche à droite : forme de réfèrence proposée par le forgeron d'Asco-forge, puis les formes optimales obtenues par BFGS, SE-Meta, AG-MC et AG-MLO*

Sur ce cas complexe, notons la « convergence » des algorithmes évolutionnaires, paramétrés pour faire 6 calculs exacts par génération, pendant 8 générations, soit un

coût total de 48 évaluations exactes. Par contre, l'algorithme BFGS stagne au bout d'une dizaine d'itérations.

Le tableau 1 comparatif donne un léger avantage à AG-MLO en ce qui concerne le gain par rapport au critère initial. Le gain en coût de calcul est beaucoup plus intéressant à souligner ; on remarque en effet que l'on divise par deux ce coût par rapport à ce que coûte l'algorithme SE-Meta.

	REF	BFGS	SE-M	AG-MC	AG-MLO
CRITÈRE	1.19	1.15	1.08	1.075	1.064
GAIN	0 %	3 %	9 %	9.3 %	10.3 %
COÛT %SE-Meta	-	(stagne)	1	0.76	0.52

Tableau 1. *Tableau comparatif des résultats d'optimisation. Le coût %SE-Meta exprime le coût de calcul nécessaire pour obtenir une solution semblable à celle de SE-Meta, prenant en compte que le calcul des gradients nécessite environ 30 % du coût de calcul du critère*

4. Conclusion

Nous avons présenté dans cette étude une approche d'optimisation hybride, dont le principe est de faire piloter l'optimisation par un algorithme global, généralement gourmand en évaluations, et de tirer profit de la disponibilité de gradients, pour fournir à ce dernier des approximations trés 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 précèdentes, utilise l'approximation continue de Liszka-Orkisz. Ces deux algorithmes se sont révélés efficaces et robustes sur des fonctions tests mathématiques, puis nous les avons appliqués en forgeage.

Notons que pour réduire le temps de calcul, la loi de comportement a été simplifiée en une loi linéaire newtonienne. Nous avons pu constater que la solution optimale du problème newtonien n'est en général pas celle du problème visco-plastique réel, tant en ce qui concerne les efforts que pour l'apparition de replis. Cette approche démonstrative est une première étape vers l'étude de modèles plus réalistes mais également plus complexes et chers en évaluations. En effet, sur un problème aussi complexe que le forgeage, et dans des situations concrètes, l'approche d'optimisation hybride permet de réduire significativement le coût de calcul ou augmente sensiblement, à coût fixé, la qualité des résultats obtenus, rendant ainsi possible l'optimisation 3D en forgeage à coût raisonnable.

5. Bibliographie

Antonio C. A., Dourado N.M., « Metal forming process optimization by inverse evolutionnary search », *J. of Materials Processing technology*, vol. 121, p. 403-413, 2002.

Bäck T., Fogel D.B., Michalewicz Z., *Evolutionary computation 1. Basic algorithms and operators*, vol. XXXVIII, IoP, Institute of Physics Publishing, Bristol, 2000.

Berard Y., Désidéri J.A., Habbal A., Janka A., Oulladji L., « Experiments with Hybridized Genetic Algorithms in Aerodynamics », *Evolutionnary Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems*, ECCOMAS, Barcelona, Spain, September, 15-17, 2003.

Conn A.R., Scheinberg K., Toint Ph.L., « Recent progress in unconstrained nonlinear optimization without derivatives », *Math. Program.*, vol. 79, p. 397-414, 1997.

Coupez T., Mocellin K., Fourment L., Chenot J.-L., « Toward large scale F.E. computation of hot forging process using iterative solvers, parallel computation and multigrid algorithms », *International Journal for Numerical Methods in Engineering*, vol. 52, p. 473-488, 2001.

Emmerich M., Giotis A., Özdemir M., Bäck T., Giannakoglou K., « Metamodel assisted evolution strategies », *in* , J. M. G. *et al.* (ed.), *Parallel Problem Solving from Nature VII*, Int'l Conf. Granada, September, 2002.

Fourment L., Vieilledent D., Chenot J.-L., « Shape optimization of the axisymmetric preform tools in forging using a direct differentiation method », *Int. J. of Forming Processes*, vol. 1, n° 4, p. 399-423, 1998.

Fourment L., Ward M., « Shape optimization for preform tool design in reverse Superplastic Forming », *VII International Conference on Computational Plasticity, COMPLAS*, Barcelonna, April 7-10, 2003.

Goldberg D.E., *Genetic algorithms in search, optimization and machine learning*, vol. XIII, Addison-Wesley, Reading, MA, 1989.

Holland J.H., « Outline for a logical theory of adaptive systems », *J. Assoc. Comput. Mach.*, vol. 9, p. 297-314, 1962.

Laroussi M., Fourment L., « The adjoint state method for sensitivity analysis of non-steady problems. Application to 3D forging », *Int. J. of Forming Processes*, vol. 7, n° 1-2, p. 35-64, 2004.

Liszka T., Orkisz J., « The finite difference method at arbitrary irregular grids and its application in applied mechanics », *Comp. and Structures*, vol. 11, p. 83-95, 1980.

Powell M.J.D., « An efficient method for finding the minimum of a function of several variables without calculating derivatives », *Comput. J.*, vol. 7, p. 155-162, 1964.

Sousa L.C., Castro C.F., « Inverse methods in design of industrial forging processes », *J. of Materials Processing technology*, vol. 128, p. 266-273, 2002.

Tho D., Optimisation de forme en forgeage 3D, PhD thesis, Ecole des Mines de Paris, Juillet, 2006.