BENCHMARK : OPTIMISATION DE LA PREFORME POUR LE FORGEAGE D'UNE ROUE DENTEE

IV.1. Introduction

Un exemple typique de forgeage est de fabriquer des engrenages (*Figure 4.1*) qui sont souvent utilisés dans les moteurs, les voitures, les avions, etc. Ils permettent le transfert de mouvements et de forces importantes d'une partie à l'autre. Ce sont donc des pièces à très haute résistance mécanique.

Figure 4.1. Exemple d'engrenages

Le forgeage des engrenages est un domaine très compétitif car ce sont souvent des pièces de grande production forgées à froid ou à mi-chaud le plus souvent et donc :

- la durée de vie des outils de forgeage est souvent courte car ils sont intensivement sollicités, alors que leur fabrication est très coûteuse.

- l'énergie nécessaire pour le forgeage d'une telle pièce est importante.

Par conséquence, prolonger la durée de vie des outils et minimiser l'énergie de mise en forme reste un enjeu important, qui peut être formulé en problème d'optimisation et en utilisant des logiciels de simulation numérique.

Dans ce chapitre, nous considérons le cas de forgeage d'une roue dentée droite proposé par la société ASCOFORGE et qui est retenu comme problème caractéristique d'optimisation de forme en forgeage, c'est-à-dire comme benchmark d'optimisation.

IV.2. Description du cas d'optimisation

IV.2.1. Présentation du cas de forgeage d'un triaxe

Ce cas consiste à forger une roue dentée de dix dents à mi-chaud à partir d'un lopin initial de forme axisymétrique simple. La *Figure 4.2* présente la forme de la pièce initiale (*Figure 4.2a*) et d'une pièce forgée finale (*Figure 4.2b*).

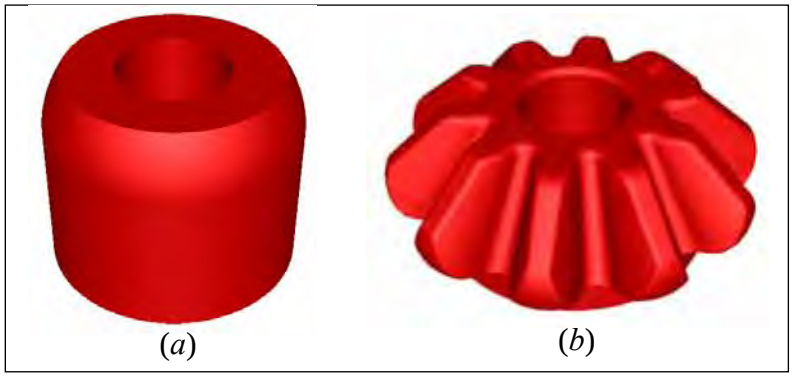

Figure 4.2. Lopin initial (a) et roue dentée (b)

L'outillage de forgeage est présenté sur la *Figure 4.3a* et la *Figure 4.3b*. Le lopin est écrasé par la matrice et l'éjecteur supérieurs. Dans son mouvement de translation verticale, la matrice supérieure entraîne la matrice inférieure. De même, l'éjecteur supérieur entraîne avec lui la tige centrale. Le dernier outil, la douille éjectrice, est fixe et ne sert qu'à caler le lopin.

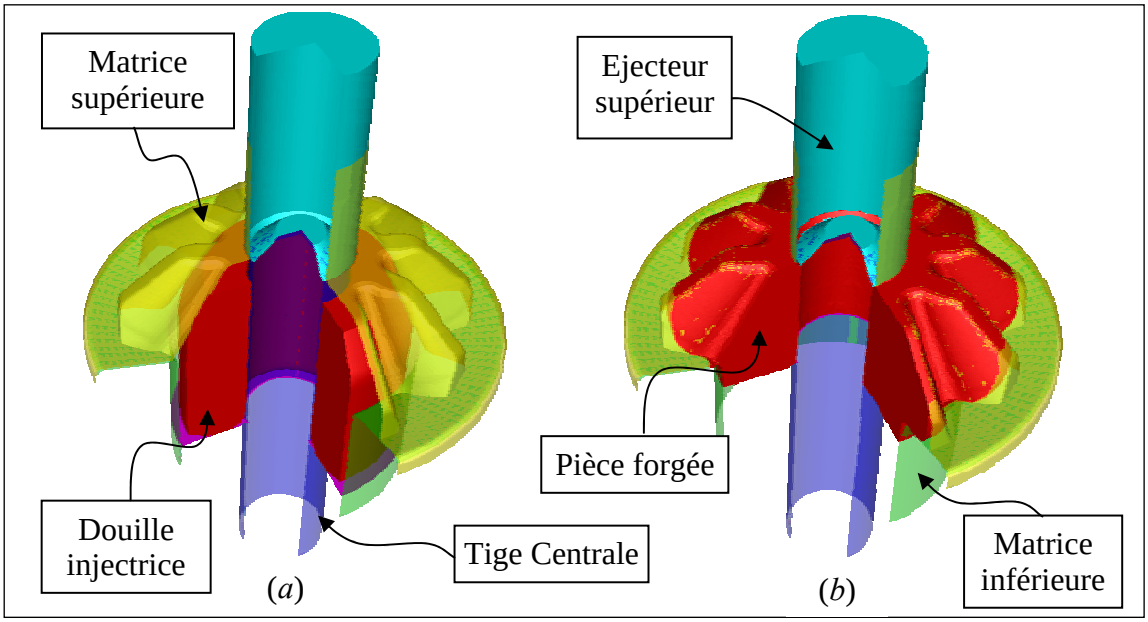

Figure 4.3. Pièces initiale (a) et finale (b) avec les outils de forgeage

La hauteur du lopin initial est de 45 *mm* et après écrasement elle est de 21 *mm*, ce qui représente une déformation de quasiment 50%.

La symétrie de la roue dentée permet de modéliser seulement un vingtième de la pièce (comme présenté sur la *Figure 4.4*) correspondant à une demi dent.

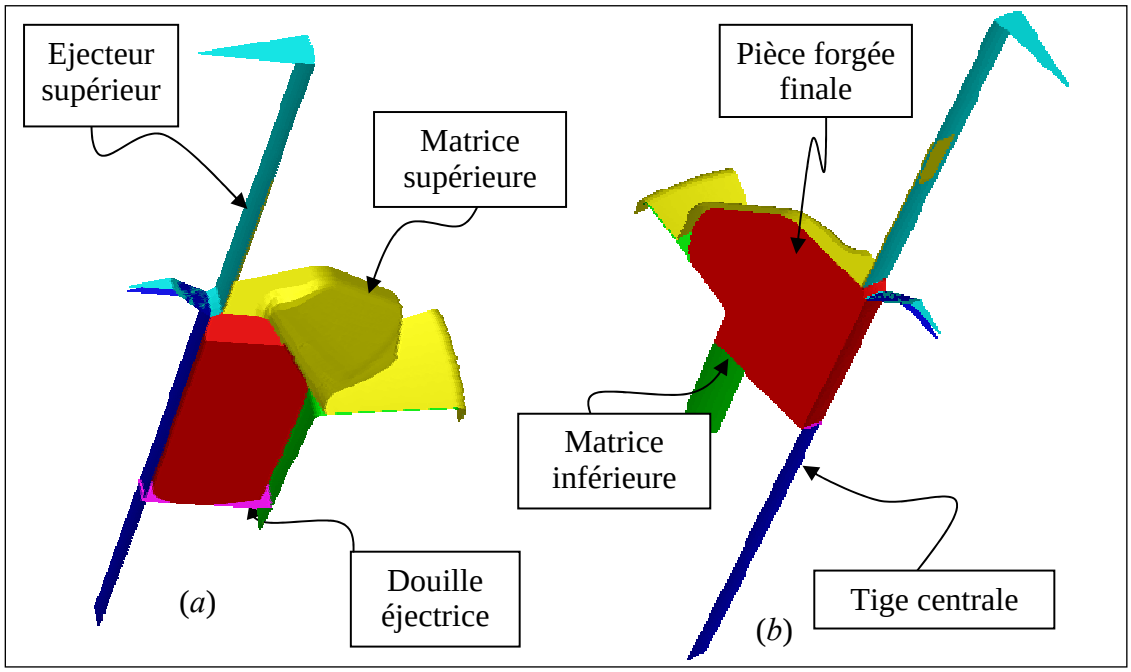

Figure 4.4.Un vingtième de la préforme à forger (a) et une demi dent (b) de la roue dentée forgée avec les outils de forgeage.

IV.2.2. Paramétrisation de la préforme

Lors de la paramétrisation de la préforme, une difficulté réside dans la prise en compte de la contrainte du volume constant, quelque soit le jeu de paramètres.

M. Laroussi [*Laroussi 2003*] a donc proposé d'utiliser une paramétrisation plus simple, de type polynomial. Le contour du plan radial est un polygone fermé (*Figure 4.5a*) composé de :

- sept portions (parties polynomiales) distinctes
  - quatre portions I, II, IV et VI sont des segments droits.
  - trois portions III, V et VII sont des polynômes de degré 2.

  Ces portions doivent satisfaire les conditions de raccordement (le contour doit être lisse) [*Laroussi 2003*].

- sept points de contrôle de coordonnées $(r_i, z_i)_{i=1,\ldots,7}$ dans le repère radial (r, z). Les mouvements de ces points sont les paramètres à optimiser. Les points retenus pour l'optimisation sont les points 2, 3 et 4. Leurs déplacements en constituent les paramètres (*Figure 4.5b*), de la manière suivante:
  - le paramètre μ_1 est le déplacement vertical du point 2 (dans la direction z).

- le paramètre μ_2 est le déplacement horizontal du point 3 (dans la direction r). Notons que dans la direction z, le déplacement du point 3 est lié à celui du point 2 de telle sorte que la portion II reste un segment parfaitement horizontal.
- le paramètre μ_3 est le déplacement vertical du point 4 (dans la direction z).
- le paramètre μ_4 est le déplacement horizontal des points 4 et 5 (identiques) dans la direction r (la portion IV doit être toujours un segment vertical).

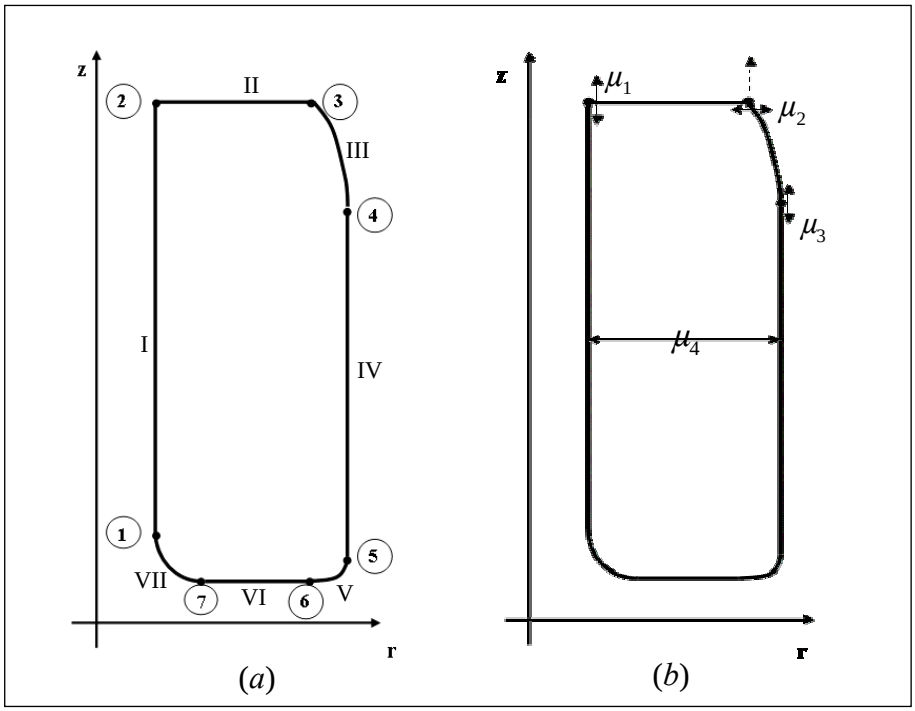

Figure 4.5. Paramétrisation 2D de la préforme axisymétrique de la roue dentée

Avec cette paramétrisation, 4 paramètres caractérisent la préforme. Le problème d'optimisation est celui de minimisation d'une fonction coût Φ (effort de forgeage, énergie totale de mise en forme, défaut de repli…) sous la contrainte de volume constant :

$$\begin{cases} Min \ \ \Phi(\mu_1,\mu_2,\mu_3,\mu_4) \\ \vartheta(\mu_1,\mu_2,\mu_3,\mu_4) = \vartheta_0 = \text{constant} \end{cases} \qquad (1)$$

où ϑ_0 est le volume initial.

La modification des paramètres de forme $(\mu_1,\mu_2,\mu_3,\mu_4)$ entraîne inévitablement un changement du volume de la préforme qui doit pourtant rester constant.

Il existe plusieurs méthodes pour traiter cette contrainte mais la méthode la plus simple est de l'éliminer en écrivant une condition de liaison sur un des paramètres. Dans notre cas, le paramètre qui permet le plus naturellement d'éliminer cette contrainte est μ_4. Le problème (1) peut donc être reformulé de la manière suivante :

$$\begin{cases} Min \;\; \Phi(\mu_1,\mu_2,\mu_3) \\ \mu_4 = f(\mu_1,\mu_2,\mu_3) \end{cases} \tag{2}$$

où l'expression de *f* est déterminée par la relation équivalente $\vartheta(\mu_1,\mu_2,\mu_3,\mu_4)=\vartheta_0$. Les détails de cette expression sont présentés dans [*Laroussi 2003*]. Le problème d'optimisation est ainsi un problème à trois paramètres (μ_1,μ_2,μ_3). On peut aussi lui ajouter les conditions de bords appliquées aux paramètres de conception.

$$Min \quad \Phi(\mu_1,\mu_2,\mu_3) \tag{3}$$

$$\text{avec} \quad \begin{cases} 40 \le \mu_1 \le 46 \\ 18 \le \mu_2 \le 22 \\ 28 \le \mu_3 \le 34 \end{cases}$$

IV.2.3. Génération automatique du maillage de la préforme

Le processus de génération des maillages de préforme 3D à partir du contour 2D est présenté dans la *Figure 4.6*.

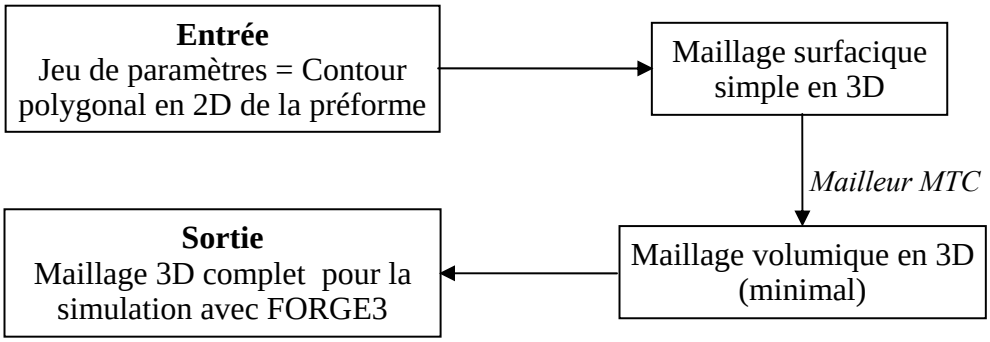

Figure 4.6. Module de génération automatique du maillage de la préforme

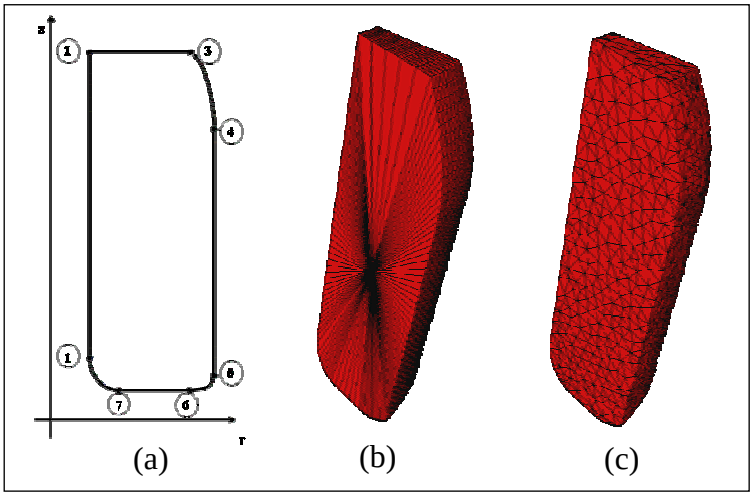

Figure 4.7. Contour polynomial 2D (a) – Maillage surfacique simple 3D (b) – Maillage surfacique et volumique (c)

A partir du jeu de paramètres, un contour polygonal 2D fermé (*Figure 4.7a*) est obtenu. Ensuite, on génère un maillage surfacique simple en 3D en extrudant ce contour dans la direction axisymétrique suivant un angle $\theta = 18°$ (correspondant à un vingtième du lopin) et en créant la connectivité entre les nœuds (*Figure 4.7b*). Le mailleur MTC permet alors d'obtenir le maillage volumique de la préforme 3D. Il manque encore des informations (comme la topologie de la frontière, les plans de symétrie, etc.) qui font l'objet de la dernière étape (*Figure 4.7c*).

IV.2.4. Fonctions coûts et benchmark d'optimisation

Du point de vue industriel, on cherche à optimiser la géométrie de la préforme pour augmenter la durée de vie des outils de forgeage. Une approche de ce problème consiste à minimiser les efforts subis par les outils durant tout le forgeage, tout en évitant la formation de défauts majeurs de type repli de matière. Cela nous conduit à formuler le benchmark suivant : minimiser la fonction coût "énergie totale" de forgeage (équation 3.31) ou/et la fonction coût " repli" (équation 3.34) vis-à-vis des trois paramètres de forme.

Pour réduire le temps de calcul et faciliter la mise en place des méthodes d'optimisation, nous décidons de simplifier le comportement du matériau et de le supposer linéaire. Ainsi, sur un PC Pentium IV (2.4Ghz – 512Mo RAM) la durée d'une simulation complète sans calcul du gradient est de 45 minutes, et est de 60 minutes avec calcul du gradient.

Enfin, nous limitons le nombre d'évaluations de la fonction coût à 50.

Nous obtenons ainsi un problème caractéristique du forgeage 3D que nous utilisons pour comparer les différents algorithmes d'optimisation entre eux.

IV.2.5. Processus d'optimisation automatique

L'algorithme d'optimisation est présenté sur la *Figure 4.8*.

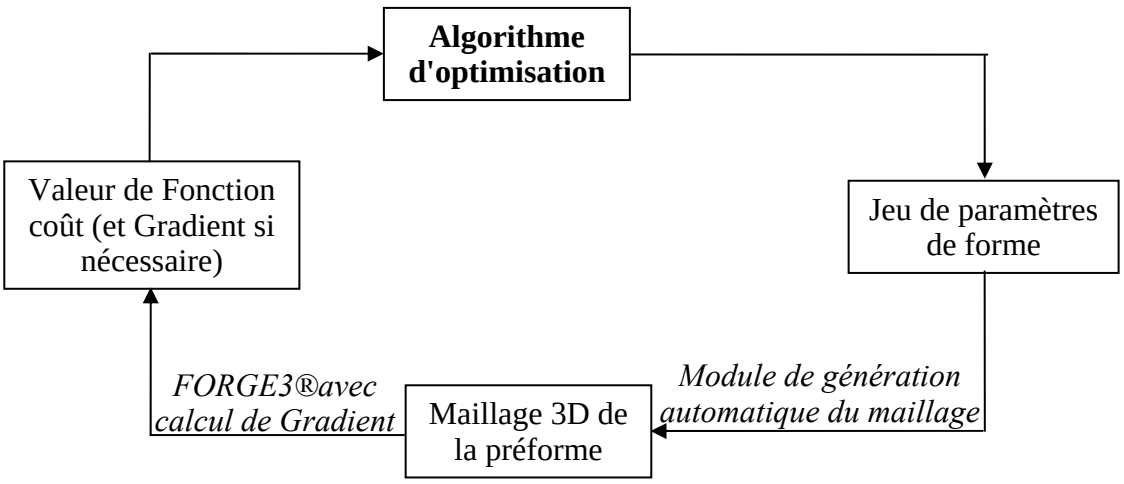

Figure 4.8. Processus d'optimisation automatique pour résoudre le problème de benchmark

IV.3. Résultats d'optimisations

Cette section présente les résultats d'optimisation obtenus. La préforme utilisée par ASCOFORGE correspond au jeu de paramètres suivant ($\mu_1 = 44,60mm$; $\mu_2 = 21,65mm$; $\mu_3 = 32,33mm$). Elle est prise comme référence. Notons que cette préforme est le fruit de l'expérience et non d'une optimisation particulière. Par ailleurs, nous avons pu constater que les géométries optimisées avec un comportement linéaire ou viscoplastique sont assez différentes. Cette préforme n'a donc pas de qualités particulières, mais elle n'est pas non plus arbitraire. C'est donc une simple référence.

Les préformes obtenues par différents algorithmes d'optimisation sont comparées à celle de référence et entre elles, en termes d'amélioration de la valeur de fonction coût, des géométries de la préforme et du coût de calcul.

IV.3.1. Optimisation par rapport à un seul critère

IV.3.1.1. Objectif 1 : Energie totale du forgeage

La fonction coût est l'énergie totale de forgeage. Elle est exprimée en valeur relative. Elle est adimensionnalisée par la valeur obtenue avec la préforme de référence.

IV.3.1.1.1. Résultats d'optimisation avec BFGS

Ce benchmark est d'abord étudié avec l'algorithme de gradient BFGS. Avec un pas de descente de *1,2mm* et un rayon de recherche linéaire de *0,6mm*, nous partons d'un point choisi aléatoirement dans l'espace correspondant au jeu de paramètres (μ_1*=40,0 ;* μ_2*=18,7 ;* μ_3*=34,0*).

L'évolution de la fonction coût et celle des paramètres sont présentées sur la *Figure 4.9.* La préforme optimisée par BFGS et la préforme de référence sont présentées sur la *Figure 4.10*.

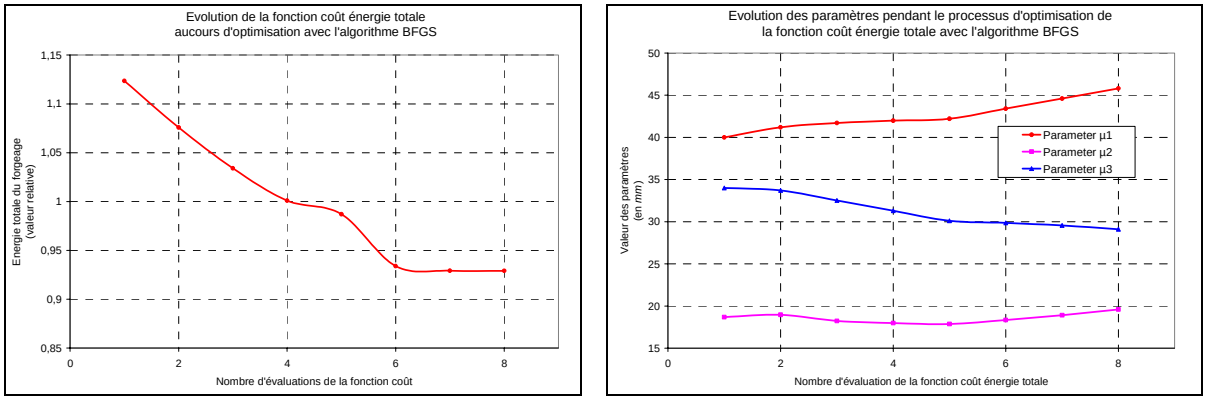

Figure 4.9. Evolution des paramètres et de la fonction coût énergie totale au cours du processus d'optimisation avec l'algorithme de gradient BFGS.

Avec cet algorithme, on obtient une amélioration de 7,1% de la valeur de la fonction coût. L'algorithme s'arrête après seulement 8 calculs de forgeage lorsque le critère de convergence sur l'amélioration relative de la fonction coût est obtenu.

Si on regarde l'évolution des paramètres, on observe qu'ils varient de manières très différentes. Le paramètre qui représente la hauteur du lopin, μ_1 augmente sans cesse. Il balaie tout l'intervalle de recherche. Au contraire, le paramètre μ_3 a tendance à décroître continûment tandis que le paramètre μ_2 fluctue légèrement. Les deux paramètres μ_1 et μ_3 prennent des valeurs très différentes par rapport à celles du point de départ alors que le paramètre μ_2 ne change pas beaucoup. Ainsi, la préforme optimisée est bien différente (plus haute et plus épaisse) de la préforme initiale, comme le montre la *Figure 4.10*.

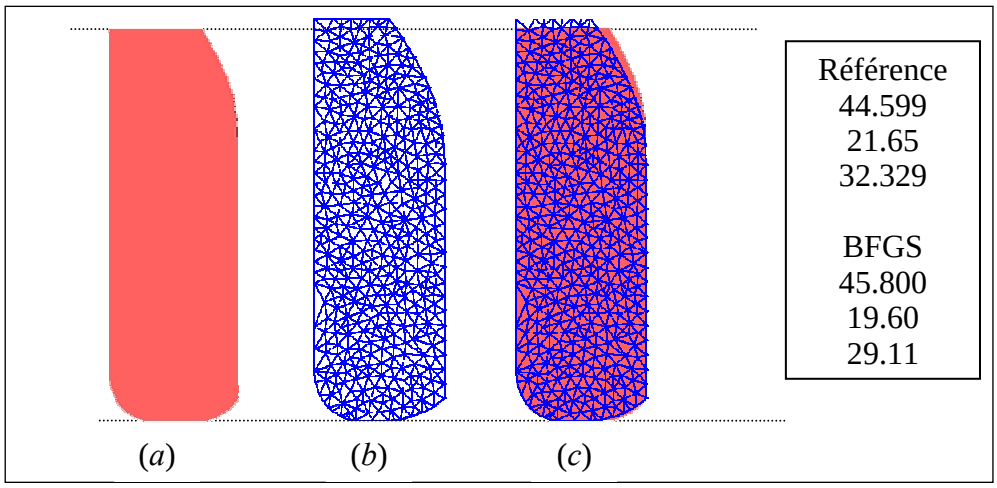

Figure 4.10. Préforme de référence (a) – Préforme optimisée obtenue avec l'algorithme de gradient BFGS (b) – Comparaison entre les deux préformes (c).

IV.3.1.1.2. Résultats d'optimisation avec SE-Meta

La Stratégie d'Evolution avec Meta modèle (SE-Meta) présenté dans la section I.5.2 est maintenant appliquée à ce benchmark. La stratégie (2+10) (population de 10 individus dont 2 parents) est retenue avec un taux de rejet de 80% de sorte que l'on évalue exactement que deux individus à chaque génération.

Après une initialisation aléatoire pour créer la base de données initiale (avec 2**nbparam* = 6 points), la SE-Meta commence la recherche de la solution optimale. Les résultats d'optimisation sont présentés sur la *Figure 4.11*.

La SE-Meta démontre son caractère « stochastique » dans la recherche de la solution optimale. La fonction coût fluctue fortement (*Figure 4.11*). Au cours du processus d'optimisation, la SE-Meta balaie tout l'espace de recherche. Grâce au métamodèle, elle choisit uniquement les solutions ayant le meilleur potentiel pour le calcul exact. Si on observe l'évolution des paramètres (*Figure 4.11*), on constate que tous balaient leur intervalle de recherche. A la fin du processus d'optimisation, la fonction coût et les paramètres ne sont pas stabilisés : la convergence de l'AE n'est pas atteinte au bout de 50 calculs, ce qui est une valeur faible pour atteindre la convergence d'une telle stratégie d'évolution.

La meilleure solution proposée par la SE-Meta est trouvée à la 27^{ième} évaluation et correspond au jeu de paramètres (μ_1*=46,0 ;* μ_2*=18,0 ;* μ_3*=31,72*). Elle apporte une amélioration de 10,4%. La préforme correspondante est assez différente de celle de référence (plus haute et plus épaisse) et ressemble bien celle trouvée par BFGS (voir *Figure 4.10* et *Figure 4.18*).

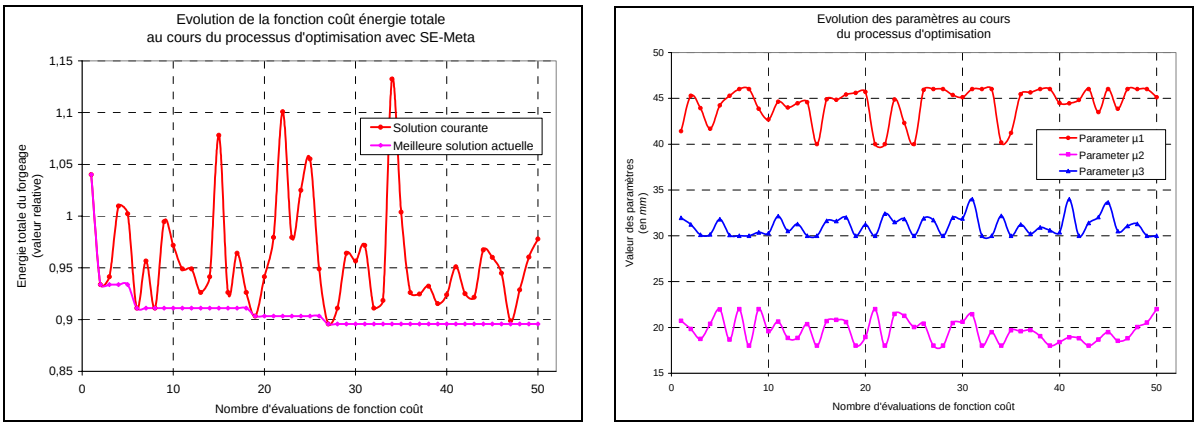

Figure 4.11. Evolution de la fonction coût énergie totale et celle des paramètres au cours du processus d'optimisation avec la Stratégie d'Evolution avec Métamodèle, SE-Meta.

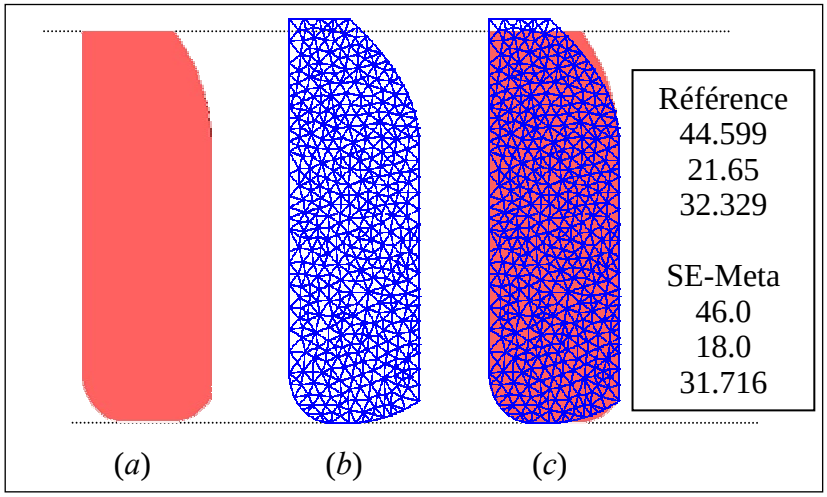

Figure 4.12. Préforme de référence (a) – Préforme optimisée obtenue avec la SE-Meta (b) – Comparaison entre les deux préformes (c).

IV.3.1.1.3. Résultats d'optimisation avec AGMGA

La première variante des deux algorithmes hybrides que nous avons développés, l'AGMGA présenté dans la section II.2, est maintenant employée.

Nous utilisons une population de 120 individus avec 6 évaluations exactes (et donc 6 agglomérats) par génération et un nombre maximal de 8 générations. Les résultats d'optimisation avec l'AGMGA sont présentés sur la *Figure 4.13*.

Cet algorithme donne des résultats très encourageants. En regardant l'évolution de la fonction coût (*Figure 4.13*), on observe qu'elle fluctue plus légèrement qu'avec la SE-Meta. Cette fluctuation diminue à la fin du processus. Tous les points maîtres évalués sont meilleurs (valeur relative <1) que la solution de référence (ce qui est un effet heureux du processus de génération aléatoire de la population initiale). Si on observe l'évolution des paramètres (*Figure 4.13*), on voit qu'ils varient fortement au début du processus mais qu'après cinq générations (30 calculs de fonction coût), leurs valeurs sont relativement stabilisées, et notamment le paramètre μ_3.

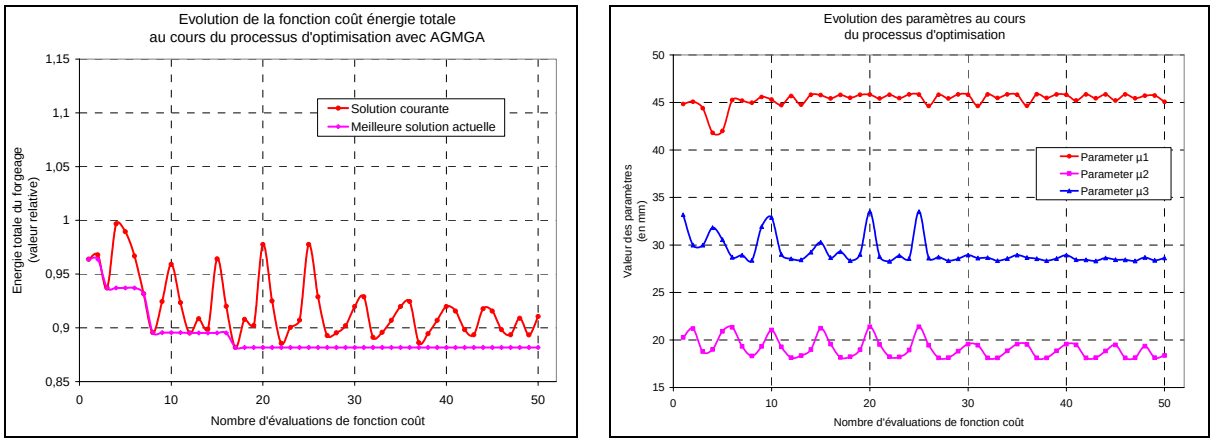

Figure 4.13. Evolution de la fonction coût énergie totale et celle des paramètres au cours du processus d'optimisation avec l'algorithme hybride AGMGA.

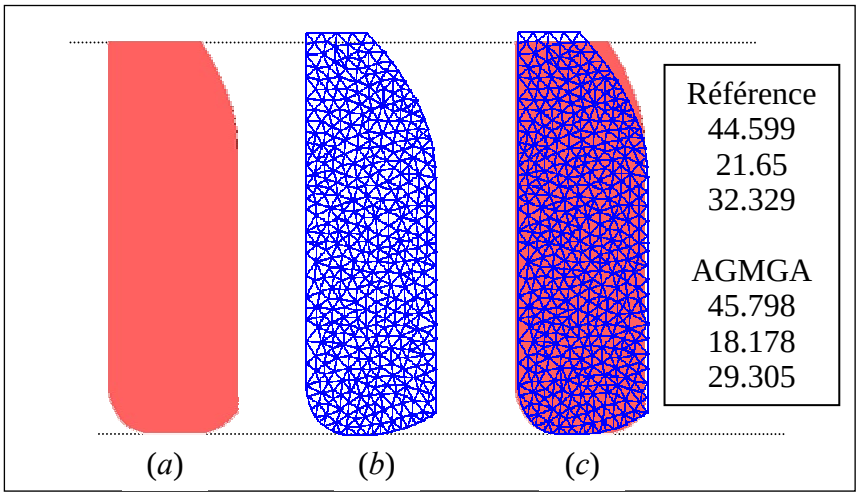

Figure 4.14. Préforme de référence (a) – Préforme optimisée obtenue avec l'algorithme hybride AGMGA (b) – Comparaison entre les deux préformes (c).

L'AGMGA a trouvé une solution qui apporte une amélioration de 11,8%. Elle est meilleure que la solution de la SE-Meta, et elle est obtenue après seulement 17 évaluations avec le calcul du gradient (ce qui correspond à un coût en temps de calcul équivalent à celui de 23 simulations sans calcul du gradient en considérant que le calcul du gradient représente généralement 30% du coût du calcul de la simulation).

La forme obtenue, correspond au jeu de paramètres (μ_1=45,80 ; μ_2=18,18 ; μ_3=29,31), et est présentée sur la *Figure 4.14*. Elle ressemble à la préforme proposée par la SE-Meta.

IV.3.1.1.4. Résultats d'optimisation avec AGMGO

La deuxième variante des algorithmes hybrides, l'AGMGO présentée dans la section II.3, est abordée avec également six évaluations exactes par génération et un nombre maximal de 8 générations. Les résultats obtenus sont présentés sur la *Figure 4.15*. La meilleure solution obtenue apporte une amélioration de 11,4% de la fonction coût. Cette solution (très proche de celle de l'AGMGA), correspond au jeu de paramètres (μ_1=45,81 ; μ_2=18,25 ; μ_3=28,29), est

trouvée dès la douzième évaluation (ce qui correspond au coût de 16 simulation dans gradient en général).

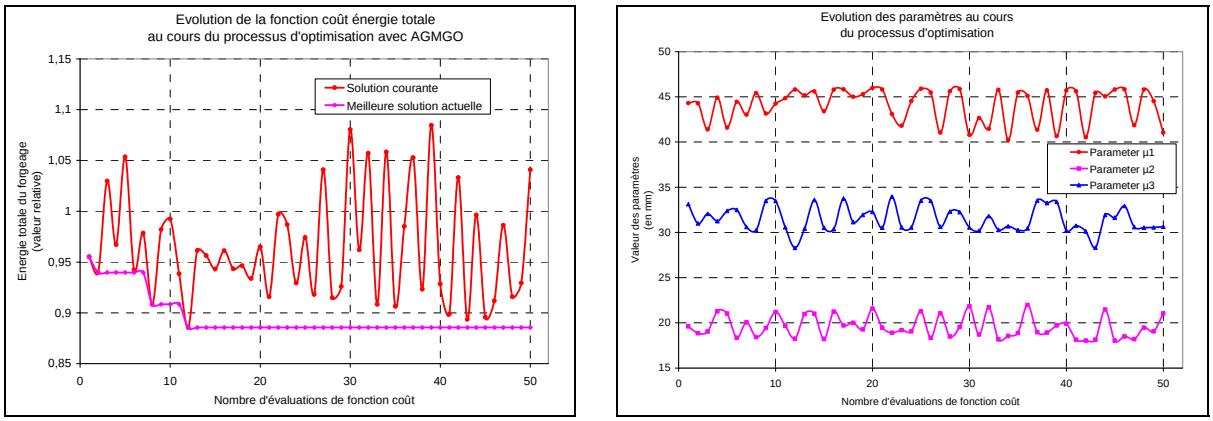

Figure 4.15. Evolution de la fonction coût énergie totale et celle des paramètres au cours du processus d'optimisation avec l'algorithme hybride AGMGO.

En revanche, nous observons une fluctuation très forte des valeurs de la fonction coût et des paramètres d'optimisation au cours de l'optimisation, du début jusqu'à la fin (*Figure 4.15*).

En fait, l'AGMGO essaie d'utiliser les informations disponibles (les points déjà évalués) pour avoir la meilleure interpolation de la fonction coût sur l'ensemble de l'espace paramétrique. A chaque génération, il introduit des nouveaux points maîtres dans les sections inexplorées de l'espace de recherche et pas nécessairement aux endroits les plus proches de l'optimum. Cela entraîne les fluctuations mentionnées précédemment.

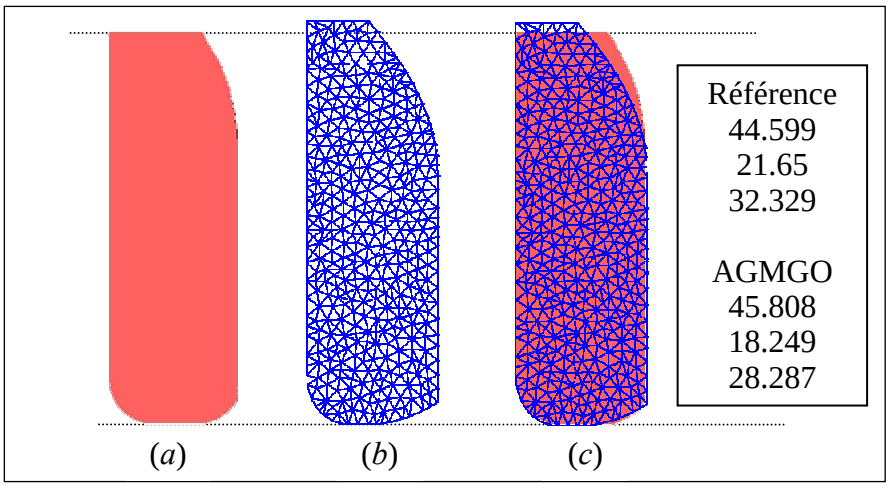

Figure 4.16. Préforme de référence (a) – Préforme optimisée obtenue avec l'algorithme hybride AGMGO (b) – Comparaison entre les deux préformes (c).

Si on compare la préforme (*Figure 4.16*b) fournie par l'AGMGO avec celle de référence, on voit qu'elles sont très différentes et que cette préforme ressemble à la préforme proposée par l'AGMGA et la SE-Meta.

IV.3.1.1.5. Synthèse des résultats d'optimisation

Le *Tableau 4.1* présente une synthèse des résultats de ce benchmark avec la fonction coût "énergie totale". Pour tous les algorithmes d'optimisation, il indique le numéro de la simulation N_{opt} où on a obtenu la meilleure solution en référence au nombre total de simulations N_{tot} effectuées. Il présente la valeur des paramètres (μ_1, μ_2, μ_3) optimaux, la valeur de la fonction coût, et l'amélioration apportée. La *Figure 4.17* synthétise les résultats de convergence des algorithmes d'optimisation employés. Les préformes optimisées et sont regroupées et comparées sur la *Figure 4.18*.

	N_{opt}/N_{tot}	μ_1(mm)	μ_2(mm)	μ_3(mm)	Φ_{opt}	Amélioration
Référence	-	44,60	21,65	32,33	1	-
BFGS	7/8	45,80	19,60	29,11	0,929	-7,1%
SE-Meta	27/50	46,00	18,00	31,72	0,896	-10,4%
AGMGA	17/50	45,80	18,18	29,31	0,882	-11,8%
AGMGO	12/50	45,81	18,25	28,29	0,886	-11,4%

Tableau 4.1 : Synthèse des résultats d'optimisation avec les différents algorithmes pour le benchmark avec la fonction coût énergie totale

On constate que tous les algorithmes arrivent à trouver une meilleure préforme que celle proposée en référence, apportant des améliorations significatives à la fonction coût (de 7% à 11 %). Les deux algorithmes hybrides montrent leurs intérêts avec plus de 11% d'amélioration.

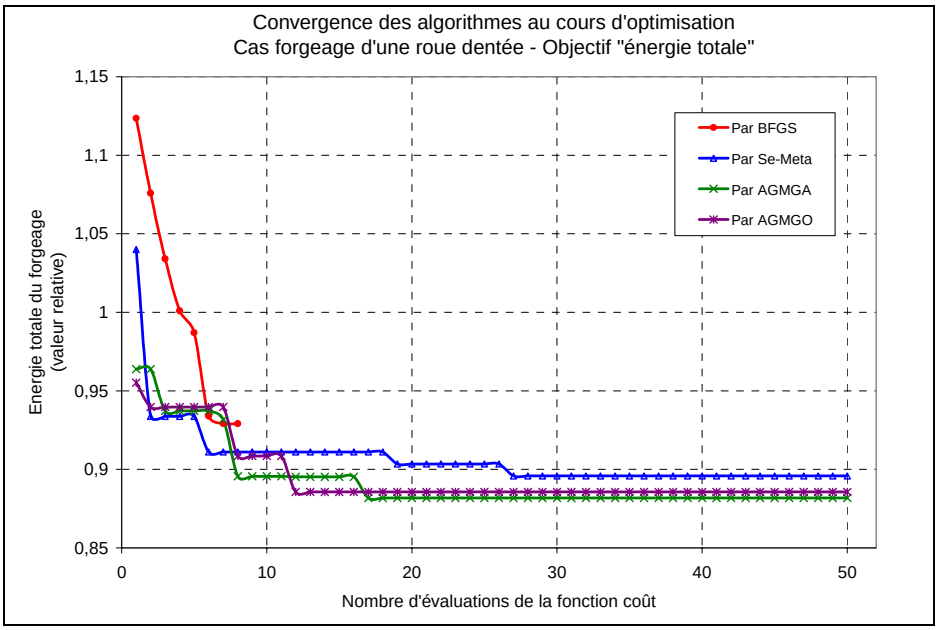

Figure 4.17. Convergence de différents algorithmes au cours du processus d'optimisation de la fonction coût énergie totale.

La *Figure 4.17* montre l'efficacité de l'algorithme BFGS qui fait diminuer rapidement la fonction coût mais qui stagne au bout de quelques itérations. Les algorithmes évolutionnaires montrent leur point fort en trouvant de meilleures solutions, mais ils nécessitent davantage d'évaluations de la fonction coût.

Les solutions obtenues avec les algorithmes évolutionnaires sont fort semblables. Nous pouvons juste comparer leur rapidité à trouver la solution la plus satisfaisante, et constater ainsi l'intérêt de l'hybridation.

Les différentes préformes optimisées présentées sur la *Figure 4.18* sont très proches de l'une de l'autre et bien différentes de la préforme de référence ce qui semble indiquer que ce problème admet un extremum unique. Ces préformes sont en général plus hautes et plus épaisses ce qui permet de diminuer la quantité d'énergie nécessaire au forgeage.

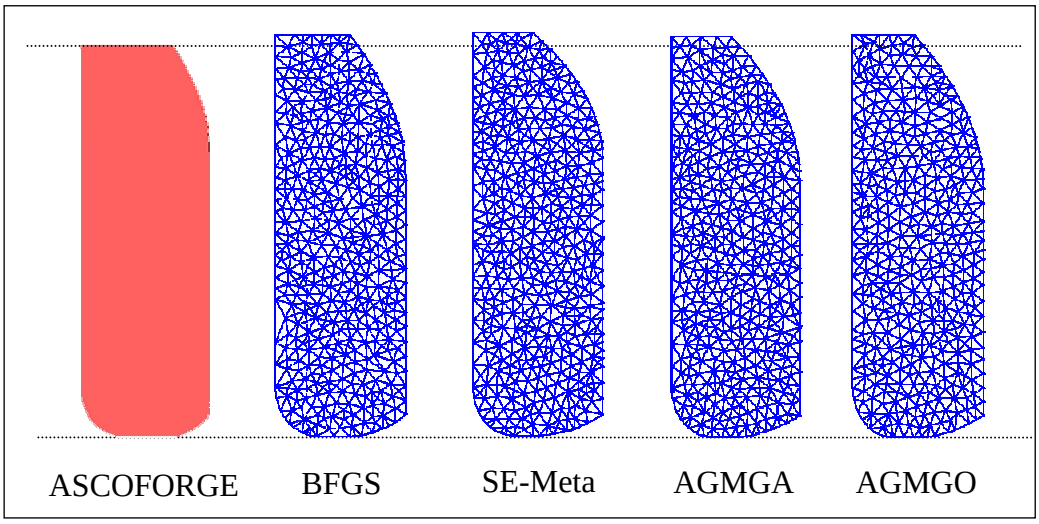

Figure 4.18. Comparaison des préformes obtenues par les différents algorithmes d'optimisation

En observant les valeurs des paramètres optimaux, on constate qu'elles ont tendance à aller vers le coin (μ_1=46,0, μ_2=18,0, μ_3=28,0) de l'espace de recherche, ce qui indique que la solution à ce problème est probablement assez simple, unimodale et que la solution optimale se trouve dans le coin de l'espace.

Afin d'étudier un problème plus complexe non convexe et également caractéristique de cas rencontrés en forgeage, nous nous orientons vers une fonction coût qui combine deux objectifs antagonistes, l'énergie totale et la mesure de la non qualité de la surface (ou de défaut de repli). Avant d'aborder cette fonction coût, nous faisons quelques tests avec la fonction "défaut repli" pour étudier le comportement du problème.

IV.3.1.2. Objectif 2 : Mesure de la non qualité de la surface ou du défaut de repli

Dans cette section, on étudie seulement les deux algorithmes BFGS et SE-Meta. On utilise toujours la préforme proposée par l'industriel ASCOFORGE comme référence et pour laquelle la valeur de la fonction coût repli est de *2,16*.

IV.3.1.2.1. Résultats d'optimisation avec BFGS

L'évolution de la fonction coût et celle des paramètres d'optimisation avec l'algorithme BFGS sont présentées sur la *Figure 4.19*. L'algorithme BFGS part d'un point choisi aléatoirement dans l'espace de recherche et stagne rapidement. Il propose une solution ayant une valeur de la fonction de *2,03*, soit une amélioration de 6% par rapport à la préforme de référence.

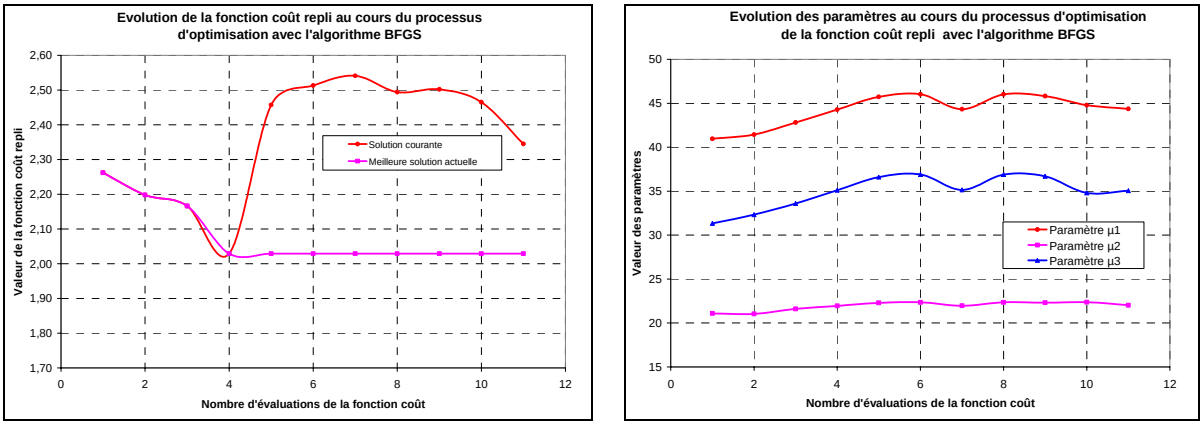

Figure 4.19. Evolution de la fonction coût "repli" et celle des paramètres au cours du processus d'optimisation avec l'algorithme de gradient BFGS.

En observant l'évolution des paramètres, on constate qu'ils varient fortement à l'exception du paramètre μ_2. Après trois itérations, les variations semblent portées dans les mauvaises directions, elles ne peuvent plus améliorer la solution. L'algorithme semble piégé dans un minimum local.

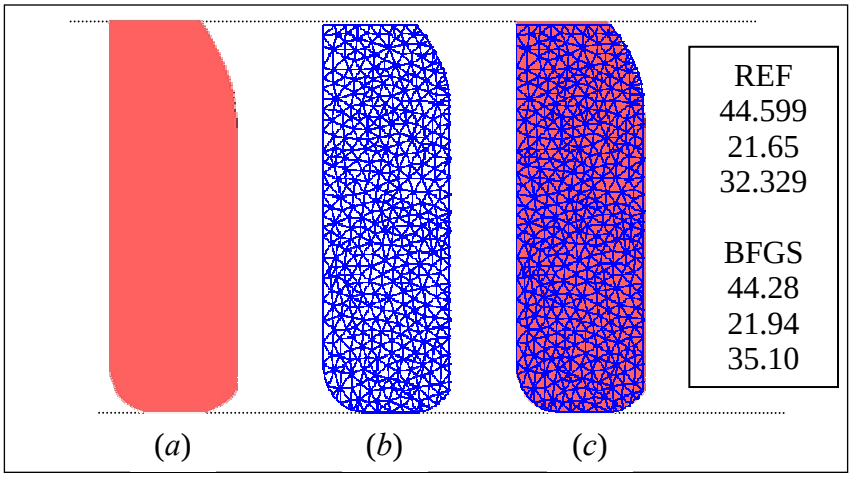

Figure 4.20. Préforme de référence (a) – Préforme optimisée obtenue avec l'algorithme de gradient BFGS (b) – Comparaison entre les deux préformes (c).

La *Figure 4.20* présente la préforme optimale obtenue qui correspond au jeu de paramètres ($\mu_1 = 44,28mm$; $\mu_2 = 21,94mm$; $\mu_3 = 35,10mm$). On voit bien que cette préforme ressemble assez à la préforme de référence.

IV.3.1.2.2. Résultats d'optimisation avec SE-Meta

Avec la SE-Meta, les évolutions de la fonction coût et des paramètres sont présentées sur la *Figure 4.21*.

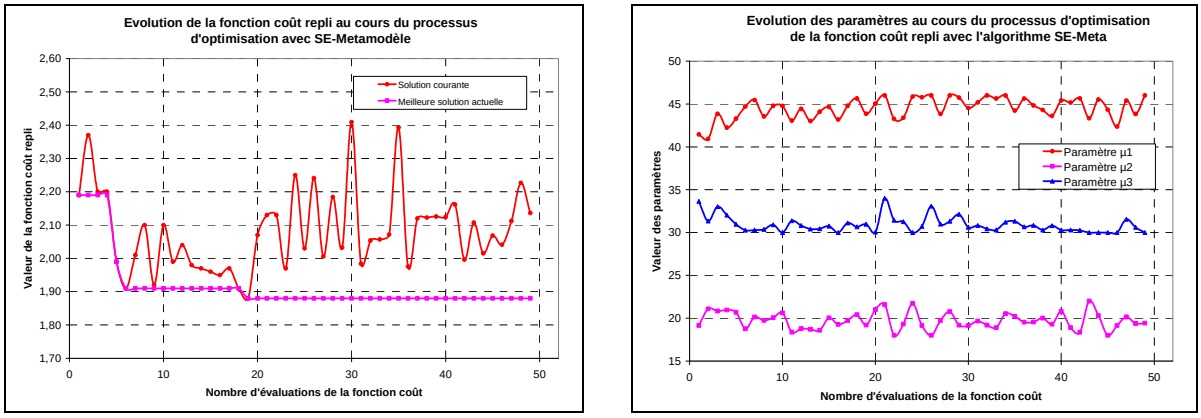

Figure 4.21. Evolution de la fonction coût "repli" et celle des paramètres au cours du processus d'optimisation avec la Stratégie d'Evolution avec Metamodele (SE-Meta).

La SE-Meta montre son efficacité en trouvant une préforme qui améliore la fonction coût d'une valeur relative de 13% ($\Phi_{repli\ opt}$ = 1,88), assez rapidement, à la 19^{ième} évaluation.

La préforme "optimale" (*Figure 4.22*), qui correspond au jeu de paramètres (μ_1=*43,87mm* ; μ_2=*21,94mm* ; μ_3=*35,10mm*), est bien différente de la préforme proposée par BFGS et de celle de référence. Elle est plus épaisse et moins haute. On peut aussi remarquer que cette préforme est bien différente de celle qui minimise l'énergie totale de forgeage (voir *Figure 4.18*). La combinaison des deux fonctions coûts conduira à la recherche d'un compromis entre deux objectifs antagonistes.

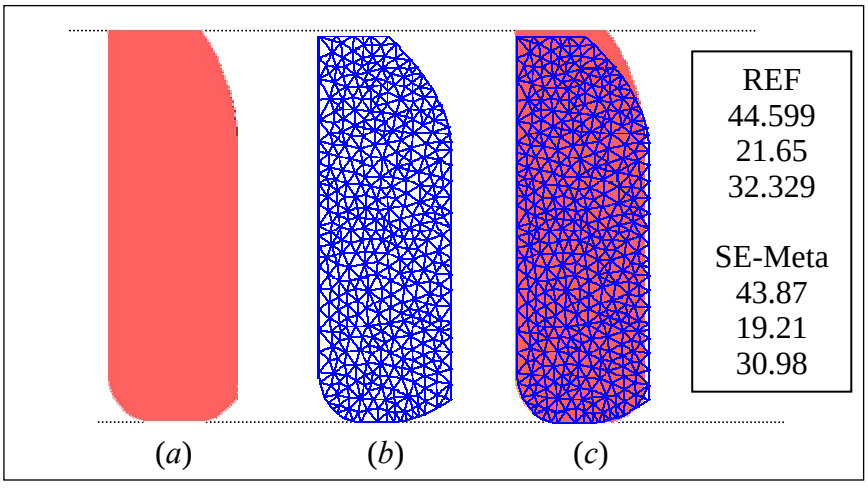

Figure 4.22. Préforme de référence (a) – Préforme optimisée obtenue avec la SE-Meta (b) – Comparaison entre les deux préformes (c).

IV.3.2. Optimisation multi-critères

Afin d'obtenir un second benchmark plus caractéristique des problèmes non convexes avec plusieurs extrema, nous considérons la fonction coût qui est la combinaison des deux fonctions précédentes. Elle s'écrit :

$$\Phi_{tot} = 0.5\frac{\Phi_1}{\Phi_{1ref}} + 0.5\frac{\Phi_2}{\Phi_{2ref}}$$

avec Φ_1 et Φ_{1ref} : fonction coût "énergie totale" et sa valeur de référence

Φ_2 et Φ_{2ref} : fonction coût "repli" et sa valeur de référence

La préforme proposée par l'industriel ASCOFORGE est toujours utilisée comme référence. En conséquence, la valeur de sa fonction coût est 1.

IV.3.2.1. Résultats d'optimisation avec BFGS

Les résultats d'optimisation obtenus avec l'algorithme BFGS sont présentés sur la *Figure 4.23* montrant sa faiblesse sur les problèmes non convexes avec une stagnation après seulement 3 itérations. Malgré un redémarrage de l'algorithme la descente dans la direction du gradient ne permet pas d'améliorer la fonction coût.

La meilleure solution correspond au jeu de paramètres (μ_1=*40,50mm* ; μ_2=*17,95mm* ; μ_3=*30,05mm*), et est obtenue en trois itérations. Elle apporte une amélioration de seulement 2,7%. Si on regarde l'évolution des paramètres, on voit qu'ils varient très légèrement autour de leurs valeurs initiales. L'algorithme BFGS est très probablement coincé dans un optimum local.

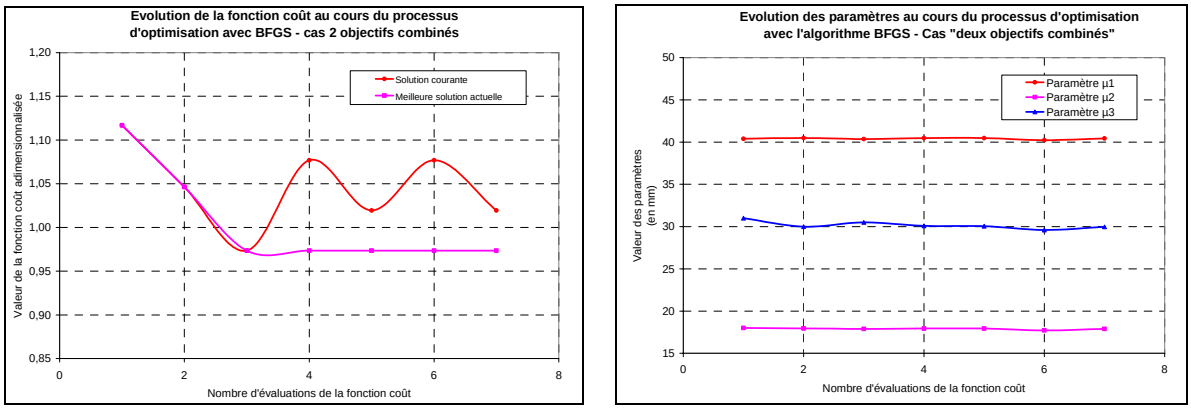

Figure 4.23. Evolution de la fonction coût "2 objectifs combinés" et celle des paramètres au cours du processus d'optimisation avec l'algorithme de gradient BFGS.

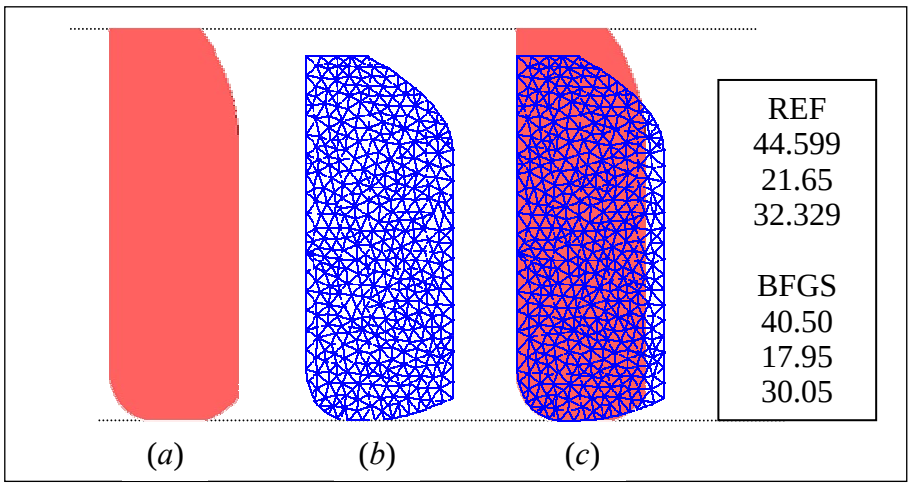

Figure 4.24. Préforme de référence (a) – Préforme optimisée obtenue avec l'algorithme de gradient BFGS (b) – Comparaison entre les deux préformes (c).

IV.3.2.2. Résultats d'optimisation avec SCPIP

Pour résoudre ce problème d'optimisation non convexe, nous faisons appel à l'autre algorithme d'optimisation à base de gradient, le SCPIP présenté dans la section I.5.1. En partant du même point de départ, le SCPIP arrive à faire mieux que BFGS, et améliore la fonction coût de 5,2%.

Avec cet algorithme, on constate une descente brutale (très forte) de la valeur de la fonction coût au début de l'algorithme. Puis, on observe une légère amélioration entre la deuxième et la troisième itération, qui est suivie par une stagnation de l'algorithme.

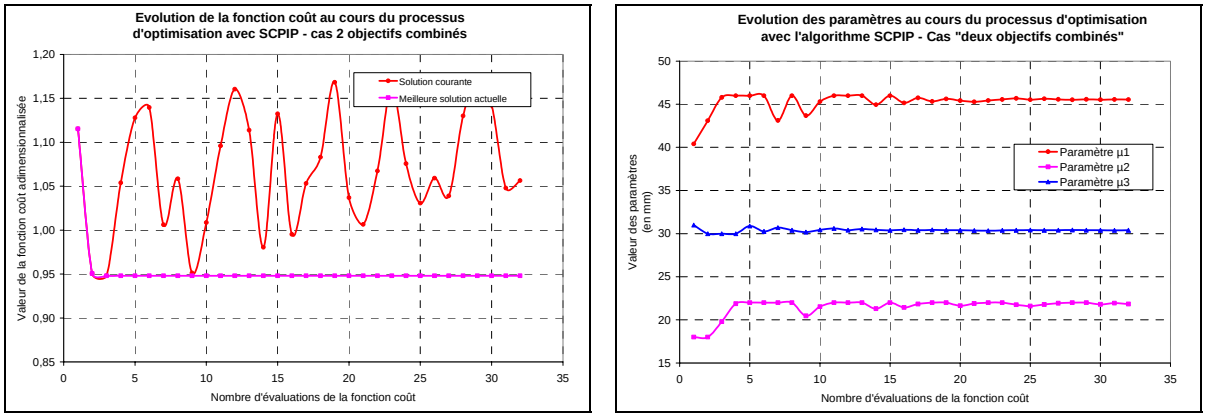

Figure 4.25. Evolution de la fonction coût "2 objectifs combinés" et celle des paramètres au cours du processus d'optimisation avec l'algorithme SCPIP.

Contrairement à l'algorithme BFGS dont le pas de descente est contrôlé, l'algorithme SCPIP fait varier très fortement les paramètres d'optimisation au début, puis plus légèrement par la suite.

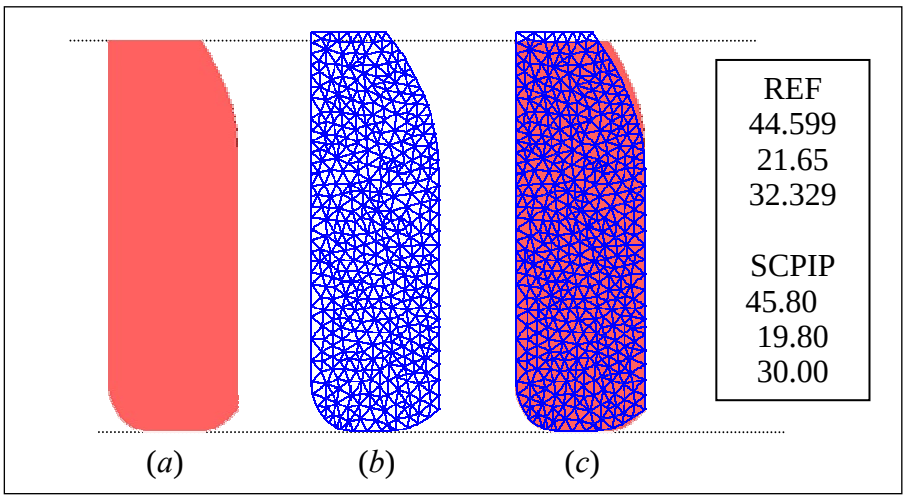

Figure 4.26. Préforme de référence (a) – Préforme optimisée obtenue avec l'algorithme de gradient SCPIP (b) – Comparaison entre les deux préformes (c).

La préforme optimisée par SCPIP est légèrement différente de celle de référence. Elle est plus haute mais moins épaisse. Il faut remarquer que cette préforme est complètement différente de celle trouvée par BFGS, alors que les deux algorithmes utilisent le même point de départ. Ce qui met bien an relief l'incapacité de BFGS à s'extraire de l'extremum local où le choix du point de départ l'a précipité.

IV.3.2.3. Résultats d'optimisation avec SE-Meta

La SE-Meta montre ici aussi son potentiel en trouvant une solution qui améliore la fonction coût de 9,05% (voir *Figure 4.27*).

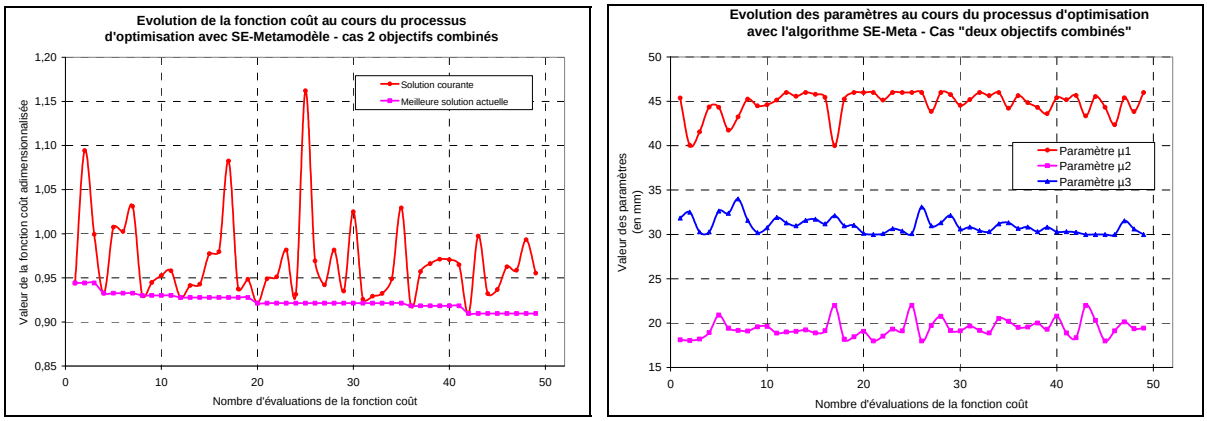

Figure 4.27. Evolution de la fonction coût "2 objectifs combinés" et celle des paramètres au cours du processus d'optimisation avec la Stratégie d'Evolution avec Metamodele SE-Meta.

Grâce à son caractère stochastique et un tirage chanceux, la SE-Meta a un excellent début en trouvant une solution meilleure que celle proposée par BFGS ainsi que celle de SCPIP. Cela signifie qu'un algorithme purement stochastique serait ici meilleur que les deux algorithmes de gradient, et est une indication sur le caractère multi-extrema de la fonction coût considérée. Il faut remarquer que SE-Meta commence par *2*nbparam* = 6 calculs sur des valeurs aléatoires. Ces 6 valeurs peuvent être considérées comme calculées "simultanément", sans

antériorité entre elles. Pendant le processus d'optimisation, la SE-Meta explore tout l'espace de recherche en faisant varier les paramètres dans toutes les directions (voir *Figure 4.27*), ce qui entraîne une forte fluctuation de la fonction coût. Cette exploration permet d'améliorer la solution initiale en trouvant d'autres solutions encore plus intéressantes.

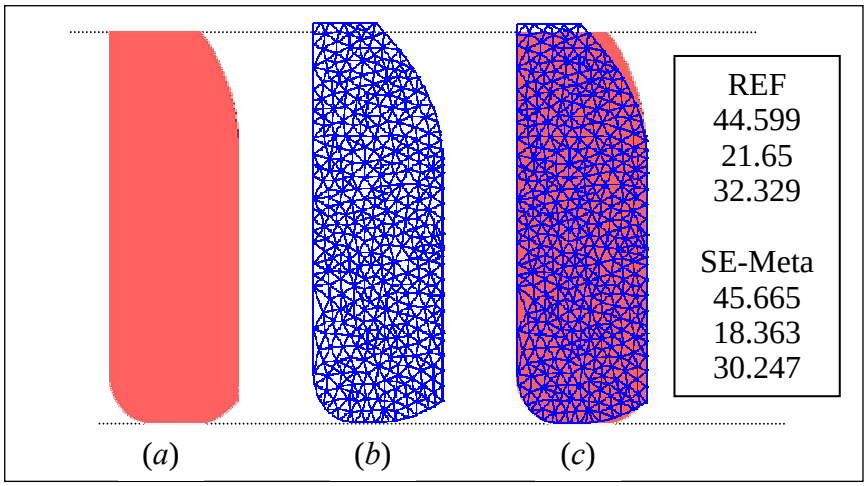

Figure 4.28. Préforme de référence (a) – Préforme optimisée obtenue avec la SE-Meta (b) – Comparaison entre les deux préformes (c).

La meilleure solution est trouvée à la 42^{ième} évaluation, avec le jeu de paramètres (μ_1=*43,87mm* ; μ_2=*21,94mm* ; μ_3=*35,10mm*) (voir *Figure 4.28*). Par rapport à la préforme de référence, elle est plus haute et plus pointue. Cette différence de forme permet une amélioration de la valeur de fonction coût de 9,05%.

IV.3.2.4. Résultats d'optimisation avec AGMGA

Avec l'algorithme hybride AGMGA et une population de 120 individus avec 6 points maîtres (6 évaluations exactes par génération), plusieurs solutions intéressantes ont été trouvées. L'évolution de la fonction coût et celle des paramètres sont présentées sur la *Figure 4.29*.

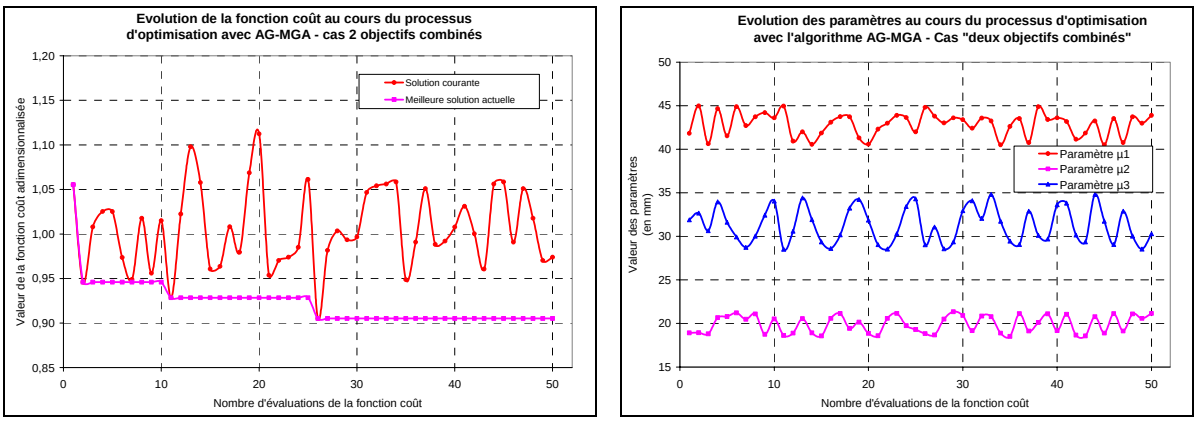

Figure 4.29. Evolution de la fonction coût "2 objectifs combinés" et celle des paramètres au cours du processus d'optimisation avec l'algorithme hybride AGMGA.

Parmi les bonnes solutions, celle trouvée à la 26^{ième} évaluation est la meilleure, avec 9,5% d'amélioration. Elle est ainsi meilleure que celle proposée par la SE-Meta.

On note une forte fluctuation de l'évolution de la fonction et des paramètres d'optimisation (*Figure 4.29*), du début de l'optimisation jusqu'à la fin. Cela est assez différent de l'application précédente avec la fonction coût "énergie totale" seule, pour laquelle ces fluctuations se résorbent à la fin de l'optimisation. Une explication est que dans ce cas présent la fonction coût a plusieurs optima. En 50 évaluations, l'AGMGA n'arrive pas encore à stabiliser les solutions. A priori, il demandait davantage d'itérations de l'algorithme pour observer cette stabilisation.

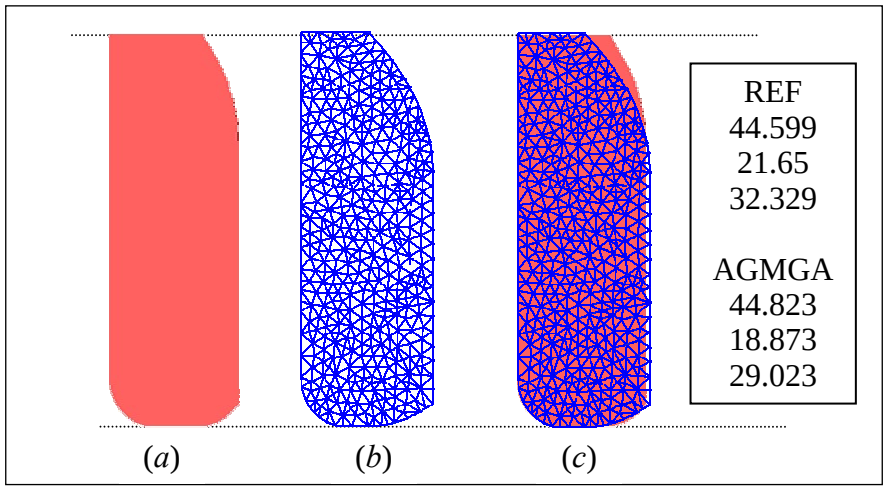

Figure 4.30. Préforme de référence (a) – Préforme optimisée obtenue avec l'algorithme hybride AGMGA (b) – Comparaison entre les deux préformes (c).

Si on examine la préforme proposée par l'AGMGA, on constate qu'elle est différente de la préforme de référence et de celle proposée par SE-Meta. Elle est plus épaisse, plus pointue et légèrement plus haute que la première.

IV.3.2.5. Résultats d'optimisation avec AGMGO

Le dernier algorithme est celui hybride AGMGO. Avec six évaluations par génération, l'AGMGO arrête sa recherche après 8 générations. La *Figure 4.31* présente les résultats obtenus avec cet algorithme.

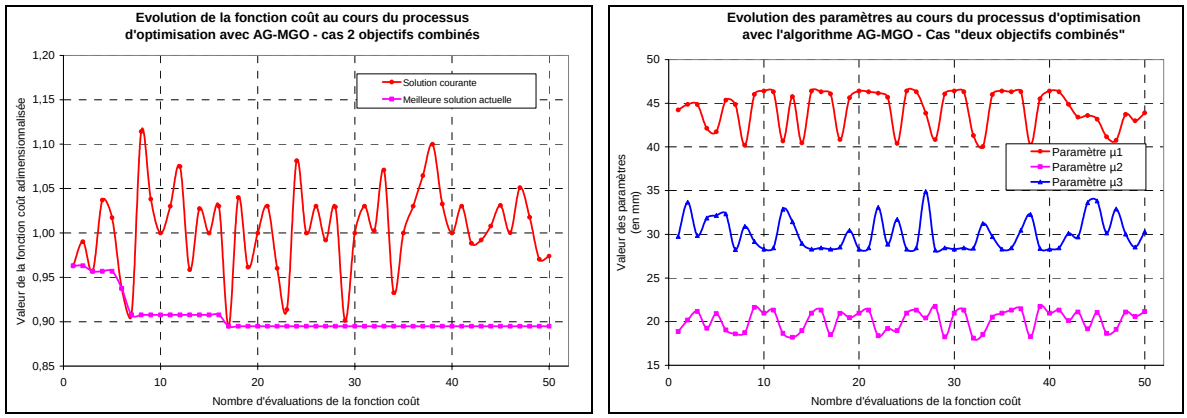

Figure 4.31. Evolution de la fonction coût "2 objectifs combinés" et celle des paramètres au cours du processus d'optimisation avec l'algorithme hybride AGMGO.

Comme avec les deux AE précédents, des l'initialisation (pour SE-Meta) ou la première génération (pour les AG), AGMGO trouve une solution meilleure que la référence (cequi confirme que BFGS a plutôt été malchanceux lors du tirage aléatoire de son point de départ). Pendant le processus d'optimisation, l'AGMGO explore bien tout l'espace de recherche, comme le montre l'évolution de la fonction coût et des paramètres (*Figure 4.31*) qui varient dans toutes les directions. La valeur de la fonction coût fluctue fortement pendant l'optimisation comme observé avec la fonction coût "énergie totale" seule.

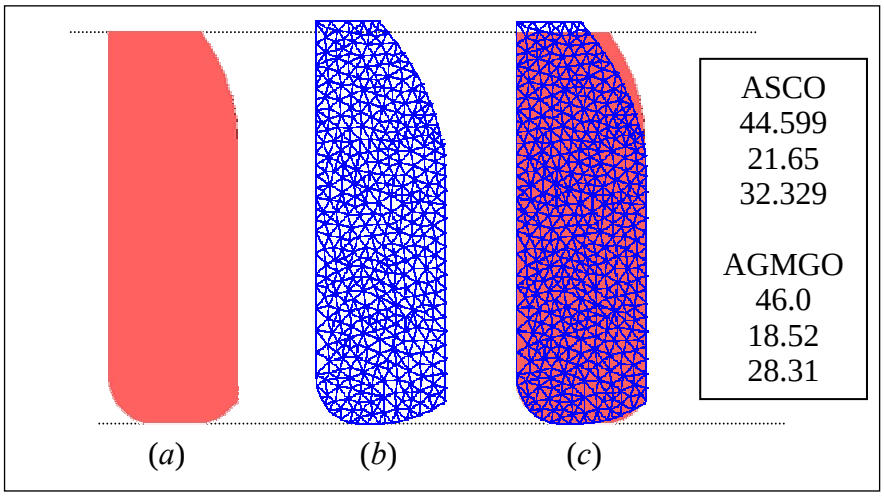

Figure 4.32. Préforme de référence (a) – Préforme optimisée obtenue avec l'algorithme hybride AGMGO (b) – Comparaison entre les deux préformes (c).

La meilleure solution est obtenue à la 17^{ième} évaluation de la fonction coût avec une amélioration de 10,5% de sa valeur. C'est l'amélioration la plus importante obtenue pour ce cas. La préforme optimale proposée est présentée sur la *Figure 4.32*. Si on la compare à celle de référence, on note une différence : celle optimisée par l'AGMGO est plus haute et plus pointue mais elles ont presque la même épaisseur. Notons que les différentes préformes obtenues sont ainsi fort différentes, preuve du caractère chahuté et multi-optima de la fonction coût étudiée.

IV.3.2.6. Comparaisons des algorithmes

Nous présentons une synthèse des résultats obtenus avec tous les algorithmes d'optimisation avec la fonction coût "2 objectifs combinés".

Pour chaque algorithme d'optimisation, le *Tableau 4.2* indique le numéro N_{opt} de la simulation où la meilleure solution est obtenue ainsi que le nombre total de simulations N_{tot} effectuées. Il présente la valeur de la fonction coût et des améliorations obtenues avec les préformes optimales. Il présente enfin le coût de chaque algorithme pour trouver une solution équivalente à celle trouvée par SE-Meta, en valeur relative par rapport à celui de la SE-Meta (on considère toujours que le calcul du gradient représente 30% du calcul de la simulation, ce qui est plutôt une extrapolation pour des problèmes plus complexes (comportement non linéaire) et avec plus de nœuds qu'ici).

Un premier regard au *Tableau 4.2* montre que tous les algorithmes trouvent une meilleure solution que le cas de référence et que la valeur des deux fonctions coûts peut être réduite de

plus de 10%. Les améliorations obtenues se divisent en 2 groupes d'amplitudes différentes. Le groupe 1 contient les algorithmes à direction de descente qui améliorent la fonction coût seulement de 3% à 5%. Le groupe 2 réunit les trois autres algorithmes évolutionnaires qui apportent des améliorations plus significatives de l'ordre de 10% grâce à leur capacité à rechercher une solution plus globale.

	N_{opt}/N_{tot}	Φ_{tot}	*Amélioration de* Φ_{ener}	*Amélioration de* Φ_{repli}	*Amélioration de* Φ_{tot}	*Coût % SE-Meta*
Référence	-	1	-	-	-	-
BFGS	3/7	0,973	-1,8%	-3,6%	-2,7%	Stagne
SCPIP	3/32	0,948	-6,9%	-3,5%	-5,2%	Stagne
SE-Meta	42/50	0,9095	-10,5%	-7,6%	-9,05%	1
AGMGA	26/50	0,905	-8,9%	-10,1%	-9,5%	0,8
AGMGO	17/50	0,895	-10,3%	-10,7%	-10,5%	0,53

Tableau 4.2 : Synthèse des résultats d'optimisation avec différents algorithmes pour le benchmark du forgeage d'une roue dentée avec la fonction coût 2 objectifs combinés.

La *Figure 4.33* synthétise la convergence des algorithmes. La *Figure 4.34* regroupe les préformes proposées et celle de référence.

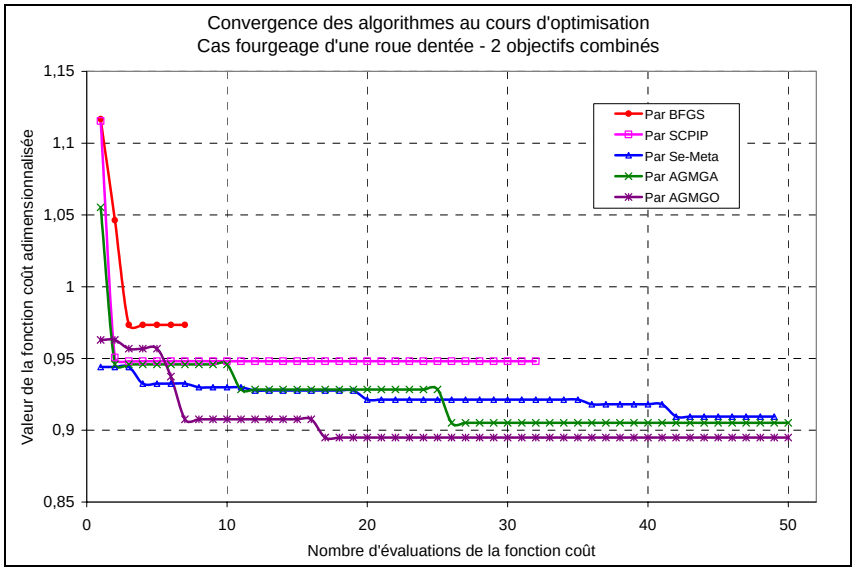

Figure 4.33. Convergence de différents algorithmes au cours du processus d'optimisation de la fonction coût "2 objectifs combinés".

La convergence des algorithmes présentée sur la *Figure 4.33* montre clairement que les algorithmes globaux (comme SE-Meta, AGMGA, AGMGO) sont plus performants que les locaux (comme BFGS). Pour les algorithmes locaux, les résultats dépendent fortement du point de départ mais ils sont obtenus en seulement quelques itérations. Les algorithmes

globaux ont souvent besoin de beaucoup plus d'évaluations pour atteindre leur optimum, mais on observe ici que dès la première génération, c'est-à-dire dès les 6 premiers calculs effectués pour des valeurs aléatoires des paramètres (que ce soit pour SE-Meta ou AGMGA, AGMGO) la solution obtenue est meilleure ou équivalente à celle proposée par SCPIP ou BFGS. Autrement dit, un tirage aléatoire de 6 valeurs est plus efficace que tout algorithme de gradient.

On compare maintenant les résultats obtenus avec les deux nouveaux algorithmes hybrides à celui obtenu par la SE-Meta. Dans le *Tableau 4.2*, nous constatons que la solution de la SE-Meta minimise au mieux l'énergie totale, tandis que les algorithmes hybrides minimisent préférentiellement plus le potentiel de repli. L'AG-MGO s'avère particulièrement performant en combinant une grande réduction des deux objectifs. Toutefois, la différence entre les solutions proposées par la SE-Meta et les algorithmes hybrides est assez fable comme le montre la *Figure 4.34*.

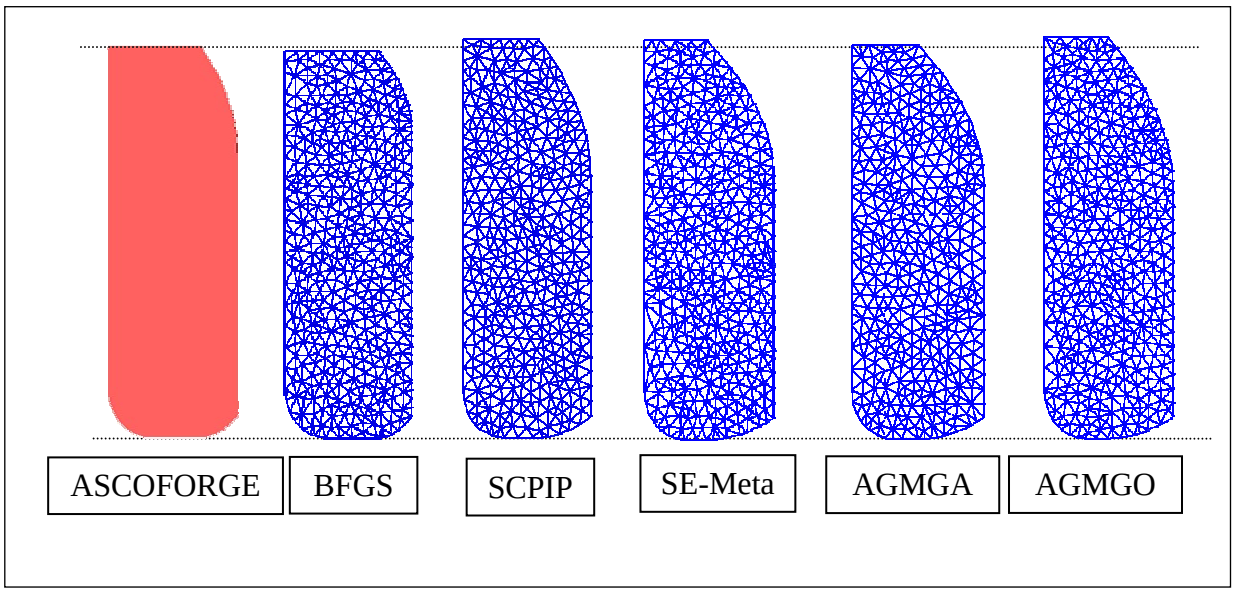

Figure 4.34. Comparaison des préformes proposées par les différents algorithmes d'optimisation

Une autre remarque sur le coût de l'optimisation peut être formulée. Les deux algorithmes permettent d'obtenir uns solution équivalente en beaucoup moins de temps. Notamment l'AGMGO, permet de diviser par deux ce coût des calculs, ce qui permet exercer les qualités de cette hybridation.

IV.4. Conclusions

Différentes conclusions peuvent être tirées :

- tous les algorithmes d'optimisation trouvent de meilleures solutions que la solution de référence
- les algorithmes de type évolutionnaire sont plus performants que les algorithmes à direction de descente (comme BFGS ou SCPIP)

- les trois algorithmes évolutionnaires (AGMGA et AGMGO et SE-Meta) proposent des améliorations semblables et d'environ 10%.

- les deux algorithmes hybrides s'avèrent surtout plus rapides (jusqu'à deux fois plus rapide que la SE-Meta) pour trouver une solution intéressante.

A travers cet exemple, nous voyons bien que l'on peut appliquer les algorithmes évolutionnaires à la résolution des problèmes complexes d'optimisation de forgeage 3D avec un coût acceptable (moins de 50 simulations).