

UNIVERSITE SAAD DAHLEB DE BLIDA

Faculté des Sciences de l'Ingénieur

Département d'Aéronautique

MEMOIRE DE MAGISTER

Spécialité : Aéronautique

OPTIMISATION AERODYNAMIQUE D'UN PROFIL D'AILE A L'AIDE D'ALGORITHMES GENETIQUES

Par

Mamar BOUZIANE

Devant le jury composé de :

S. BOUKRAA	Professeur, U. de Blida	Président
K. MOHAMMEDI	Maître de Conférences, U.M.B., Boumerdes	Rapporteur
T. REZOUG	Maître de Conférences, U. de Blida	Examineur
A. TAHI	Maître- assistant, U. de Blida	Examineur

Blida, décembre 2006

RESUME

Ces dernières années, les algorithmes génétiques ont fait l'objet d'un grand intérêt pour l'optimisation de certaines classes de problèmes.

L'objectif de ce travail porte principalement sur l'optimisation de profils d'aile à l'aide des algorithmes génétiques pour le cas d'un écoulement subsonique. La modélisation de l'écoulement considéré comme potentiel, est réalisée par une méthode des panneaux 2-D couplée avec un modèle de couche limite qui fournit un calcul rapide et robuste des coefficients de portance, du moment de tangage et de la traînée. Une fonction objective (fitness) à plusieurs paramètres est définie par la minimisation du rapport de la traînée sur la portance pour des valeurs fixes de la portance et du moment de tangage.

ABSTRACT

The recent years, a lot of interest has been demonstrated in using genetic algorithms to optimise certain classes of problems.

The main objective of this work is the optimization of an airfoil using genetic algorithms for a subsonic flow. The modelling of the flow considered as potential is realized by 2D panel method coupled with a boundary layer model provides fast and robust computation of the lift, pitching moment and drag. A multi parameter objective function (fitness) is defined by the minimum drag to lift ratio at fixed lift and pitching moment.

ملخص

في السنوات الأخيرة، الخوارزميات الوراثة، كانت محل اهتمام علمي كبير من أجل إيجاد حلول حسنة لبعض المواضيع.

الهدف الأساسي من هذا العمل هو إيجاد حلول مثالية لمقطع من الجناح (جناح الطائرة) بواسطة الخوارزميات الوراثة في سيلان دون سرعة الصوت. إن تشكيل السيلان في إطار نظرية السيلان الكموني، تهدف إلى تفعيل طريقة الأضلاع في فضاء ذو بعدين و بموازاة مع نظام جملي للطبقة النهائية التي تقوم بحساب سريع و دقيق لعوامل قوة الرفع، عزم الخفقان و إعاقاة الحركة. دالة موضوعية (فتناس) بعدة متغيرات مبنية على أساس جعل نسبة إعاقاة على قوة الرفع صغيرة لأجل قيم ثابتة لقوة الرفع و عزم الخفقان.

REMERCIEMENTS

Je tiens à exprimer ma profonde gratitude à mon directeur de recherche, Kamel MOHAMMEDI, maître de conférences de l'université M. BOUGARA de Boumerdès, pour sa disponibilité et ses conseils avisés. Je souhaite également le remercier pour m'avoir laissé une grande liberté dans mon travail tout en me faisant bénéficier de ses connaissances dans le domaine de l'aérodynamique.

Ce travail n'aurait pu aboutir sans la précieuse aide de M^r MELBOUS A.E.K, dont l'expérience m'a beaucoup apporté.

J'adresse également mes sincères remerciements:

À Monsieur S. BOUKRAA, Professeur au département d'Aéronautique de l'Université de Blida, pour avoir accepté d'être président du jury.

À Monsieur T. REZOUG, Maître de conférences au département d'Aéronautique de l'Université de Blida, pour l'honneur qu'il m'a fait en acceptant de participer à ce jury.

A Monsieur A. TAHI, Maître-assistant à l'Université de Blida, qui a bien voulu accepter, de participer au jury.

Je me dois aussi de remercier tous les enseignants ainsi que tout le personnel du département d'aéronautique.

A tous ceux qui ont contribué d'une manière directe ou indirecte, de près ou de loin, à la concrétisation de ce travail.

Mamar BOUZIANE

TABLES DES MATIERES

RESUME	
REMERCIEMENTS	
TABLE DES MATIÈRES	
LISTE DES ILLUSTRATIONS, FIGURES ET TABLEAUX	
INTRODUCTION	11
1. OPTIMISATION DE FORME EN AERODYNAMIQUE : METHODES ET ETAT DE L'ART	15
1.1. Introduction.	15
1.2. Classification des méthodes d'optimisation	15
1.2.1 Optimisation déterministe locale.....	17
1.2.2 Optimisation déterministe globale.....	20
1.2.3 Méthodes d'Optimisation Stochastiques	21
1.3 Etat de l'art de l'optimisation de forme en aérodynamique	26
1.3.1. Les approches déterministes.....	26
1.3.2. Les approches stochastiques.....	30
1.3.3. Les méthodes hybrides.....	32
1.3.4. Méthodes diverses.....	33
2. LES ALGORITHMES GÉNÉTIQUES.....	34
2.1 Principes généraux.....	34
2.2 Description détaillée.....	36
2.2.1 Codage des données.....	36
2.2.2 Génération aléatoire de la population initiale.....	37
2.2.3 Gestion des contraintes.....	37
2.2.4 Opérateur de Croisement.....	38
2.2.5 Opérateur de mutation	40
2.2.6 Principes de sélection	41
2.3 Améliorations classiques.....	42

2.3.1 Introduction.....	42
2.3.2 Scaling.....	43
2.3.3. Sharing.....	45
2.3.4. Sharing clustérisé.....	47
2.3.5. L'élitisme.....	48
2.3.6. Algorithmes génétiques et recuit simulé.....	49
2.3.7. Recherche multi objectifs.....	51
2.3.8 Association des AG avec des méthodes locales.....	52
2.4 Parallélisme.....	54
2.5.1 Parallélisme par îlots.....	55
2.5.2 Parallélisation des calculs.....	56
2.5.3 Conclusion.....	57
2.5 Résultats de convergence théorique des algorithmes génétiques.....	57
3. APPLICATION À L'OPTIMISATION DE PROFILS D'AILE.....	59
3.1. Introduction.....	59
3.2. Modélisation de l'écoulement.....	59
3.2.1 Introduction.....	59
3.2.2 Ecoulements potentiels et méthode de panneaux	61
3.2.3 Modèle de la couche limite.....	66
3.2.4 Calcul du coefficient de la traînée.....	75
3.3. Paramétrisation du profil.....	77
3.3.1. Représentation du profil NACA à 4 chiffres	77
3.3.2. représentation en B-spline	78
3.4. Spécificités de l'Algorithme Génétique.....	81
3.4.1 Le codage.....	81
3.4.2 L'espace de recherche	82
3.4.3. La fonction objective (fitness) et les contraintes.....	84
3.4.4. L'opérateur de sélection	84
3.4.5. Croisement et mutation	85
3.4.6. Procédure élitiste	86
3.4.7. Les paramètres d'AGs.....	86
3.5. Organigramme.....	87
4. RÉSULTATS ET INTERPRÉTATION	88
4.1. Convergence de l'algorithme génétique.....	88

4.2. L'approche inverse.....	89
4.3. L'approche directe.....	92
4.3.1. Représentation NACA	92
4.3.2. Représentation à l'aide de B-Splines	99
CONCLUSION	102
APPENDICE A.....	103
APPENDICE B.....	105
REFERENCE.....	106

LISTE DES ILLUSTRATIONS, GRAHIQUES ET TABLEAUX

Figure 1.1 : Techniques d’optimisation.....	17
Figure 1.2 : Méthodes Evolutionnistes.....	24
Figure 1.3 : Méthode d’optimisation vs efficacité.....	25
Figure 2.1: Principe général des algorithmes génétiques.....	35
Figure.2.2 : Croisement à 1 point.....	38
Figure 2.3: Croisement à 2 points.....	39
Figure 2.4: Croisement barycentrique.....	40
Figure 2.5: Principe de l’opérateur de mutation.....	40
Figure 2.6: Exemple où les sélections classiques risquent de ne reproduire qu’un individu.....	43
Figure 2.7: Fonction de Scaling exponentielle.....	44
Figure 2.8: Allure de l’évolution de k en fonction des générations.....	45
Figure 2.9: Objectif du Sharing.....	46
Figure 2.10: Allure de $S\left(\frac{d}{\sigma_{share}}\right)$	46
Figure 2.11: Principe du croisement avec recuit simulé.....	49
Figure 2.12: Surface de Pareto et voisinages.....	53
Figure 2.13: Principe de fonctionnement du parallélisme par îlots.....	55
Figure 2.14 : Principe de fonctionnement de la parallélisation des calculs.....	57
Figure 3.1 : L’approximation d’un modèle d’écoulement non visqueux.....	59
Figure 3.2 : Couplage du modèle d’écoulement non visqueux avec modèle de couche limite.....	60
Figure 3.3 : Modification de profil d’aile.....	61
Figure 3.4 : Définition de s , de r et de θ	62
Figure 3.5 : Notation des panneaux sur le profil.....	63
figure 3.6 : Validation du modèle d’écoulement non visqueux.....	65
Figure 3.7 : Équivalence en débit d’écoulement.....	69

Figure 3.8 : Modèle de couche limite laminaire-turbulente utilisé pour les différents codes.....	74
figure 3.9 : Validation du modèle d'écoulement visqueux.....	76
Figure 3.10 : Représentation d'un profil d'aile type NACA.....	78
Figure 3.11 : L'influence de l'ordre sur une courbe B-spline à 5 points de contrôle.....	80
Figure 3.12 : Représentation B-spline et points de contrôles de profil d'aile.....	81
Figure 3.13 : Les profils d'aile avant croisement.....	85
Figure 3.14 : Les profils d'aile après croisement.....	85
Figure 3.15 : Mutation simple des bits.....	86
figure 3.16 : Organigramme de calcul.....	87
Figure 4.1 : Variation de fitness en fonction de nombre de génération et individus.....	88
Figure 4.2 : Influence de type de codage.....	89
Figure 4.3 : L'évolution de fitness en fonction des génération.....	90
Figure 4.4 : L'évolution de forme de profil d'aile en fonction des génération.....	91
Figure 4.5 : L'évolution de coefficient de pression en fonction des génération.....	91
Figure 4.6 : Evolution de la fitness en fonction des générations.....	93
Figure 4.7.a : Les formes des profils d'aile pour différentes générations.....	94
Figure 4.7.b : Les coefficients de pression des profils d'aile pour différentes générations.....	94
Figure 4.8.a : Forme du profil d'aile optimal obtenu et sa distribution de pression.....	95
Figure 4.8.b : Forme du profil d'aile optimal et sa distribution de pression obtenu par différentiation automatique (C.Wauquiez) [3].....	95
Figure 4.9.a : Formes des meilleurs profils d'aile initiale et optimal.....	96
Figure 4.9.b : Distributions de pression des meilleurs profils d'aile initial et optimal.....	97
Figure 4.10 :Variation du coefficient de portance en fonction du nombre de générations.....	97
Figure 4.11 : Variation du coefficient de moment tangage en fonction du nombre de générations.....	98
Figure 4.12 : Variation du coefficient de traînée vs le nombre de générations.....	98
Figure 4.13 : Evolution de la fitness en fonction des générations.....	99
Figure 4.14.a : Formes des profils d'aile pour différentes générations.....	100

Figure 4.14.b : Coefficients de pression des profils d'aile pour différentes génération.....	100
Figure 4.15: Variation du coefficient de traînée vs. le nombre de génération.....	101
Tableau 1.1 : Abréviations utilisées dans la figure (1.3).....	26
Tableau 3.1:Le domaine de l'espace de recherche pour profil type NACA.....	83
Tableau 3. 2 : Le domaine de l'espace de recherche pour profil type B-spline.....	83
Tableau 4.1 : Les valeurs calculées et cibles pour l'approche inverse.....	92
Tableau 4.2 : Profils d'ailes initiales type NACA.....	93
Tableau 4.3 : Les valeurs calculées et cibles.....	96

INTRODUCTION

La compétition qui régit actuellement le domaine de l'industrie aéronautique oblige les constructeurs d'avions à automatiser le plus possible leurs méthodes de conception afin de réduire le temps de développement de leurs appareils. La conception des ailes est un point particulièrement délicat à cause de leur impact sur une grande partie des performances de l'avion et aussi à cause de la complexité des équations de l'aérodynamique.

Lors de la réalisation d'un profil d'aile, sa modélisation géométrique par une méthode mathématique est un élément influent sur le processus de conception. De plus, l'utilisation croissante de méthodes d'optimisation pour la conception des profils rend crucial le besoin d'une représentation géométrique précise et nécessitant peu de paramètres.

Les méthodes d'optimisation des problèmes de technologie ont été étudiées pendant plusieurs années. L'approche générale la plus répandue est le calcul du gradient. Leur emploi a souvent connu des succès, ces méthodes disposent de nombreuses qualités : preuve de convergence, taux de convergence super linéaire possible (faible nombre d'évaluations nécessaire), faible dépendance vis-à-vis de la dimension du problème n , prise en compte de contraintes complexes. Néanmoins, plusieurs difficultés limitent l'usage de ces méthodes :

- Pour certains problèmes, la fonction de coût est non différentiable par nature ou présente des discontinuités (par exemple les problèmes de type Min Max). Dès lors, ces méthodes ne peuvent pas être employées sans aménagement spécifique.
- Le calcul du gradient de la fonction de coût peut être délicat.

- Approximation par différences finies [1] qui dépend du pas de la discrétisation. Un pas trop grand génère une erreur de troncature et un pas trop petit entraîne une erreur d'arrondi.
- Résolution du problème adjoint [2] Cette approche est limitée car elle nécessite l'écriture d'un algorithme dérivé de celui résolvant les équations d'état, dont le développement se révèle au mieux coûteux en temps humain et au pire délicat à réaliser pour des problèmes complexes.
- Calcul par l'utilisation d'outils de différentiation automatique [3]. Cette approche, consistant à générer automatiquement un code évaluant la différentielle de la fonction de coût est expéditive, mais est encore limitée à des problèmes (et des codes de calcul) modérément complexes.
- La fonction de coût peut présenter de multiples minima locaux. Or les méthodes classiques ne sont pas capables de détecter et éviter ceux-ci. La convergence vers le premier minimum trouvé, peut être caractérisée par une valeur de la fonction de coût médiocre, en comparaison du minimum absolu.

C'est pour ces raisons qu'on a recours à l'utilisation des méthodes de recherche aléatoire dans l'espace de recherche comme les algorithmes génétiques qui est la méthode la plus connue des algorithmes évolutionnaires. Elle n'exige pas un espace régulier. L'idée de base de l'approche des algorithmes génétiques est de rechercher les solutions optimales en utilisant une analogie à la théorie d'évolution de Darwin « la survie au plus adapté ». Pendant l'itération de solution (ou l'"évolution" employant la terminologie des algorithmes génétiques), les variables de décision ou les "gènes" sont manipulés en utilisant divers opérateurs (sélection, combinaison, croisement et mutation) pour créer une nouvelle génération.

L'algorithme génétique est une méthode d'optimisation semi statistique, développée dans les années 70 et qui repose sur les principes de l'évolution naturelle. Cette méthode connaît aujourd'hui un grand regain d'intérêt pour son efficacité dans les problèmes non linéaires se prêtant mal aux méthodes aux gradients. C'est pourquoi on trouve depuis peu de nombreux ouvrages traitant du sujet [4] [5] [3]. La méthode consiste à utiliser une population

d'individus, chaque individu étant un vecteur de design, et à simuler l'évolution par sélection avec cette population. Partant d'une population initiale tirée aléatoirement dans le domaine de design ou autour d'une solution de départ, on la fait évoluer en itérant les étapes des opérateurs des algorithmes génétiques [6].

Les contraintes peuvent facilement être incluses dans cette approche dans la fonction fitness. Par exemple, si un design viole une contrainte, sa fitness est mise à zéro, c à d, il ne survit pas à la prochaine génération.

Une des caractéristiques principaux algorithmes génétiques est qu'il recherche simultanément des points multiples dans l'espace de recherche, au lieu de se déplacer d'un seul point comme dans les méthodes aux gradients. En outre, ces méthodes travaillent seules aux évaluations de la fonction fitness et n'exigent pas des dérivés ou des gradients de la fonction objective. Ces caractéristiques permettent les avantages telles que la robustesse, la convenance au calcul parallèle et la simplicité en les couplant avec d'autres codes. En raison de ces avantages par rapport aux méthodes analytiques, les algorithmes génétiques sont devenus de plus en plus populaires dans une grande proportion des problèmes de conception.

Il existe principalement deux méthodes permettant l'optimisation de profils d'aile : la méthode directe et la méthode inverse. La première consiste à modifier la géométrie du profil d'aile afin d'améliorer ses caractéristiques globales que la portance ou la traînée qui sont calculées par une résolution de l'écoulement autour du profil. La deuxième (la méthode inverse) consiste quant à elle à fixer les performances exigées par profil d'aile et calcule la géométrie du profil qui produit ces performances. Elle est donc souvent utilisée comme première étape dans la conception d'une aile [7] ou comme méthode de validation avant l'utilisation de la méthode directe [8].

Les solutions d'écoulement autour du profil sont obtenues en utilisant la méthode des panneaux de Hess et Smith couplée avec un modèle de couche limite. La couche limite permet de prendre en compte les effets visqueux qui sont responsable de la précision de la traînée. La méthode est examinée pour plusieurs tailles de population et paramètres. Des résultats sont obtenus en utilisant les paramètres optima du programme avec une population initiale

aléatoirement choisie parmi les profils connus de NACA à 4 chiffres et des profils définis des courbes de B-spline qui permettent de représenter une grande famille de profils et donne tout un éventuel de profils d'aile.

Cette étude présente, dans le premier chapitre, une description des méthodes d'optimisation usuelles avec une recherche bibliographique en ce qui concerne les travaux d'optimisation de forme. Dans le deuxième chapitre, les algorithmes génétiques sont présentés. L'implémentation et l'application des algorithmes génétiques pour optimiser un profil d'aile est l'objet du troisième chapitre ensuite au quatrième chapitre les résultats obtenus sont commentés et enfin une conclusion.

CHAPITRE 1

OPTIMISATION DE FORME EN AERODYNAMIQUE : METHODES ET ETAT DE L'ART.

Dans ce premier chapitre, nous introduisons les différentes techniques et méthodes utilisées dans la résolution des problèmes d'optimisation. Ensuite nous présentons une synthèse bibliographique concernant les travaux effectués sur l'optimisation de formes notamment en aérodynamique.

1.1 Introduction:

Lorsque l'on aborde un problème d'optimisation, il est important de définir :

- l'objectif à optimiser ;
- les paramètres sur lesquels on peut ou on doit agir ;
- les contraintes à satisfaire.

Ensuite il est important d'examiner les propriétés de la fonction objectif, de l'espace d'état et des contraintes afin d'orienter le choix de la méthode d'optimisation en tenant compte des critères suivants :

- la dimension du problème ;
- le temps dont on dispose pour effectuer le calcul ;
- la taille mémoire d'un point de l'espace d'état.

1.2. Classification des méthodes d'optimisation :

On distingue deux grandes classes de méthodes d'optimisation suivant le mode de déplacement dans l'espace d'état, à savoir :

- Les méthodes d'optimisation déterministes: se caractérisent par une exploration déterministe de l'espace d'état. En effet, sous certaines conditions, elles permettent de situer la direction de l'optimum par rapport à chacun des points de cet espace et de proche en proche, de diminuer la distance entre le point courant et l'optimum. Il existe un grand nombre de méthodes d'optimisation déterministes. Nous nous attacherons à rappeler les principales sachant que les autres méthodes ont pour la plupart été construites à l'aide de celles-ci. Les méthodes d'optimisation déterministes peuvent être Locales ou Globales.

Les Méthodes Locales : ces méthodes assurent la convergence vers l'optimum (local) le plus proche. On peut citer les méthodes d'ordre zéro (Nelder-Mead), d'ordre 1 (gradient) et d'ordre 2 (Newton, BFGS).

Les Méthodes Globales : ces méthodes, s'attachent à converger vers l'optimum global, on distingue entre autres, la Programmation Linéaire (PL), les algorithmes de Séparation-Evaluation (Branch and Bound), l'algorithme A* ainsi que les méthodes Homotopiques.

- Les méthodes d'optimisation stochastiques : toutes les méthodes précédentes ont pour point commun des déplacement déterministe dans l'espace d'état. Lorsque ce type d'exploration ne peut être envisagé (à cause des coûts de calcul qu'elle engendre), on explore aléatoirement l'espace de recherche, d'où le nom donné à ces méthodes. Parmi les méthodes d'optimisation stochastiques, on peut citer la méthode de Monté Carlo, le Recuit Simulé, la Recherche Tabou, la « Branch and Bound » stochastique, évolution artificielle, etc.. Le chapitre suivant sera entièrement consacré à une technique particulière de l'évolution artificielle à savoir les algorithmes génétiques.

Un organigramme des différentes classes d'optimisation est montré sur la figure (1.1) et une présentation des différentes méthodes d'optimisation citées est développée dans le paragraphe suivant.

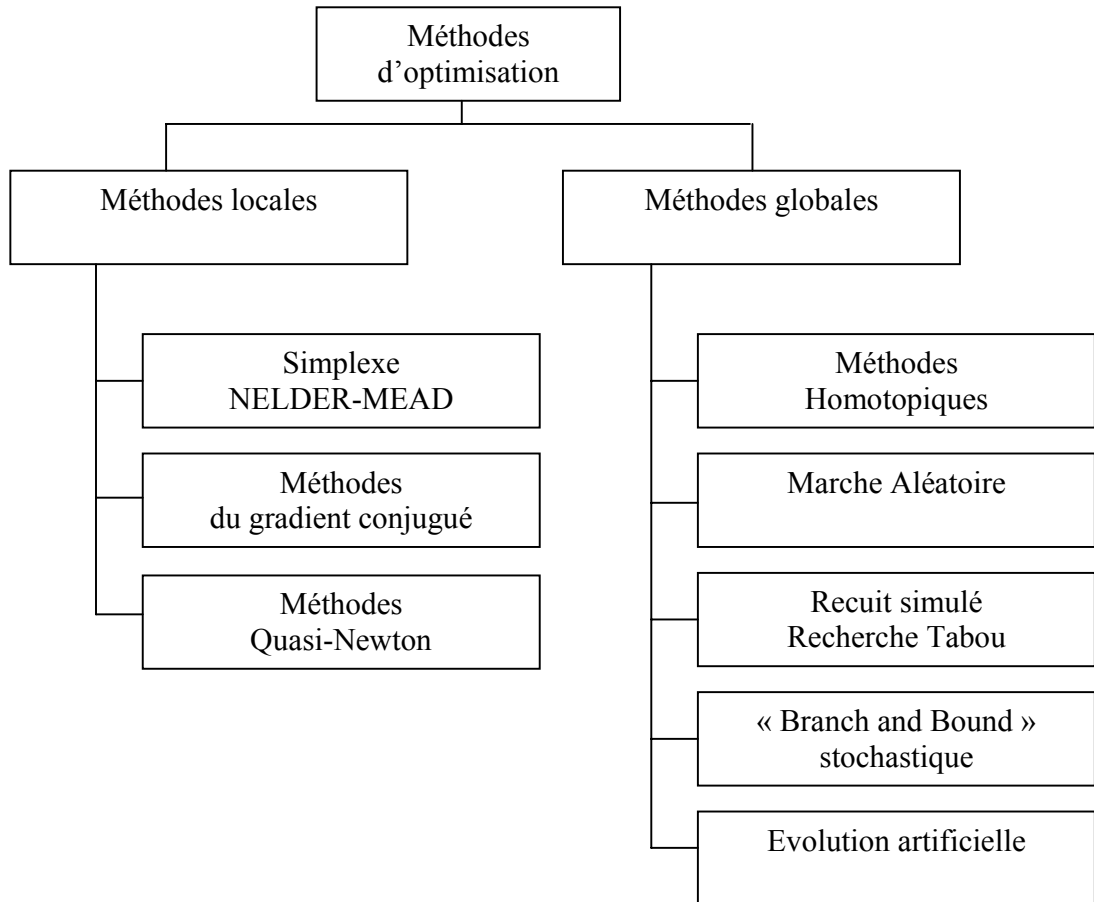


Figure 1.1. Techniques d'optimisation.

1.2.1. Optimisation déterministe locale

1.2.1.1. Algorithme de Nelder-Mead

La méthode de Nelder-Mead (*Simplex*) [9] utilise pour seule information l'espace d'état, ainsi que les valeurs du critère associé et n'utilise pas de calcul de gradient ou de hessien. Ceci présente un avantage pour certaines catégories de problèmes. On construit tout d'abord $n+1$ points de \mathfrak{R}^n définissant n vecteurs indépendants (simplexe géométrique), ensuite on modifie le simplexe obtenu par des mouvements élémentaires (réflexions, contractions, expansions et contractions massives) jusqu'à la convergence. Le processus s'arrête au moment où le déplacement donné par une des transformations devient plus petit qu'une tolérance ε prédéterminée. Ainsi à l'image d'autres méthodes géométriques, la

méthode du Simplexe n'est pas assez performante car elle utilise un nombre important d'évaluations de la fonction avant d'atteindre le point minimal.

1.2.1.2. Méthode du Gradient

Les algorithmes itératifs d'ordre 1 se basent sur la connaissance d'une direction de recherche donnée par le gradient de la fonction. Parmi les méthodes du gradient, on peut citer :

- La Méthode de la plus grande pente :

La méthode de la plus grande pente (*Steepest Descent*) [10] ou méthode du gradient à pas optimal est une des méthodes les plus classiques utilisées pour minimiser une fonction à plusieurs variables. Cette méthode d'ordre 1 est basée sur la constatation que la direction opposée à celle du gradient de la fonction représente une direction de descente. Nous pouvons donc, à partir d'un point initial x_0 , calculer la valeur du gradient et utiliser une méthode de recherche linéaire pour minimiser la fonction dans la direction de descente opposée. Cette minimisation permet de calculer la valeur du pas optimal α_k qui nous emmène à un nouveau point de recherche à chaque itération du processus, en utilisant l'équation (1.1),

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) \quad (1.1)$$

avec

x_k : point de recherche à l'itération k

x_{k+1} : nouveau point de recherche calculé à partir de la minimisation de f dans la direction opposée à son gradient.

Le processus s'arrête lorsque $\|x_{k+1} - x_k\| < \varepsilon$, ε étant une tolérance prédéterminée.

- Gradient Conjugué :

Le Gradient Conjugué [10] utilise le même principe que la méthode précédente, à la différence que la direction de descente n'est plus donnée par le gradient de la fonction, mais par des directions conjuguées successives. Deux directions h_0 et h_1 sont dites conjuguées par rapport à la matrice Hessienne H d'une fonction, Si

$$h_0^T H h_1 = 0$$

Les directions conjuguées peuvent être obtenues directement à partir du Hessien de la fonction objectif. Cependant, ce calcul peut représenter un coût important pour la méthode

d'optimisation. Pour éviter ce problème, la méthode du Gradient Conjugué effectue le calcul des directions conjuguées à partir des équations (1.2) et (1.3).

$$h_{k+1} = g_{k+1} + \beta_k h_k \quad (1.2)$$

$$\beta_k = \frac{g_{k+1} \cdot g_{k+1}}{g_k \cdot g_k} \quad (1.3)$$

où

h_k et h_{k+1} sont des directions conjuguées.

g_k et g_{k+1} sont les directions opposées aux gradients calculés aux points x_k et x_{k+1} , respectivement.

Ce calcul est uniquement valable si le point x_{k+1} est obtenu à partir d'une minimisation linéaire :

$$x_{k+1} = x_k + \alpha_k h_k \quad (1.4)$$

où

α_k est le pas optimal donné par une minimisation linéaire.

h_k représente la direction de recherche

1.2.1.3. Méthodes d'ordre deux

- Méthode de Newton

La méthode de Newton [10] utilise un modèle quadratique de la fonction à optimiser en prenant en compte son Hessien. La structure de cet algorithme est la même que celle de l'algorithme du gradient à l'exception de la direction de la recherche qui est calculée de la façon suivante : $d_N = -(f''(X_N))^{-1} \cdot f'(X_N)$ avec $f''(X_N)$ le hessien de la fonction f en X_N . Cet algorithme converge en une seule itération sur des formes quadratiques définies positives. La convergence globale n'est cependant pas assurée. De plus, il est possible que son hessien ne soit pas inversible en certains points. Le calcul de la direction de recherche nécessite une inversion de la matrice, ce qui est coûteux en temps de calcul, d'autant que la complexité de l'algorithme croît comme le cube de la dimension du problème à traiter.

- Méthode à métrique variable

La méthode à métrique variable qui s'appelle aussi méthode quasi-Newtonienne [10], utilise aussi le modèle quadratique qui n'est pas calculé à partir du hessien, mais approché par une suite de matrices définies positives (la limite de cette suite à l'optimum est le hessien en ce point). Pour une forme quadratique définie positive de dimension N , la méthode converge en $N+1$ itérations. Les algorithmes quasi-Newtoniens les plus utilisés sont l'algorithme BFGS (Broyden Fletcher Golfard Shanno) ou sa variante LM-BFGS pour les problèmes de grande taille. La convergence est beaucoup plus rapide que celle de l'algorithme du gradient. Une variante avec contraintes utilise le cône des directions admissibles en un point pour rediriger les directions de recherche produites par l'algorithme BFGS. Elle nécessite à chaque itération la résolution d'un problème de minimisation quadratique sous contraintes linéaires.

1.2.2. Optimisation déterministe globale

1.2.2.1. Programmation linéaire

La programmation linéaire [11] est la branche des mathématiques consacrée à l'étude des problèmes linéaires et particulièrement aux conditions d'existence de solutions optimales. Elle regroupe l'ensemble des techniques permettant la résolution et l'analyse des problèmes linéaires et englobe toutes les techniques d'optimisation d'une fonction linéaire, composée de n variables de décision, tout en respectant un ensemble de contraintes linéaires prenant la forme d'équations ou d'inéquations.

Comme son nom l'indique, la programmation linéaire nécessite une modélisation linéaire du problème. Celle-ci est souvent impossible ou très difficile à réaliser pour les problèmes réels qui sont majoritairement non-linéaires.

1.2.2.2. Algorithmes de Séparation–Evaluation (Branch and Bound)

La technique de « Branch and Bound » [12] consiste à découper l'espace initial de recherche en domaines de plus en plus restreints afin d'isoler l'optimum global. Une borne inf (ou sup.) du critère est associée à chacun des domaines afin de guider la recherche sans exploration exhaustive de ces derniers. L'algorithme crée un arbre de recherche dont chacun des nœuds représente un sous domaine. Celui-ci est alors évalué par son minorant : si ce

minorant est inférieur à l'évaluation d'une solution déjà atteinte, la recherche à partir du nœud est arrêtée. Cet arbre qui évolue au gré de la recherche de l'algorithme consiste à parcourir dynamiquement ce dernier jusqu'à l'obtention de l'optimum global.

1.2.2.3. Méthodes Homotopiques

Les méthodes homotopiques [13] sont des méthodes déterministes d'optimisation globale utilisant des déformations continues d'un problème initial unimodal vers le problème objectif (celui que l'on veut résoudre).

1.2.2.4. Algorithme A*

Il a été développé en 1968 en vue de combiner des méthodes heuristiques (qui utilisent des informations spécifiques au problème traité) et des méthodes formelles. L'Algorithme A*[14] optimise une distance sur un graphe, et comme la plupart des algorithmes de recherche dans les graphes, il utilise une fonction heuristique qui estime la distance entre un point quelconque et le nœud à atteindre. L'heuristique permet d'explorer la meilleure érection au premier. Si celle-ci échoue, les autres chemins peuvent être explorés. L'A* est probablement le meilleur algorithme pour la recherche du chemin le plus court sur un graphe de grande taille et utilise des " backtrack " (retour en arrière pour prendre un autre route délaissée avant).

A* explore en priorité le chemin de coût minimum. Dès lors que d'autres chemins sont explorés le coût de l'exploration de celui-ci devient excessif. Dans l'A*, on sauvegarde les branches non explorées dans une liste, mais on les ignore dès que le but est atteint.

1.2.3. Méthodes d'optimisation stochastiques

1.2.3.1. Méthode de Monte-Carlo

La méthode stochastique la plus simple est la méthode de Monte Carlo [15] dont les étapes sont les suivantes :

1. générer aléatoirement un point X dans l'espace d'état ;
2. générer un point X' (suivant une loi uniforme dans l'espace de recherche) ;
3. si X est meilleur que X' ; donc $X \rightarrow X'$;
4. si le critère d'arrêt n'est pas satisfait donc aller à l'étape 2.

Cette méthode assure la convergence vers l'optimum global à condition que la structure du voisinage dans l'étape 2 permette de couvrir l'ensemble de l'espace d'état. Cette méthode très lente a été améliorée par le recuit simulé.

1.2.3.2. Le Recuit simulé

Le Recuit Simulé (*Simulated Annealing*) [16] est basé sur le processus de recuit utilisé en métallurgie, dans lequel on cherche à obtenir un matériau sans impuretés, représenté par son état d'énergie minimale. Dans le processus de recuit en métallurgie, nous élevons la température du matériau jusqu'à ce qu'il se trouve dans un état d'énergie élevée. Ensuite, nous le refroidissons très lentement de façon à obtenir, à la fin du processus, un matériau constitué par des atomes bien ordonnés, correspondant à une valeur d'énergie stable et minimale.

En 1953, Metropolis a proposé un modèle qui simulait l'évolution d'une configuration d'atomes vers l'équilibre thermique. Dans ce modèle, une nouvelle configuration est obtenue à partir d'une petite perturbation de la configuration courante. Cette nouvelle configuration est acceptée avec une probabilité $p = 1$ lorsque la différence d'énergie ΔE entre elle et la configuration courante est inférieure à 0. Dans le cas où $\Delta E > 0$, la probabilité d'acceptation p est donnée par une équation basée sur la *Loi de Boltzmann* suivant :

$$p = e^{\frac{-\Delta E}{T}} \quad (1.5)$$

où T : est la température du système.

Kirkpatrick [16] a transposé ce modèle proposé par Metropolis à la méthode du Recuit Simulé à partir d'une correspondance entre l'énergie du système et la fonction objectif à minimiser. Dans cette transposition, les différentes configurations d'atomes sont représentées par les paramètres du problème d'optimisation, tandis que la température du système est représentée par une variable de contrôle T .

L'algorithme proposé par Kirkpatrick commence avec une configuration de paramètres choisie au hasard et une température initiale T_0 élevée. Ensuite, à l'aide d'une transformation

de voisinage faite à partir d'une petite perturbation aléatoire des paramètres, une nouvelle configuration est générée. La fonction est donc évaluée sur ces deux configurations, ce qui nous permet de calculer l'écart ΔE entre les deux évaluations. Si ΔE est inférieur à zéro, nous remplaçons la configuration originale par la nouvelle configuration obtenue. Dans le cas contraire, nous considérons la probabilité donnée par (1.5) pour décider si la configuration initiale doit être remplacée ou pas.

A chaque itération de la méthode, ce processus est répété jusqu'à ce que nous obtenions l'équilibre thermique. Au cours de cette transition vers l'équilibre, l'énergie peut localement augmenter, ce qui nous permet de sortir d'un minimum local. Ensuite, nous baissons la température du système et nous recommençons le processus. L'algorithme s'arrête lorsque nous n'avons plus d'amélioration sensible de la solution ou lorsqu'une certaine valeur de température est atteinte

1.2.3.3. Recherche Tabou

La recherche Tabou (*tabu search*) [17] permet d'atteindre le minimum global d'un problème d'optimisation à partir d'une analogie avec la mémoire du cerveau humain. Le processus de résolution développé par cette méthode commence avec la génération d'une configuration de paramètres au hasard. Ensuite, à chaque itération, le voisinage de la configuration courante est parcouru par une série de mouvements aléatoires de façon à trouver une meilleure solution. Après son exécution, chacun de ces mouvements est ajouté à une liste qui représente la "mémoire" de la méthode. Cette liste, de taille limitée, est dénommée liste Tabou.

Les mouvements qui font partie de la liste Tabou sont considérés comme interdits. C'est-à-dire qu'ils ne peuvent pas être exécutés une autre fois tant qu'ils sont dans liste. Par contre, s'il existe un mouvement qui appartient à la liste Tabou mais qui en même temps nous emmène à une meilleure solution du problème, une nouvelle exécution de ce mouvement sera alors acceptée. Dans ce cas-là, la nouvelle solution obtenue remplace la solution courante et le processus recommence.

Si pendant le processus d'optimisation la liste Tabou devient pleine, nous retirons le plus ancien mouvement de la liste avant d'y ajouter un nouveau. L'algorithme s'arrête lorsque nous n'avons plus d'amélioration sensible de la valeur de la solution.

1.2.3.4. Méthode « Branch and Bound » stochastique.

La « Branch and Bound » stochastique [18] est une technique d'optimisation globale qui utilise les intervalles de confiance d'une fonction. Elle permet de déterminer un domaine dans lequel la probabilité de présence du maximum est connue. C'est aussi, le seul algorithme stochastique permettant de fixer le niveau de risque sur le résultat. Un autre avantage est qu'il se termine en un temps fini.

Il est cependant limité, en général, à un nombre de variables inférieur à 100 et se retrouve ainsi souvent utilisé pour réduire l'espace de recherche d'une autre méthode d'optimisation. A chaque étape de l'algorithme, une partition de l'ensemble des solutions réalisables est effectuée. Pour chaque sous-ensemble, des estimations stochastiques de la borne supérieure ou inférieure de la fonction objective sont calculées et utilisées pour générer de nouveaux sous-ensembles. Au même titre que le théorème central limite décrivant la loi asymptotique de la moyenne d'un échantillon aléatoire (loi normale), il existe une loi asymptotique du maximum d'un échantillon. Elle est basée sur l'élimination progressive des zones à faible probabilité de présence du maximum global, jusqu'à ce que subsiste des domaines dans lesquels le ou les maxima se trouvent avec une probabilité donnée. La méthode est appliquée pour les problèmes d'optimisation mono-objectif.

1.2.3.5. Evolutions artificielles

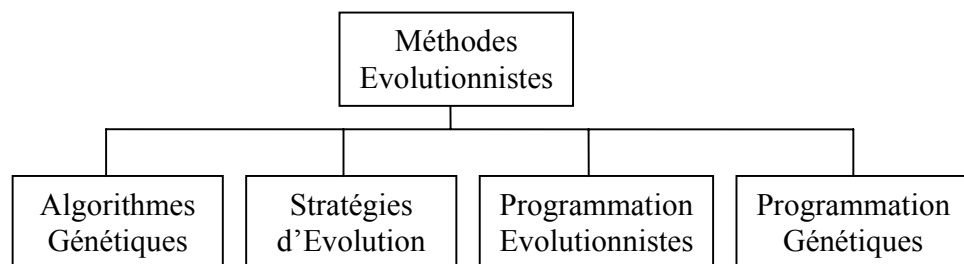


Figure 1.2. Méthodes Evolutionnistes.

Les Méthodes Évolutionnistes présentées sur la figure (1.2) regroupent différents algorithmes basés sur le même principe d'exploration de l'espace de recherche en utilisant un ensemble de solutions et pas seulement une solution unique. Comme représentantes des méthodes évolutionnistes, nous avons les algorithmes génétiques, les stratégies d'évolution [19], la programmation évolutionniste [20] et la programmation génétique [21].

Le chapitre suivant est entièrement consacré aux algorithmes génétiques, les algorithmes évolutionnaires sont des techniques de recherches inspirées par l'évolution biologique des espèces, apparues dans les années 1960, les algorithmes génétiques en constituent certainement l'exemple le plus connu. Les principes de base de ces algorithmes sont une transposition informatique simplifiée de la très célèbre théorie de Darwin. En d'autres termes, on imite au sein d'un programme la capacité d'une population d'organismes vivants à s'adapter à son environnement à l'aide de mécanismes de sélection et d'héritage génétique.

	Lin	N lin	Cont	Disc	L Di	GLO	Mu M	Mu O
PL	●		●					
PNL	●	●	●					
B & B	●	●	●	●		●		
TAB	●	●	●	●	●	●		
RS	●	●	●	●	●	●		
BBS	●	●	●	●	●	●		
HOM	●	●	●	●	●	●		
EA	●	●	●	●	●	●	●	●

Figure 1.3 : Méthodes d'optimisation vs efficacité.

Tableau 1.1 : Abréviations utilisées dans la figure (1.3)

Méthodes	Signification
PL : Programmation Linéaire;	Lin : Linéaire;
PNL : Programmation Non Linéaire;	Nlin : Non Linéaire;
B & B : Branch & Bound;	Cont : Continue;
TAB : Recherche Tabou;	Disc : Discrète;
RS : Recuit Simulé;	Ldi : Grande Dimension;
BBS : Branch and Bound Stochastique;	Glo : Globale;
HOM : Méthodes Homotopiques;	Mu M : Multi Mode;
EA : Evolution artificielle.	Mu O : Multi Objectifs ;

La figure (1.3) donne un résumé de ces méthodes et leur domaine d'application ; le cercle gris sur la figure et d'autant plus grand que la méthode est efficace pour traiter une classe de problèmes.

1.3. Etat de l'art de l'optimisation de forme en aérodynamique

L'optimisation de formes en aérodynamique représente une préoccupation technique importante dans l'industrie aéronautique depuis de nombreuses années. Les techniques d'optimisation peuvent être classifiées en trois catégories différentes : méthodes locales, globales ou autres. Les méthodes locales sont des algorithmes à bases de gradient, qui recherchent seulement une part de l'espace d'état et s'arrêtent lorsqu'un optimum local est obtenu. L'Adjoint, les pré-conditionneurs simples ou multi maillages, les méthodes implicites des directions alternatives sont des méthodes locales. Les méthodes globales sont des méthodes stochastiques, qui prennent en compte l'espace de recherche entier. Les algorithmes évolutionnaires, le recuit simulé, les méthodes aléatoires de recherche sont considérés comme des méthodes globales. Ils ont également l'avantage de fonctionner sur les espaces de contrôle discontinus. D'autres algorithmes d'optimisation qui n'appartiennent pas à ces deux catégories sont appelées méthodes « One-Shot » ou méthodes inverses.

1.3.1. Les approches déterministes

1.3.1.1. Algorithme du simplexe :

La méthode non linéaire du simplexe n'exige pas le calcul du gradient pour produire à chaque étape d'optimisation d'une nouvelle forme. Cette méthode s'appelle aussi méthode de recherche directe. Elle a été appliquée avec succès à une grande variété de problèmes. Elle établit une représentation de la fonction objective à minimiser, basée seulement sur des évaluations de la fonction objective. L'implémentation de cette méthode pour les procédures d'optimisation est plus facile que la méthode du gradient. Elle est plus robuste et plus flexible pour des problèmes d'optimisation complexes. R. Duvigneau et M. Visonneau [1] ont développé un outil d'optimisation de forme bidimensionnel pour les écoulements incompressibles et turbulents en utilisant la méthode de simplexe de Nelder-Mead.

1.3.1.2 Les méthodes du gradient.

Pour les méthodes du gradient, la fiabilité de l'optimisation dépend de l'exactitude des gradients calculés. Il y a plusieurs manières de calculer les gradients :

- La méthode adjointe est la plus efficace pour l'évaluation de la dérivée d'une fonctionnelle, parce que le coût de calcul pour calculer les gradients ne dépend pas du nombre de variables de contrôle utilisées. Le gradient adjoint peut être calculé en utilisant une formulation discrète ou continue. Anderson et Venkatakrishnan [22] ont résolu le problème inverse et le problème de minimisation de traînée pour une profil d'aile. La formulation adjointe continue a été employée sur des maillages non structurés.

Le schéma préconditionné GMRES couplé à un solveur multi-blocs implicite a été employé pour résoudre les équations adjointes en mettant en oeuvre le code KSOPT pour l'optimisation. Nielsen et Anderson [23] avaient utilisé la méthode d'optimisation aérodynamique adjointe discrète pour les profils d'aile et les ailes à multi éléments. Ils ont employé les équations de Navier-Stokes 3D sur des maillages non structurés. Elliot et Peraire [24] ont résolu les problèmes contraint et multipoint d'optimisation de forme pour les corps tridimensionnels en utilisant la méthode BFGS. M. Nemec et D. W. Zingg [25] présentent un algorithme précis et efficace pour le calcul du gradient de la fonction objective par

l'intermédiaire de l'adjoint discret et des méthodes directes. Ils ont linéarisé les équations Navier-Stokes 2D couplées au modèle de turbulence de Spalart-Allmaras. Ils ont adopté l'approche présentée dans A. Pueyo et de D. Zingg, [26] pour résoudre les équations de sensibilité adjointes et d'écoulement en utilisant l'algorithme préconditionné GMRES exclusivement. En outre, le solveur de l'écoulement est basé sur un algorithme de Newton-Krylov où la linéarisation des équations résiduelles est réutilisée.

- Une méthode alternative pour calculer les gradients est d'employer les équations de sensibilité d'écoulement. Cette méthode est moins efficace que l'approche adjointe parce que le nombre d'équations de sensibilité à résoudre dépend du nombre de paramètres de conception utilisés. Cependant, il y a quelques avantages à considérer :
 - L'approche de sensibilité est avantageuse quand des contraintes d'écoulement sont appliquées au problème de contrôle parce que des sensibilités d'écoulement peuvent être réutilisées.
 - les sensibilités d'écoulement peuvent donner au concepteur plus de perspicacité sur la façon dont le système physique répond à différentes variables de contrôle.
 - Les équations de sensibilité peuvent être résolues en utilisant l'approche de matrice libre de GMRES, qui réduit l'utilisation de mémoire, et peut éviter des difficultés dans la linéarisation.

Edmond et al [27] présentent une méthode générale d'équation de sensibilité pour des écoulements laminaires compressibles subsoniques autour de profil d'ailes. Les équations d'écoulement et de sensibilité sont écrites en utilisant les variables primitives et résolues en utilisant la méthode des éléments finis avec adaptation de maillage. Wong et Zingg [28] résolvent l'équation de sensibilité d'écoulement en utilisant une matrice-libre GMRES en même temps que le préconditionnement de premier ordre de la matrice Jacobienne. Anderson et al [29] a exécuté l'analyse de sensibilité pour les équations de Navier-Stokes sur les maillages non structurées en utilisant des variables complexes. Ils ont signalé que les dérivés complexes de sensibilité ne sont pas enclins aux erreurs provoquées par l'annulation subtractive.

- Une approche traditionnelle pour calculer des gradients est la méthode des différences finies. Cette méthode est la plus coûteuse en termes de temps de calcul. Cependant, elle est précise et peut être facilement élargie aux conceptions multidisciplinaires et de technologie générales. Elle a été utilisée par Hicks et Henne [30] pour trouver la direction de plus grande pente (*steepest descent*) dans la conception d'aile 3-D, et par Lee et Eyi [31] dans les conceptions des profils d'aile transsoniques 2-D. La méthode des différences finies (MDF) avec des variables complexes a été explorée par Anderson et al [32] pour améliorer sa précision. Newman et al [33] ont étudié les dérivées de sensibilité multidisciplinaires en utilisant des variables complexes. Un problème aérien structural a été analysé.

Xie Lei [2] implémente l'approche de sensibilité d'écoulement aux problèmes d'optimisation en CFD. L'algorithme est examiné sur divers problèmes de conception aérodynamiques. Quoiqu'il soit très facile de mettre en application la méthode de différence finie (MDF) dans le codage de programme, ses coûts informatiques prohibitifs (résolvant des équations beaucoup de fois) incite à l'utilisation d'autres méthodes plus efficaces pour calculer des gradients réduits.

- La méthode par l'utilisation d'outils de différentiation automatique (AD) a été développée sous MATLAB, C et Fortran. Chaque opération utilisée dans les codes d'analyse (y compris le générateur de maillage et le solveur d'écoulement) est différenciée en se basant sur la règle à chaînes. C. Wauquiez [34] a résolu l'optimisation de forme des profils d'ailes à vitesse réduite en utilisant la différentiation automatique programmée sous MATLAB et comme indiqué par Taylor et Oloso [3], cette méthode a été appliquée par Green et al [35] pour calculer les dérivés des fonctionnelles aérodynamiques en modélant des paramètres à certaine turbulence. Sundaram et al [36] ont rapporté des efforts d'employer des gradients d'ADIFOR 3.0 dans des conceptions d'optimisation de forme d'avion. Taylor et Green [37] ont augmenté le mode renversé d'AD pour calculer les dérivés de second ordre d'une certaine fonctionnelle aérodynamique.

1.3.2. Les approches stochastiques

1.3.2.1 Evolutions artificielles.

Contrairement aux méthodes basées sur le gradient, les méthodes de balayage de l'espace de recherche telles que les algorithmes évolutionnaires (AEs) offrent une approche alternative avec plusieurs avantages. L'idée fondamentale liée à l'approche des algorithmes évolutionnaires est de rechercher les solutions optimales en utilisant une analogie à la théorie d'évolution. Pendant l'itération de solution (ou l'"évolution" en employant la terminologie des AEs), les variables ou les "gènes" de décision sont manipulés en utilisant divers opérateurs (sélection, combinaison, croisement ou mutation) pour créer les nouvelles populations de design, c.-à-d., de nouveaux ensembles de variables de décision.

Les algorithmes évolutionnaires (AEs) sont des algorithmes d'optimisation émergents imitant le mécanisme de l'évolution naturelle, où une population biologique évolue au gré des générations pour s'adapter à un environnement par sélection, recombinaison et mutation. Quand les AEs sont appliqués aux problèmes d'optimisation, la fonction fitness, les individus et les gènes correspondent habituellement à des valeurs de la fonction objective, des individus candidats, et des variables de conception respectivement. Un des avantages principaux des AEs est qu'ils recherchent simultanément des points multiples dans l'espace de recherche, au lieu de se déplacer d'un seul point comme les méthodes du gradient. En outre, ces méthodes travaillent seules aux évaluations de fonction fitness et n'exigent pas des dérivées ou des gradients de la fonction objective. Ces dispositifs confèrent aux algorithmes des avantages tels que la robustesse, l'adaptation au calcul parallèle, la simplicité de couplage avec d'autres codes, etc.

En raison de ces avantages par rapport aux méthodes analytiques, les algorithmes évolutionnaires sont devenus de plus en plus utilisés pour une large classe de problèmes de conception. Ils ont été également appliqués avec succès aux problèmes aérodynamiques d'optimisation de forme tels que la conception de forme d'aile [38] [39] [40], l'optimisation de profils d'aile multi-éléments [41], la forme d'aile subsonique et supersonique [42] [43], l'optimisation de profil d'aubage de compresseur [44] et conception de profil d'aubage d'une grille d'aube [45].

Il y a un intérêt croissant pour l'optimisation aérodynamique de profils d'ailes à l'aide des algorithmes évolutionnaires. Dans les approches directe et inverse, Obayashi et al [7] ont élaboré un outil basé sur des algorithmes évolutionnaires pour optimiser la distribution de pression cible pour des méthodes inverses ; une fois la distribution de pression cible obtenue, les géométries correspondants de profil ou d'aile peuvent être calculées par un code inverse couplé avec un solveur Navier-Stokes. Quagliarella et Dalla Cioppa [5] ont développé une méthode de conception aérodynamique d'aile transsonique basée sur des Algorithmes génétiques couplés à un solveur potentiel. Ils ont étudié l'efficacité de la méthode d'optimisation pour un problème avec une condition d'écoulement et deux conditions d'écoulement. Vicini et Quagliarella [46] pour l'optimisation de profil d'aile multipoint et multi-objectif. Obayashi et al [47] ont optimisé des ailes transsoniques multidisciplinaire. Les exemples d'optimisation d'aile multi-design en utilisant des solveurs d'Euler et Navier/Stokes peuvent être trouvés en Sasaki, al [48] et Oyama [49] [50]. A .Oyama et S .Obayashi [51] ont la développé un outil d'optimisation fiable et efficace pour des problèmes d'optimisation de forme des aubages de compresseur en écoulement transsoniques en utilisant un algorithme génétique (real-coded Adaptive-range : real-coded ARGGA).

Une autre approche des algorithmes évolutionnaires est celle de la stratégie d'évolution, employée comme un algorithme génétique. Guido Dietz, Ralph Voß et Roeland De Breuker [52] ont présenté l'optimisation de profil d'aile basée sur une stratégie d'évolution en tenant compte de l'aéroélasticité, ses objectifs sont la minimisation de la traînée et du temps de propagation du son.

Un version très récente est appliqué avec succès, il s'agit parallélisation de Algorithme génétique est appliquées pour l'optimisation de profil d'aile et à l'optimisation d'aile [53]. Falco et Balio [54] présentent un algorithme génétique parallèle pour l'optimisation de profils d'aile transsoniques.

1.3.2.2 L'algorithme du recuit simulé.

Une autre méthode stochastique employée est le recuit simulé (simulated annealing : SA). Elle s'appuie sur une analogie avec le phénomène de cristallisation en métallurgie. Ce processus alterne des cycles de refroidissement lent et de réchauffage (recuit) qui tendent à minimiser l'énergie du matériau.

La méthode de base SA permet d'optimiser des problèmes complexes. X. Wang et M. Damodaran [55] optimisent la forme aérodynamique en recourant à la CFD en parallèle avec un algorithme de recuit simulé, qui est employé pour évaluer les fonctions objectives. Ils abordent la question du développement de la méthode en utilisant un schéma SA parallèle (PSA) avec les équations d'Euler ou de Navier-Stokes qui sont employées pour calculer les fonctions objectives. Une variété de problèmes aérodynamiques de conception de forme telles que l'approche inverse et directe ont été présentées. Ils ont étudié aussi les problèmes de l'implémentation de l'algorithme de PSA sur un ordinateur parallèle. S. Ziaei-Ziaei-Rad et A.R. Pishavar [8] ont optimisé la conception d'un profil d'aile en utilisant un recuit simulé adaptatif (ASA) et des algorithmes génétiques. L'inconvénient principal de SA est l'énorme dépense en terme de temps de calcul pour des évaluations de fonction objective dans le cas des problèmes à grande échelle.

1.3.2.3 Les algorithmes d'essaim

Ils sont basés sur une analogie avec le comportement collectif d'un groupe d'insectes. Chaque membre du groupe tend à trouver la meilleure source de nourriture, compte tenu de sa mémoire et des communications venant des autres membres.

1.3.3 Les méthodes hybrides

H. Lièvre , J.A. Désidéri , A. Habbal [56] présentent une série d'expériences numériques visant à résoudre un problème d'optimisation de formes en aérodynamique. Ils utilisent la théorie de Nash couplé avec un algorithme génétique pour trouver le meilleur compromis entre un profil adapté au régime de croisière (profil à faible traînée en transonique à faible incidence) et un profil adapté au décollage (profil à grande portance en subsonique et grande incidence).

Vicini et Quagliarella [57] ont présenté des algorithmes d'optimisation hybrides obtenus en couplant une technique de calcul de gradient avec un algorithme génétique multi-objectif dans un but de conception de forme pour des problèmes aérodynamiques, y compris l'aile et le profil d'aile.

Uelschen et Lawerenz [44] ont accéléré la méthode des algorithmes génétiques seuls par l'utilisation de réseaux de neurones fixes. Ils ont utilisé un mode d'évaluation approché s'appuyant sur les réseaux de neurones de type RBF (*Radial Basis Function*). Ces réseaux nécessitent un choix de paramètres qui conditionnent la qualité de l'approximation de la fonction et, par conséquent, la qualité de la convergence de l'algorithme d'optimisation vers l'optimum à un niveau souhaité. L'utilisation des réseaux de neurones dans ce type de couplage s'est faite au niveau de la phase d'évaluation des Algorithmes génétiques dans le but de conception de profil d'aubage de compresseur.

1.3.4. Méthodes diverses

1.3.4.1. One-Shot method

On peut également utiliser l'approche de résolution simultanée "One-Shot" pour résoudre simultanément le problème d'optimisation et la résolution des équations de mécanique des fluides associées. Cette approche est intéressante car on peut obtenir la solution optimale pour un coût inférieur à N analyses, N étant le nombre de paramètres de contrôle, ce qui permet d'utiliser des analyses de plus en plus coûteuses et d'augmenter le nombre de paramètres de contrôle. Marco et Beux. [58], Kuruvila et Shlomo [59] et Christophe Held [60] grâce à l'utilisation de méthode de la résolution simultanée ("One-Shot") présentent une optimisation de formes en aérodynamique. Pourtant bien que ces méthodes soient de plus en plus utilisées, elles ne sont pas toujours pratiques. Il existe donc toujours un besoin de développer des approches "sans gradient" efficaces.

CHAPITRE 2

LES ALGORITHMES GENETIQUES

2.1. Principes généraux

Les algorithmes génétiques sont des algorithmes d'optimisation s'appuyant sur des techniques dérivées de la génétique et de l'évolution naturelle : croisements, mutations, sélection, etc. Les algorithmes génétiques datent des premiers travaux de John Holland sur les systèmes adaptatifs remontent à 1962 [61]. L'ouvrage de David Goldberg [62] a largement contribué à les vulgariser.

Un algorithme génétique recherche le où les extrema d'une fonction définie sur un espace de données. Pour l'utiliser, on doit disposer des cinq éléments suivants :

1. Un principe de codage de l'élément de population : cette étape associe à chacun des points de l'espace d'état une structure de données. Elle se place généralement après une phase de modélisation mathématique du problème traité. La qualité du codage des données conditionne le succès des algorithmes génétiques. Le codage binaire a été très utilisé à l'origine. Le codage réel est désormais largement utilisé, notamment dans les domaines applicatifs pour l'optimisation de problèmes à variables réelles.

2. Un mécanisme de génération de la population initiale : ce mécanisme doit être capable de produire une population d'individus non homogènes qui serviront de base pour les générations futures. Le choix de la population initiale est important pour la vitesse de convergence vers l'optimum global. Dans le cas où l'on ne connaît rien du problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche.

3. Une fonction à optimiser : celle-ci retourne une valeur réel positive appelée *fitness* ou fonction d'évaluation de l'individu.

4. Des opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace d'état : l'opérateur de croisement recompose les gènes d'individus existant dans la population alors que l'opérateur de mutation a pour but de garantir l'exploration de l'espace d'états.

5. Des paramètres de dimensionnement: tels que la taille de la population, le nombre total de générations ou le critère d'arrêt, la probabilité d'application des opérateurs de croisement et de mutation.

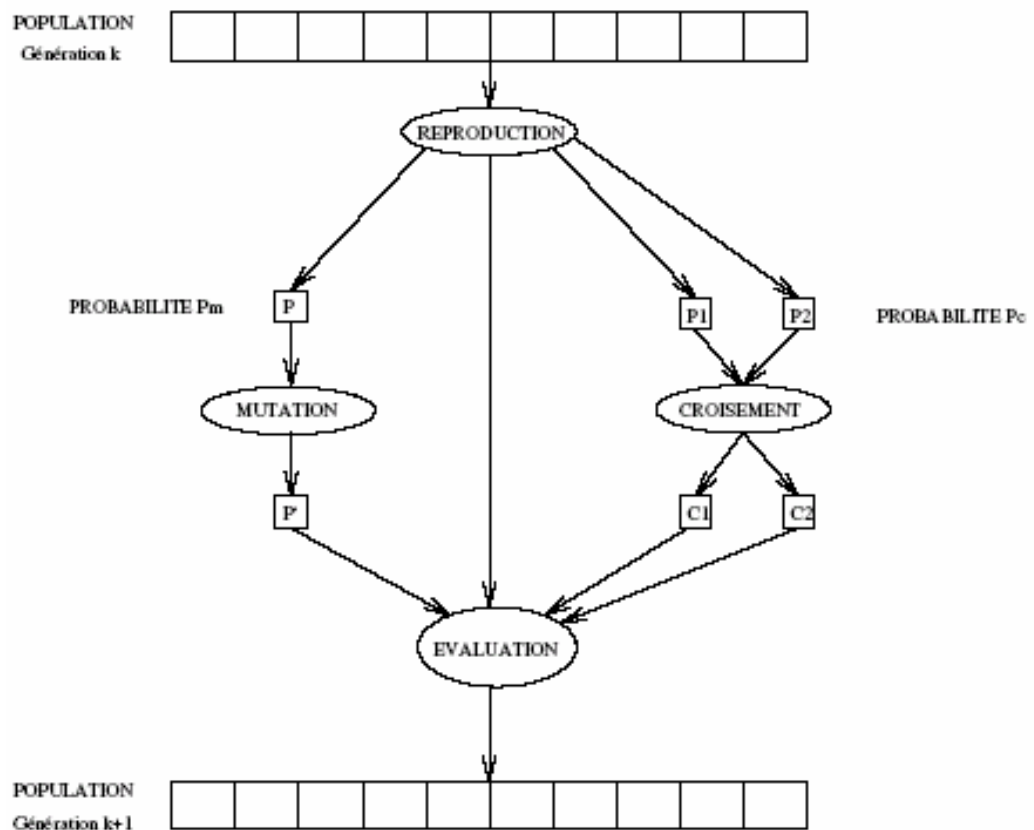


Figure 2.1: Principe général des algorithmes génétiques

Le principe général du fonctionnement d'un algorithme génétique est représenté sur la figure (2.1). on commence par générer une population d'individus de façon aléatoire. Pour passer d'une génération k à la génération $k + 1$, les trois opérations suivantes sont répétées pour tous les éléments de la population k . Des couples de parents P_1 et P_2 sont sélectionnés en fonction de leurs adaptations.

L'opérateur de croisement leur est appliqué avec une probabilité P_c (généralement autour de 0.7) et génère des couples d'enfants C_1 et C_2 . D'autres éléments P sont sélectionnés en fonction de leur adaptation. L'opérateur de mutation est appliqué avec la probabilité P_m (P_m est généralement très inférieur à 0.1) et génère des individus mutés P' . Le niveau d'adaptation des enfants (C_1, C_2) et des individus mutés P' sont ensuite évalués avant insertion dans la nouvelle population.

Différents critères d'arrêt de l'algorithme peuvent être choisis :

- Le nombre de générations que l'on souhaite exécuter peut être fixe à priori. C'est ce que l'on est tenté de faire lorsque l'on doit trouver une solution dans un temps limité.
- L'algorithme peut être arrêté lorsque la population n'évolue plus ou plus suffisamment rapidement.

Nous allons maintenant détailler chacun de ces points.

2.2. Description détaillée

2.2.1 Codage des données

Historiquement, le codage utilisé par les algorithmes génétiques était représenté sous forme de chaînes de bits contenant toute l'information nécessaire à la description d'un point dans l'espace d'état. Ce type de codage a pour intérêt de permettre de créer des opérateurs de croisement et de mutation. C'est également en utilisant ce type de codage que les premiers résultats de convergence théorique ont été obtenus. Cependant, ce type de codage n'est pas toujours bon comme le montrent les deux exemples suivants :

- Deux éléments voisins en terme de distance de Hamming ne codent pas nécessairement deux éléments proches dans l'espace de recherche. Cet inconvénient peut être évité en utilisant un codage de Gray.
- Pour des problèmes d'optimisation dans des espaces de grande dimension, le codage binaire peut rapidement devenir mauvais. Généralement, chaque variable est représentée par une partie de la chaîne de bits et la structure du problème n'est pas bien reflétée, l'ordre des variables ayant une importance dans la structure du chromosome alors qu'il n'en a pas forcément dans la structure du problème.

Les algorithmes génétiques utilisant des vecteurs réels [63] évitent ce problème en conservant les variables du problème dans le codage de l'élément de population sans passer par le codage binaire intermédiaire. La structure du problème est conservée dans le codage. Beaucoup d'autres codage en adéquation avec la structure du problème ont été expérimentés sur les algorithmes génétiques : chaînes d'entiers, de réel, graphe ...

2.2.2. Génération aléatoire de la population initiale

Le choix de la population initiale d'individus conditionne fortement la rapidité de l'algorithme. Si la position de l'optimum dans l'espace d'état est totalement inconnue, il est naturel de générer aléatoirement des individus en faisant des tirages uniformes dans chacun des domaines associés aux composantes de l'espace d'état en veillant à ce que les individus produits respectent les contraintes. Si par contre, des informations à priori sur le problème sont disponibles, il paraît bien évidemment naturel de générer les individus dans un sous domaine particulier afin d'accélérer la convergence. Dans l'hypothèse où la gestion des contraintes ne peut se faire directement, les contraintes sont généralement incluses dans le critère à optimiser sous forme de pénalités. Il est clair qu'il vaut mieux, lorsque c'est possible ne générer que des éléments de population respectant les contraintes.

2.2.3. Gestion des contraintes

Un élément de population qui viole une contrainte se verra attribuer une mauvaise fitness et aura une probabilité forte d'être éliminé par le processus de sélection. Il peut cependant être intéressant de conserver, tout en les pénalisant, les éléments non admissibles

car ils peuvent permettre de générer des éléments admissibles de bonne qualité. Pour de nombreux problèmes, l'optimum est atteint lorsque l'une au moins des contraintes de séparation est saturée, c'est à dire sur la frontière de l'espace admissible.

Gérer les contraintes en pénalisant la fonction fitness est difficile, un "dosage" s'impose pour ne pas favoriser la recherche de solutions admissibles au détriment de la recherche de l'optimum ou inversement.

Disposant d'une population d'individus non homogène, la diversité de la population doit être entretenue au cours des générations afin de parcourir le plus largement possible l'espace d'état. C'est le rôle des opérateurs de croisement et de mutation.

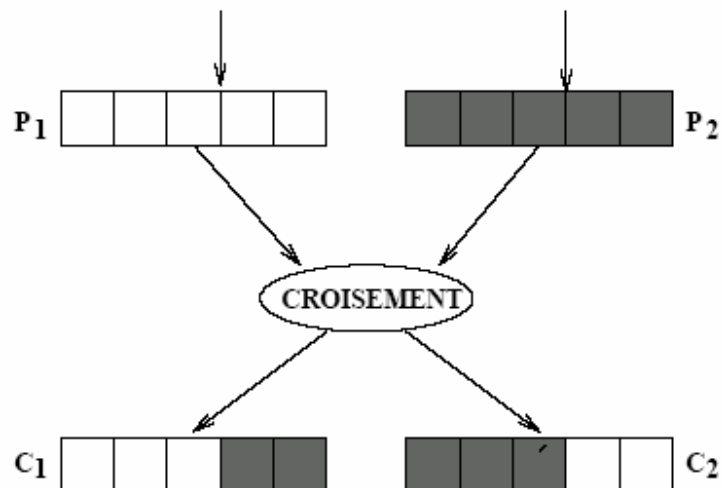


Figure.2.2 - Croisement à 1 point

2.2.4. Opérateur de Croisement

Le croisement a pour but d'enrichir la diversité de la population en manipulant la structure des chromosomes. Classiquement, les croisements sont envisagés avec deux parents et génèrent deux enfants. Initialement, le croisement associé au codage par chaînes de bits est le croisement à découpage de chromosomes (Croisement à 1 point). Pour effectuer ce type de croisement sur des chromosomes constitués de N gènes, on tire aléatoirement une position dans chacun des parents. On échange ensuite les deux sous chaînes terminales de chacun des deux chromosomes, ce qui produit deux enfants C_1 et C_2 (voir figure 2.2).

On peut étendre ce principe en découpant le chromosome non pas en 2 sous chaînes mais en 3, 4, etc. (Voir figure 2.3). Ce type de croisement à découpage de chromosomes est très efficace pour les problèmes discrets.

Pour les problèmes continus, un croisement “barycentrique” est souvent utilisé : deux gènes $P_1(i)$ et $P_2(i)$ sont sélectionnés dans chacun des parents à la même position i . Ils définissent deux nouveaux gènes $C_1(i)$ et $C_2(i)$ par combinaison linéaire :

$$\begin{cases} C_1(i) = \alpha P_1(i) + (1 - \alpha) P_2(i) \\ C_2(i) = (1 - \alpha) P_1(i) + \alpha P_2(i) \end{cases} \quad (2.1)$$

où α est un coefficient de pondération aléatoire adapté au domaine d’extension des gènes (il n’est pas nécessairement compris entre 0 et 1, il peut par exemple prendre des valeurs dans l’intervalle $[-0.5, 1.5]$ ce qui permet de générer des points entre, ou à l’extérieur des deux gènes considérés).

Dans le cas particulier d’un chromosome matriciel constitué par la concaténation de vecteurs, on peut étendre ce principe de croisement aux vecteurs constituant les gènes (voir figure 2.4) :

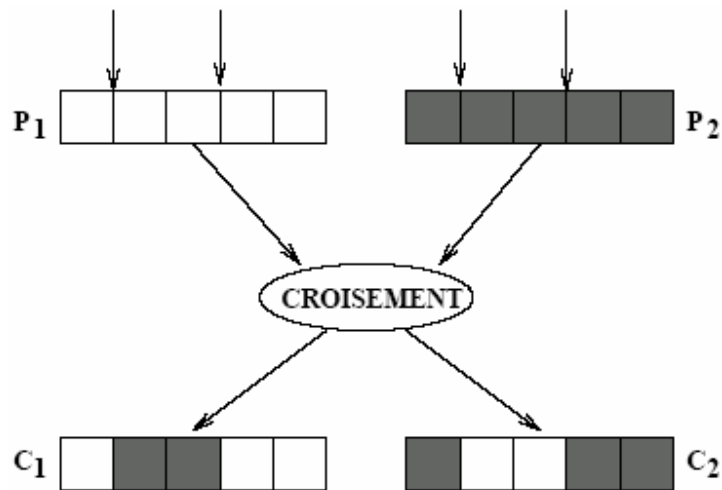


Figure 2.3: Croisement à 2 points

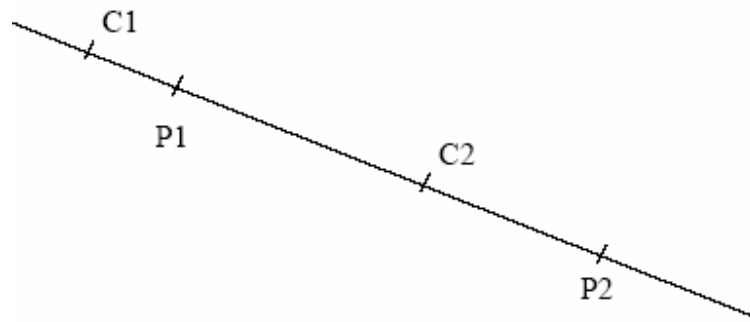


Figure 2.4: Croisement barycentrique

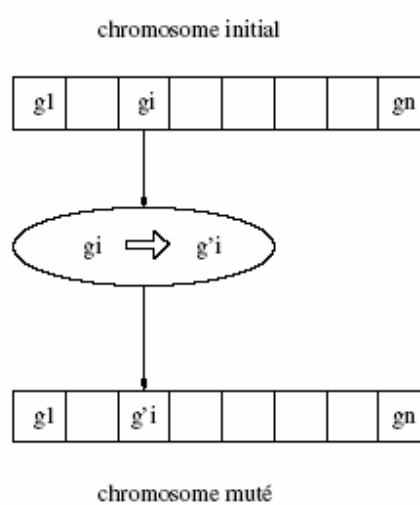


Figure 2.5: Principe de l'opérateur de mutation

$$\begin{cases} \vec{C}_1(i) = \alpha \vec{P}_1(i) + (1 - \alpha) \vec{P}_2(i) \\ \vec{C}_2(i) = (1 - \alpha) \vec{P}_1(i) + \alpha \vec{P}_2(i) \end{cases} \quad (2.2)$$

On peut imaginer et tester des opérateurs de croisement plus ou moins complexes sur un problème donné mais l'efficacité de ce dernier est souvent liée intrinsèquement au problème.

2.2.5. Opérateur de mutation

L'opérateur de mutation apporte aux algorithmes génétiques la propriété d'ergodicité de parcours d'espace. Cette propriété indique que l'algorithme génétique sera susceptible d'atteindre tous les points de l'espace d'état, sans pour autant les parcourir tous dans le

processus de résolution. Ainsi en toute rigueur, l’algorithme génétique peut converger sans croisement, et certaines implémentations fonctionnent de cette manière [64]. Les propriétés de convergence des algorithmes génétiques sont donc fortement dépendantes de cet opérateur sur le plan théorique.

Pour les problèmes discrets, l’opérateur de mutation consiste généralement à tirer aléatoirement un gène dans le chromosome et à le remplacer par une valeur aléatoire (voir figure 2.5). Si la notion de distance existe, cette valeur peut être choisie dans le voisinage de la valeur initiale.

Dans les problèmes continus, on procède un peu de la même manière en tirant aléatoirement un gène dans le chromosome, auquel on ajoute un bruit généralement gaussien (aléatoire). En veillant à ce que le gène résultant reste dans le domaine d’extension qui lui est propre.

2.2.6. Principes de sélection

A l’inverse d’autres techniques d’optimisation, les algorithmes génétiques ne requièrent pas d’hypothèse particulière sur la régularité de la fonction objective. L’algorithme génétique n’utilise notamment pas ses dérivées successives, ce qui rend très vaste son domaine d’application. Aucune hypothèse sur la continuité n’est non plus requise. Néanmoins, dans la pratique, les algorithmes génétiques sont sensibles à la régularité des fonctions qu’ils optimisent.

Le peu d’hypothèses requises permet de traiter des problèmes très complexes. La fonction à optimiser peut ainsi être le résultat d’une simulation. La sélection permet d’identifier statistiquement les meilleurs individus d’une population et d’éliminer les mauvais. On trouve dans la littérature un nombre important de principes de sélection plus ou moins adaptés aux problèmes qu’ils traitent. Les deux principes de sélection suivants sont les plus couramment utilisés :

- La roulette (*wheel selection*) [62];
- La méthode “*Stochastic remainder without replacement selection*” [62];

- Le principe de la roulette (*wheel selection*) consiste à associer à chaque individu un segment dont la longueur est proportionnelle à la valeur de *fitness*. On reproduit ici le principe de tirage aléatoire utilisé dans les roulettes de casinos avec une structure linéaire. Ces segments sont ensuite concaténés sur un axe que l'on normalise entre 0 et 1. On tire alors un nombre aléatoire de distribution uniforme entre 0 et 1, puis on "regarde" quel est le segment sélectionné. Avec ce système, les grands segments, c'est-à-dire les bons individus, seront plus souvent choisis que les petits.

Lorsque la dimension de la population est réduite, il est difficile d'obtenir en pratique l'espérance mathématique de sélection en raison du peu de tirages effectués. Un biais de sélection plus ou moins fort existe suivant la dimension de la population.

- La méthode de sélection par échantillonnage stochastique de la partie restante sans remplacement (*Stochastic remainder without replacement selection*) évite ce genre de problème et donne de bons résultats. Décrivons ce principe de sélection :
 - Pour chaque élément i , on calcule le rapport r_i de sa *fitness* sur la moyenne des *fitness*.
 - Soit $e(r_i)$ la partie entière de r_i , chaque élément est reproduit exactement $e(r_i)$ fois.
 - La roulette (*wheel selection*) précédemment décrite est appliquée sur les individus affectés des fitness $r_i - e(r_i)$. Lorsque des populations de faible taille sont utilisées, ce principe de sélection s'avère généralement efficace dans les applications pratiques.

2.3. Améliorations classiques

2.3.1. Introduction

Les processus de sélection présentés sont très sensibles aux écarts de fitness et dans certains cas, un très bon individu risque d'être reproduit trop souvent et peut même provoquer l'élimination complète de ses congénères; on obtient alors une population homogène contenant un seul type d'individu. Ainsi, dans l'exemple de la figure (2.6) le second mode M_2 risque d'être le seul représentant pour la génération suivante et seule la mutation pourra aider à atteindre l'objectif global M_1 au prix de nombreux essais successifs.

Pour éviter ce comportement, il existe d'autres modes de sélection tel que le « *ranking* » ainsi que des principes (*scaling*, *sharing*) qui empêchent les individus "forts" d'éliminer

complètement les plus “faibles”. On peut également modifier le processus de sélection en introduisant des tournois entre parents et enfants, basés sur une technique proche du recuit.

Enfin, on peut également introduire des recherches multi-objectives, en utilisant la notion de dominance lors de la sélection. Dans la littérature, cette méthode porte parfois le nom de méthode de Monte-Carlo.

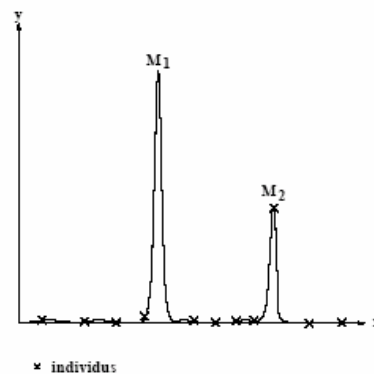


Figure 2.6: Exemple où les sélections classiques risquent de ne reproduire qu’un individu

2.3.2. Scaling

Le *Scaling* ou mise à l’échelle, modifie les fitness afin de réduire ou d’amplifier artificiellement les écarts entre les individus. Le processus de sélection n’opère plus sur la fitness réelle mais sur son image après *Scaling*. Parmi les fonctions de *Scaling*, on peut envisager le *Scaling* linéaire et le *Scaling* exponentiel. Soit f_r la fitness avant *Scaling* et f_s la fitness modifiée par le *Scaling*.

2.3.2.1. Scaling linéaire

Dans ce cas la fonction de *Scaling* est définie de la façon suivante [65] :

$$f_s = af_r + b$$

$$a = \frac{\max' - \min'}{\max - \min}; b = \frac{\min' \cdot \max - \min \cdot \max'}{\max - \min}$$

En règle générale, le coefficient a est inférieur à 1, ce qui permet de réduire les écarts de fitness et donc de favoriser l’exploration de l’espace. Ce *Scaling* est statique par rapport au

numéro de génération et pénalise la convergence lorsque l'on désire favoriser les modes dominants.

2.3.2.2. Scaling exponentiel

Il est défini de la façon suivante [65] (voir figure 2.7):

$$f_s = (f_r)^{k(n)} \quad (2.3)$$

où n est la génération courante.

- Pour k proche de zéro, on réduit fortement les écarts de fitness ; aucun individu n'est vraiment favorisé et l'algorithme génétique se comporte comme un algorithme de recherche aléatoire et permet d'explorer l'espace.

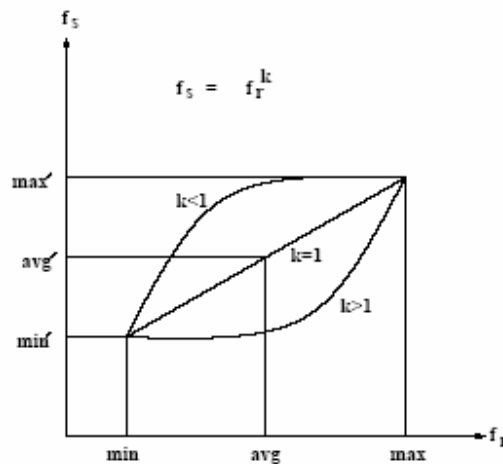


Figure 2.7: Fonction de Scaling exponentielle

- Pour k proche de 1: le scaling est inopérant.
- Pour $k > 1$ les écarts sont exagérés et seuls les bons individus sont sélectionnés ce qui produit l'émergence des modes.

Dans la pratique, on fait généralement varier k des faibles valeurs vers les fortes valeurs au cours des générations. Pour cela on peut utiliser la formule suivante :

$$k = \left(\tan \left[\left(\frac{n}{N+1} \right) \frac{\pi}{2} \right] \right)^p \quad (2.4)$$

n étant la génération courante, N le nombre total de générations prévues, p un paramètre à choisir.

Le choix de $p = 0.1$ s'est avéré pertinent dans les applications. L'évolution de k en fonction de la génération n est donnée à la (figure 2.8).

Dans le cas des fonctions objectifs multi modes présentant des optimums quasi-équivalents, cette technique de Scaling, en amplifiant les écarts de fitness en fin de convergence, va effectivement favoriser le mode dominant mais aussi masquer les modes sous optimums qui peuvent tout de même présenter un intérêt. Le Scaling permet donc une bonne exploration de l'espace d'état mais ne favorise pas la répartition des individus sur les différents modes de la fonction objective.

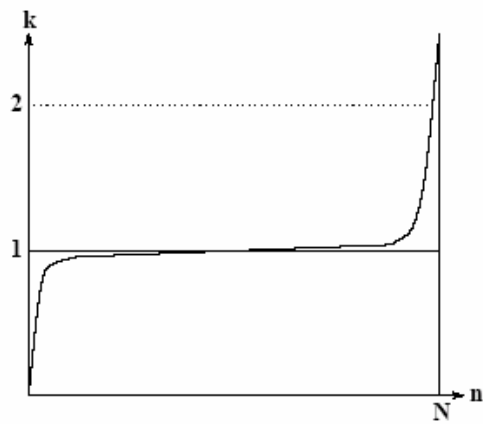


Figure 2.8: Allure de l'évolution de k en fonction des générations

2.3.3. Sharing

2.3.3.1. Introduction

L'objectif du Sharing est de répartir sur chaque sommet de la fonction à optimiser un nombre d'individus proportionnel à la fitness associée à ce sommet. La figure (2.9) présente deux exemples de répartition de population dans le cas d'une fonction à cinq sommets : le premier sans Sharing, le second avec Sharing.

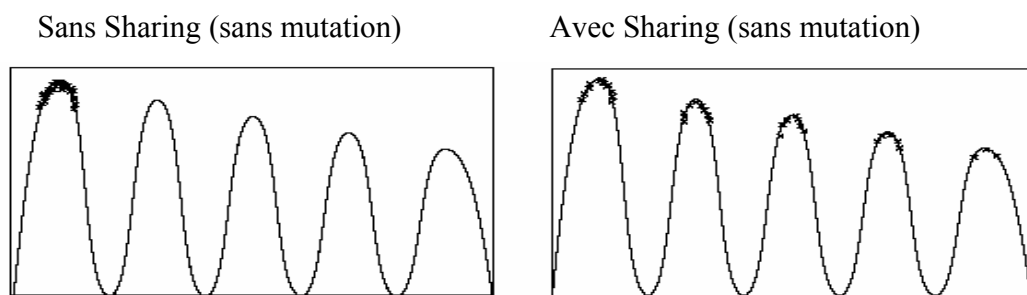


Figure 2.9: Objectif du Sharing

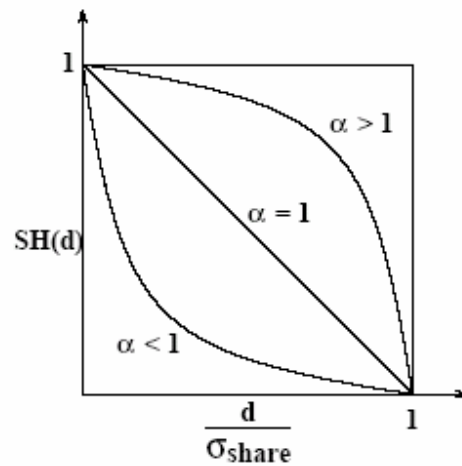


Figure 2.10: Allure de $SH\left(\frac{d}{\sigma_{share}}\right)$

2.3.3.2. Principe

De la même façon que le Scaling, le Sharing consiste à modifier la fitness utilisée par le processus de sélection. Pour éviter le rassemblement des individus autour d'un sommet dominant, le Sharing pénalise les fitness en fonction du taux d'agrégation de la population dans le voisinage d'un individu. Il requiert l'introduction d'une notion de distance. Dans la pratique, il faut définir une distance indiquant la dissimilarité entre deux individus. Cette distance est alors utilisée pour calculer la nouvelle fitness de la façon suivante :

$$f'_i = \frac{f_i}{m'_i}; \quad m'_i = \sum_{j=1}^N S(d(x_i, x_j)) \quad (2.5)$$

avec

$$S(d) = 1 - \left(\frac{d}{\sigma_{share}} \right)^\alpha \quad \text{si } d < \sigma_{share}$$

$$S(d) = 0 \quad \text{si } d > \sigma_{share}$$

Le paramètre σ_{share} permet de délimiter le voisinage d'un point et dépend du problème traité.

La figure (2.10) donne l'allure de $S(d)$ pour différentes valeurs de α . Suivant la valeur donnée à α le Sharing sera plus ou moins efficace. Ainsi pour $\alpha < 1$, on pénalise les groupes très agglomérés. Dans la pratique ce type de Sharing donne effectivement de bons résultats mais au prix de N^2 calculs de distances entre chromosomes à chaque génération pour une population de taille N . Or les algorithmes génétiques induisent une complexité en N sans Sharing et le fait de passer en N^2 peut être pénalisant, notamment pour N grand. Pour réduire ce nombre, on utilise un Sharing "clustérisé".

2.3.4. Sharing clustérisé

Pour effectuer ce type de Sharing [66], on commence par identifier les différents 'clusters' d'individus dans la population. Ce dernier utilise deux paramètres d_{min} et d_{max} respectivement pour fusionner des clusters ou en créer de nouveaux. Initialement, chaque individu de la population est considéré comme le centre d'un cluster. On applique alors successivement les deux principes suivants :

- Si deux centres sont à une distance inférieure à d_{\min} , on fusionne ces derniers dans un cluster unique dont le centre résultant est le barycentre des deux centres initiaux.
- Un nouvel individu est agrégé à un cluster si sa distance au centre le plus proche est inférieure à d_{\max} . Dans ce cas, on recalcule le centre du cluster global. Sinon, on crée un nouveau cluster contenant ce seul individu.

Ce principe de fusion-agrégation permet d'obtenir un nombre de clusters fluctuant avec la répartition des individus dans l'espace d'état. On applique ensuite le principe de Sharing en modifiant les fitness de la façon suivante :

$$f_i' = \frac{f_i}{m_i}; m_i = n_c \left(1 - \left(\frac{d_{ic}}{2d_{\max}} \right)^\alpha \right); \quad (2.6)$$

Avec

- n_c : Nombre d'individus contenus dans le cluster auquel appartient l'individu i .
- α : Coefficient de sensibilité.
- d_{ic} : Distance entre l'individu i et le centre du cluster c .

On montre que ce type de Sharing induit une complexité en $O(N \log N)$ [66] pour des résultats tout à fait comparables à ceux fournis par le Sharing classique. Dans la pratique, on remarque que le réglage des coefficients d_{\min} et d_{\max} est assez délicat car l'efficacité de ces derniers dépend essentiellement de la connaissance a priori des distances inter modes dans l'espace d'état, distance qu'il est très difficile d'estimer.

2.3.5. L'élitisme

L'élitisme a pour but de préserver à chaque génération les meilleurs éléments de la population, et ce, afin d'éviter qu'un opérateur de croisement ou de mutation ne les détruise. Après chaque génération on relève l'indice i_{meil} du meilleur élément de population. Dans les procédures croisement et mutation, lorsque cet indice i_{meil} est rencontré, un nouveau tirage aléatoire (de distribution uniforme sur la population) est effectué.

L'élitisme sera systématiquement utilisé dans les applications.

2.3.6. Algorithmes génétiques et recuit simulé

2.3.6.1. Introduction

Les algorithmes génétiques et le recuit simulé étant deux techniques d'optimisation stochastique travaillant sur les mêmes types de problèmes, il est logique de chercher à les associer afin de tirer partie de leurs avantages respectifs. Après plusieurs évaluations de ces deux techniques sur les mêmes problèmes test, on remarque que le recuit simulé converge généralement plus vite vers la solution optimale lorsque le problème est de taille raisonnable. Toutefois, il ne donne qu'une solution dans le cas des problèmes multi modes, ceci confirme les résultats donnés dans [67]. A l'inverse, les algorithmes génétiques fournissent plusieurs solutions quasi-optimales mais au prix d'un temps de convergence généralement plus long. Il semble alors naturel d'associer ces deux techniques afin d'améliorer la convergence des algorithmes génétiques.

Il y a eu de nombreuses tentatives de fusion entre les algorithmes génétiques et le recuit simulé, les travaux les plus intéressants étant ceux de Mahfoud et de Goldberg [68].

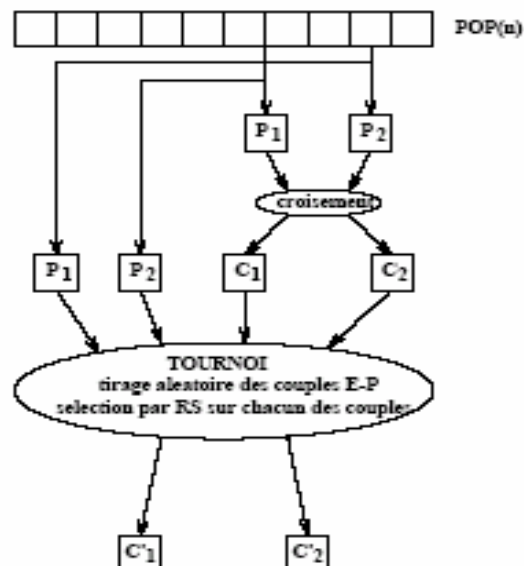


Figure 2.11: Principe du croisement avec recuit simulé

2.3.6.2. Principe du croisement avec recuit simulé

Pour appliquer ce principe de croisement, on commence par sélectionner deux parents P_1 et P_2 dans la population (voir figure 2.11). On applique ensuite l'opérateur de croisement classique qui génère deux enfants C_1 et C_2 . Un tournoi est alors effectué entre les parents et les enfants pour lequel les deux vainqueurs sont sélectionnés par le schéma de recuit suivant.

On considère l'individu C_1 . On tire ensuite aléatoirement un des deux parents, soit P_i ce parent :

- Si C_1 est meilleur que $P_i \Rightarrow C_1$ est sélectionné.
- Sinon C_1 est sélectionné avec la probabilité :

$$P = e^{-\left(\frac{|f_{C_1} - f_{P_i}|}{c_n}\right)} \quad (2.7)$$

où c_n est un coefficient décroissant en fonction de la génération courante (n).

On fait de même pour C_2 avec le parent restant et l'on détermine ainsi deux individus C'_1 et C'_2 .

L'évolution de la variable c_n se fait de la façon suivante. On utilise un schéma de recuit standard géométrique à un palier de basculement. Pratiquement, on calcule trois "températures" dont les valeurs dépendent de la connaissance des écarts *min* et *max* des fitness de la population initiale.

$$\left\{ \begin{array}{l} C_s = -\frac{\Delta f_{\max}}{\ln\left(\frac{1}{k-1}\right)} \quad k = 0.75 \quad \text{"Température initiale"} \\ C_x = -\frac{\Delta f_{\max}}{\ln\left(\frac{1}{k-1}\right)} \quad k = 0.99 \quad \text{"Température de basculement"} \\ C_f = -\frac{\Delta f_{\min}}{\ln\left(\frac{1}{k-1}\right)} \quad k = 0.99 \quad \text{"Température finale"} \end{array} \right. \quad (2.8)$$

où Δf_{\min} et Δf_{\max} représentent les écarts minimum et maximum des fitness de la population initiale.

Le schéma géométrique fait évoluer la température courante de la façon suivante :

$$\begin{cases} C_0 = 0_s \\ C_{n+1} = \alpha_1 C_n & \text{pour } C_s \rangle C_n \rangle C_x; \\ C_{n+1} = \alpha_2 C_n & \text{pour } C_x \rangle C_n \rangle C_f; \end{cases} \quad (2.9)$$

avec $0 < \alpha_1 < \alpha_2 < 1$

Pour chaque palier, on calcule le nombre d'itérations de stabilisation à l'aide des formules :

$$N_1 = \frac{\ln\left(\frac{C_x}{C_s}\right)}{\ln\alpha_1} \quad N_2 = \frac{\ln\left(\frac{C_f}{C_x}\right)}{\ln\alpha_2} \quad (2.10)$$

Ces deux formules, nous permettent de calculer le nombre total de générations pour un problème donné. Ce même principe de recuit simulé a été essayé sur l'opérateur de mutation en faisant un schéma de recuit entre l'individu muté et l'individu d'origine mais il ne produit pas l'effet escompté. En effet, on peut supposer que dans ce cas le recuit simulé réduit le brassage de la population provoqué par la mutation en limitant l'espace exploré aux zones qui améliorent statistiquement la fitness en "interdisant" les domaines qui la dégradent. L'exploration du domaine admissible est fragilisée. Ce constat sur les problèmes que nous avons testé ne permet pas de généraliser.

2.3.7. Recherche multi objectifs

2.3.7.1. Introduction

Dans le cadre de la recherche multi objectifs, on cherche à optimiser une fonction suivant plusieurs critères, dont certains peuvent d'ailleurs être antagonistes. On définit alors la classique notion de dominance : on dit que le point A domine le point B si, $\forall i, f_i(A) \geq f_i(B)$ et $\exists i, f_i(A) > f_i(B)$ où les f_i représentent les critères à maximiser. L'ensemble des points qui ne sont dominés par aucun autre point forme la surface de Pareto. Tout point de la surface de Pareto est "optimal", dans la mesure où on ne peut améliorer la valeur d'un critère pour ce point sans diminuer la valeur d'au moins un autre critère.

Les algorithmes génétiques peuvent permettre de trouver l'ensemble de la surface de Pareto, car il est possible de répartir la population de l'algorithme génétique sur ladite surface.

2.3.7.2. Technique employée

La technique employée dérive directement des travaux de [69]. Le principal changement induit concerne le processus de sélection : en multi-objectifs, comment va t-on décider qu'un individu est meilleur qu'un autre? On introduit alors une variante de la notion de dominance que l'on définit ainsi : on peut par exemple décider que l'élément E_i domine E_j si le nombre des valeurs contenues dans son vecteur d'adaptation qui sont supérieures aux valeurs correspondantes dans E_j dépasse un certain seuil. A partir de là, la technique proposée pour effectuer la sélection est simple : on tire deux individus au hasard ainsi qu'une sous population à laquelle ils n'appartiennent pas et qui va servir à les comparer.

Trois cas se présentent alors :

- Si le premier élément domine tous ceux de la sous population et que ce n'est pas le cas pour le second alors le premier sera sélectionné.
- Inversement si seul le second domine l'ensemble de la sous population alors c'est lui qui sera conservé.
- Restent maintenant deux possibilités : soit les deux sont dominés, soit les deux dominant. On ne peut se prononcer sans ajouter un autre test, c'est pourquoi dans ce cas il est fait usage d'un nouveau type de sharing qui opère sur l'espace objectif. Le sharing va conduire à sélectionner celui des deux individus qui au moins de voisins proches, autrement dit on élimine celui qui se situe dans une zone d'agrégation pour conserver celui qui est dans une région moins dense.

Encore une fois, le terme *voisin proche* n'a pas de signification précise mais il est possible de définir un voisinage (aussi appelé niche) correct en se servant de la distance de Holder :

$$d_H(E_i, E_j) = \left(\sum_{k=1}^n |f_i^k - f_j^k|^p \right)^{\frac{1}{p}} \quad (2.11)$$

f_i^k désignant la $k^{\text{ème}}$ composante du vecteur adaptation de l'élément i . Le paramètre p permet de faire varier la forme et la taille du voisinage. A l'intérieur des voisinages ainsi définis dans l'espace objectif, il suffit de compter le nombre d'individus pour favoriser les zones les moins denses et de cette façon maintenir la diversité de la population. Ainsi la figure (2.12) montre comment les voisinages sont choisis autour des individus de la région de Pareto lorsque ceux-ci ne peuvent être départagés sans sharing.

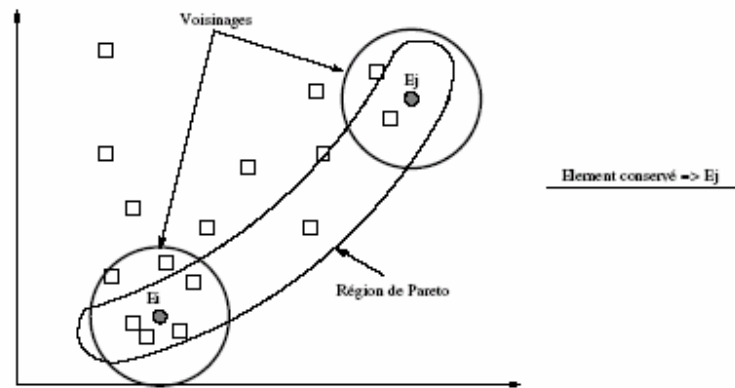


Figure 2.12: Surface de Pareto et voisinages

2.3.8. Association des AG avec des méthodes locales

La grande force des algorithmes génétiques est leur capacité à trouver la zone de l'espace des solutions contenant l'optimum de la fonction. En revanche, ils sont inefficaces lorsqu'il s'agit de trouver la valeur exacte de l'optimum dans cette zone. Or, c'est précisément ce que les algorithmes locaux d'optimisation réalisent le mieux.

Il est donc naturel de penser à associer un algorithme local à l'algorithme génétique de façon à trouver la valeur exacte de l'optimum. On peut aisément le faire en appliquant à la fin de l'algorithme génétique un algorithme local sur le meilleur élément trouvé. Cette technique est d'autant plus efficace que l'on utilise simultanément du clustering, et que l'algorithme local est appliqué à chaque meilleur élément de chaque cluster. En effet, on constate souvent que le meilleur élément trouvé par l'algorithme génétique ne reste pas le meilleur élément après amélioration par l'algorithme local de tous les meilleurs éléments de clusters.

Une autre technique consiste à utiliser un algorithme local associé à l'algorithme génétique pour calculer la fitness d'un élément. On peut, par exemple dans un espace fortement combinatoire, rechercher avec l'algorithme génétique les zones intéressantes de l'espace en générant les éléments, l'adaptation de chaque élément étant calculée par un programme d'optimisation local (linéaire par exemple).

En fait, l'association algorithmes génétiques–méthodes locales est une quasi-nécessité. Les deux méthodes sont complémentaires et ont des champs d'application différents. L'algorithme génétique permet de faire “disparaître” la combinatoire du problème, laissant alors le champ libre aux méthodes locales dans chacune des zones connexes qu'il pense susceptible de contenir l'optimum global.

2.4. Parallélisme

L'intérêt de la parallélisation des algorithmes génétiques est de gagner en temps de calcul. Il existe pour cela au moins deux méthodes utilisées classiquement (on pourra se reporter à [70] et [71] pour plus de précisions sur les modèles de parallélisme utilisables dans les algorithmes génétiques) :

- La première consiste à diviser la population de taille n en N sous populations et à les répartir sur l'ensemble des machines dont on dispose.
- La seconde maintient la population totale sur une seule machine mais se sert des autres pour y envoyer les évaluations afin qu'elles se fassent en même temps.

Dans les deux cas il est nécessaire d'avoir un mécanisme de communication interprocessus. Les résultats présentés ici ont été obtenus en utilisant PVM qui a été réalisé par le Oak Ridge National Laboratory. Il permet à un réseau hétérogène de machines d'apparaître comme une seule entité ayant plusieurs processeurs capables de communiquer entre eux (comparable à un ordinateur à mémoire distribuée). La communication entre les divers composants de cette machine virtuelle se fait par l'envoi de paquets contenant des données ou encore des messages de contrôle. Pour plus de détails, on peut se reporter à [72].

2.4.1. Parallélisme par îlots

L'idée ici est de faire fonctionner plusieurs algorithmes génétiques en parallèle avec une population réduite pour chacun. Le programme maître lance N occurrences du programme d'algorithmes génétiques appelées esclaves sur des machines différentes en passant à chacune les paramètres nécessaires à leur bon fonctionnement comme la montre la figure (2.13).

Ensuite chaque processus fait évoluer sa population indépendamment jusqu'à ce qu'il décide (selon une probabilité fixée à l'avance) de rassembler ses meilleurs individus pour en transmettre une certaine quantité (proportionnelle à la taille de la population) à une autre machine de son choix. La machine réceptrice intègre alors ces nouveaux éléments dans sa propre population en éliminant les moins bons de ses individus. L'intérêt du parallélisme par îlots est qu'il offre la possibilité de travailler sur de grandes populations (n_0) tout en donnant des résultats dans un temps raisonnable puisque la durée nécessaire est à peu de choses près

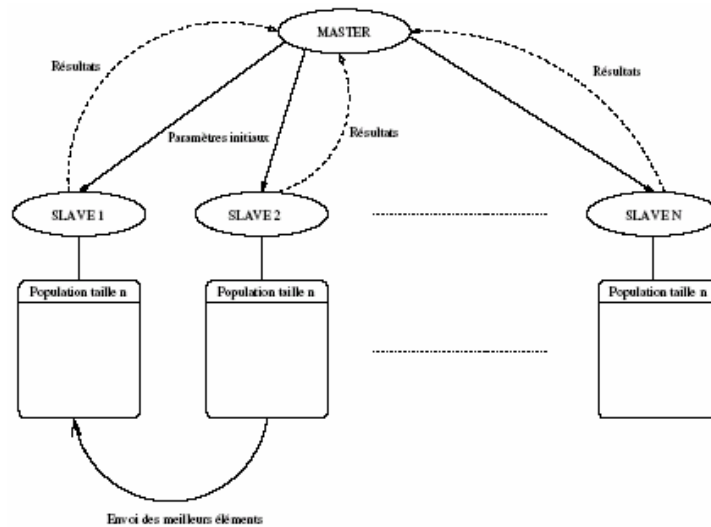


Figure 2.13: Principe de fonctionnement du parallélisme par îlots

celle qu'il faudrait pour une population de taille $\frac{n_0}{N}$, si N est le nombre d'ordinateurs disponibles et si l'on néglige les temps de communication.

La méthode introduit un clustering forcé car chaque îlot peut être considéré comme un cluster subdivisé en petits groupes. Chaque machine a la possibilité de converger vers des optima qui seront différents de ceux calculés sur les autres ce qui correspond au comportement

introduit avec le clustering. D'autre part, le surcoût de temps passé pour les communications n'est a priori pas excessif puisqu'il n'y a de gros paquets à transmettre que de temps en temps. Il faut tout de même garder à l'esprit qu'une subdivision en sous-populations de taille trop réduite risque de conduire à faire tourner des algorithmes génétiques non fiables statistiquement. En effet, il faut quand même qu'une population contienne suffisamment d'individus pour que l'espace d'état puisse être exploré de façon correcte afin que les résultats aient une certaine valeur.

2.4.2. Parallélisation des calculs

Contrairement à la méthode qui vient d'être décrite qui divise la population totale, on utilise ici des démons de calcul de fitness dont la seule fonction est de recevoir un individu et de renvoyer son adaptation. Pour utiliser la puissance de calcul parallèle offerte de manière optimale, il faut retarder au maximum les calculs pour les envoyer en blocs aux démons et faire en sorte qu'aucun ne reste inactif. Le principe se trouve résumé sur la figure (2.14): le programme maître se charge de faire la sélection, les croisements, etc. En d'autres termes il fait évoluer la population, puis répartit les calculs dont il a besoin sur l'ensemble des démons.

Enfin, dès qu'il a reçu tous les résultats, l'algorithme commence une nouvelle génération. Il faut noter que ce mécanisme demande un grand nombre de communications pour envoyer les données et les évaluations. La méthode n'est donc intéressante que si le temps passé pour un calcul d'adaptation est grand devant le temps de communication, elle sera par conséquent utilisée pour des problèmes dont les évaluations de fitness prennent du temps, on pense essentiellement à des cas faisant appel à des réseaux de neurones ou à de gros calculs matriciels.

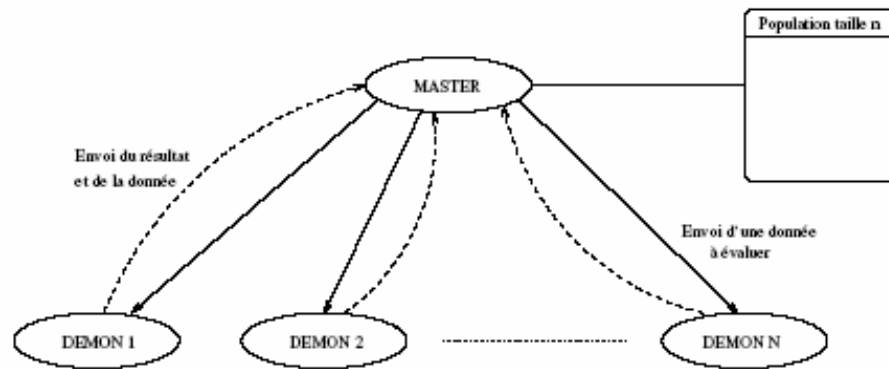


Figure 2.14 : Principe de fonctionnement de la parallélisation des calculs

2.4.3. Conclusion

La parallélisation est extrêmement efficace pour accélérer les temps de résolution des algorithmes génétiques. Il faut certes bien étudier le problème afin d'utiliser le bon mécanisme de parallélisme, mais les gains en temps sont alors importants.

2.5. Résultats de convergence théorique des algorithmes génétiques

Les premières études théoriques sur les algorithmes génétiques se sont intéressées à leur fonctionnement avec un codage binaire. Ces résultats sont connus sous le nom de la *théorie des schémas*. Ils sont largement développés dans [62]. Ils permettent de mieux comprendre l'effet du processus de sélection et de certains opérateurs de mutation et de croisements sur les *schémas* constituant le génotype des individus : il apparaît par exemple qu'avec les opérateurs envisagés, les schémas de petite taille (définis sur un nombre restreint de bits) sont favorisés et ont plus de chance d'être reproduits au cours des générations.

L'algorithme génétique est considéré comme un processus Markovien, perturbé par les opérateurs génétiques [6]. Cette étude théorique permet notamment d'établir le comportement asymptotique des algorithmes génétiques, ainsi que les cycles qui régissent la dynamique du processus. Elle démontre par ailleurs que l'algorithme peut converger sans croisement. Ces résultats théoriques ne rendent cependant pas encore compte du fonctionnement des algorithmes génétiques tels qu'ils sont utilisés en pratique, où leur convergence est fortement accélérée par des raffinements tels que le scaling et le sharing, ou l'utilisation d'opérateurs

adaptatifs dont l'efficacité est augmentée au fil des générations à partir d'approximations quadratiques de la fonction d'adaptation. De plus, les nombreux paramètres de ces algorithmes impliquent un réglage final qui nécessite souvent l'emploi de la méthode essai erreur.

CHAPITRE 3

APPLICATION A L'OPTIMISATION DE PROFILS D'AILE

3.1 Introduction

En aérodynamique, le rapport de la traînée sur la portance s'appelle l'efficacité aérodynamique. L'objectif de conception d'une aile est de minimiser ce rapport. Dans un écoulement typique ce rapport est lié à l'angle d'attaque d'écoulement, le nombre de Reynolds, de nombre de Mach et la forme du profil d'aile. Pour des écoulements à faibles vitesses où les fluctuations de densité sont relativement petites, des effets de nombre de Mach peuvent être négligés et le nombre de Reynolds peut être fixé à une certaine valeur. Cette approximation laisse seulement deux variables indépendantes : l'angle d'attaque et la forme du profil d'aile.

3.2 Modélisation de l'écoulement

3.2.1. Introduction

Les équations de Navier-Stokes sont très complexes et donnent une description complète de l'écoulement. Cependant, leur résolution numérique est très coûteuse en temps. Dans notre cas particulier, les effets visqueux sont importants seulement dans la région près du profil. Dans cette région, les équations de Navier-Stokes peuvent être approximées par les équations de couche limite dont les effets visqueux peuvent être négligés. On peut alors utiliser un modèle d'écoulement non visqueux.

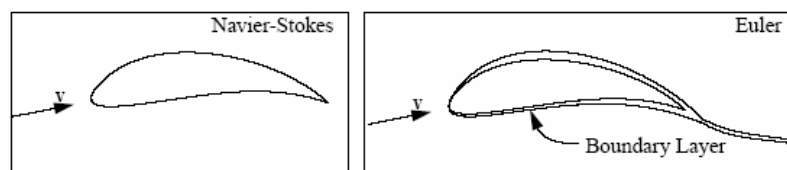


Figure 3.1 : L'approximation d'un modèle d'écoulement non visqueux.

- Le modèle d'écoulement non visqueux

Le modèle d'écoulement non visqueux peut être résolu de plusieurs manières. En utilisant une discrétisation par différences finies de l'équation d'Euler sur un maillage autour de profil d'aile, comme dans ISES [73], ou une méthode de panneau, comme dans Xfoil [74] et Panda [75]. Dans ce travail, on utilise une méthode de panneau couplée avec une technique d'approximation de couche limite [76] [34] [77]. La méthode des panneaux est basée sur une distribution de singularités (sources, tourbillons, doublets) non pas sur la corde, mais sur la surface du profil, qui donne une bonne précision de la solution, même avec seulement quelques panneaux. On détermine l'intensité de ces singularités grâce aux conditions de tangence et de Kutta. Le modèle d'écoulement non visqueux fournit la distribution tangentielle de vitesse sur la surface de l'aile. La distribution de pression est alors calculée par le champ de vitesse en utilisant l'équation de Bernoulli. Les coefficients de portance et de moment de tangage sont calculés en intégrant la pression sur la surface de profil.

On utilise les quantités : l'épaisseur de déplacement δ^* , l'épaisseur de quantité de mouvement θ , le facteur de forme H et le coefficient de frottement C_f de modèle de couche limite pour calculer le coefficient de traînée (voir figure 3.2).

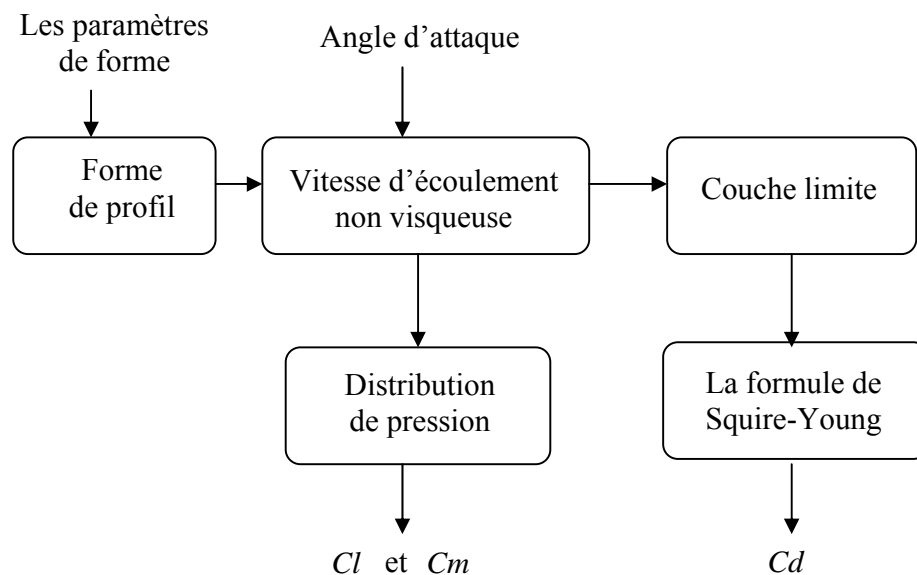


Figure 3.2 : Couplage du modèle d'écoulement non visqueux avec modèle de couche limite.

- Couplage

Le couplage des deux modèles se fait comme suit :

L'effet de la couche limite modifie la forme de profil. Ceci donne à l'écoulement non visqueux un état normal zéro de vitesse sur la frontière obtenue en ajoutant l'épaisseur de déplacement de couche limite à la surface de profil.

Les équations de couche limite dépendent de la distribution tangentielle de vitesse.

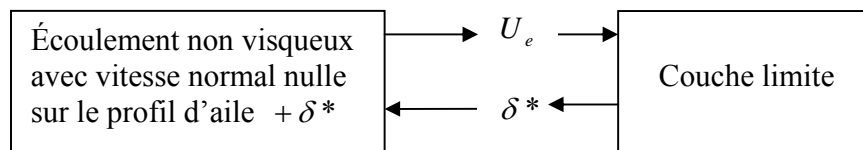


Figure 3.3 : Modification de profil d'aile

La vitesse tangentielle externe est calculée par le modèle non visqueux avec la condition $V \cdot n = 0$ sur la surface de profil d'aile, puis introduit dans le modèle de couche limite. Le coefficient de traînée est obtenu en utilisant la formule de Squire-Young [77], qui calcule en effet le déficit de quantité de mouvement en fonction des résultats de couche limite (épaisseur de quantité de mouvement et facteur de forme). On trouve la traînée total en ajoutant la traînée de frottement obtenue en intégrant le coefficient de frottement C_f à la traînée de pression obtenue en intégrant la distribution de pression de l'écoulement non visqueux.

3.2.2. Ecoulements potentiels et méthode de panneaux

La théorie du profil mince donne des résultats acceptables pour les coefficients de portance et de moment, mais elle ne tient pas compte de l'effet de l'épaisseur du profil sur les coefficients aérodynamiques. Suivant cette théorie, la pression aux points d'arrêt n'est pas valable.

- Fondements mathématiques :

La méthode élaborée par Hess et Smith [78] consiste tout d'abord à diviser le profil en un certain nombre de panneaux afin d'en approcher le mieux possible le contour. Ensuite, on utilise des sources et des tourbillons distribués sur chaque panneau pour modéliser

l'écoulement. On détermine l'intensité de ces singularités grâce aux conditions de tangence et de Kutta. Une fois les singularités connues, on peut calculer la vitesse et, par conséquent, le coefficient de pression C_p , en tout point de la surface de même que la distribution du coefficient de portance en fonction de l'angle d'attaque.

▪ Conditions aux limites :

Pour un écoulement irrotationnel, le potentiel total autour du profil correspond à l'équation :

$$\Phi = \Phi_{\infty} + \Phi_s + \Phi_v \quad (3.1)$$

où : Φ_{∞} = potentiel de l'écoulement uniforme

Φ_s = potentiel d'une distribution de sources $q(s)$ par unité de longueur

Φ_v = potentiel d'une distribution de tourbillons d'intensité $\gamma(s)$

D'après Hess et Smith [78] et à partir la figure (3.4), on peut écrire les trois expressions suivants pour le potentiel des vitesses :

$$\Phi_{\infty} = V_{\infty} (x \cdot \cos \alpha + y \cdot \sin \alpha)$$

$$\Phi_s = \int \frac{q(s)}{2\pi} \ln r ds$$

$$\Phi_v = \int \frac{\gamma(s)}{2\pi} \theta ds \quad (3.2)$$

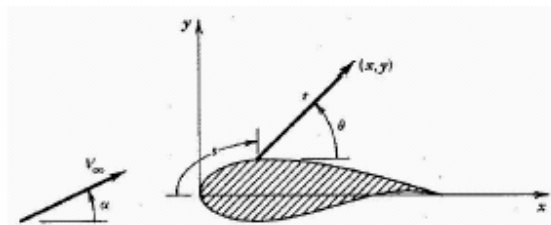


Figure 3.4 : Définition de s , de r et de θ

On doit intégrer ces deux dernières équations sur toute la surface du corps. En raison du principe de la superposition des écoulements, le potentiel total Φ satisfait automatiquement à l'équation de Laplace :

$$\nabla^2 \Phi = \frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} \quad (3.3)$$

De plus, il satisfait à la condition limite à l'infini, $r \rightarrow \infty$, $\Phi \rightarrow \Phi_\infty$. Il reste à déterminer les valeurs des intensités $q(s)$ et $\gamma(s)$ qui satisferont à la condition de l'écoulement tangent et à celle de Kutta. La distribution des sources remplira la condition de tangence de l'écoulement sur la surface et l'intensité tourbillonnaire, la condition de Kutta au bord de fuite.

- Discrétisation du profil

La discrétisation consiste à remplacer le profil par un polygone à N cotés. Il faut donc diviser la surface du profil en un certain nombre de panneaux, en commençant par le bord de fuite figure (3.5). Avec une soixantaine de panneaux, on obtient une assez bonne précision.

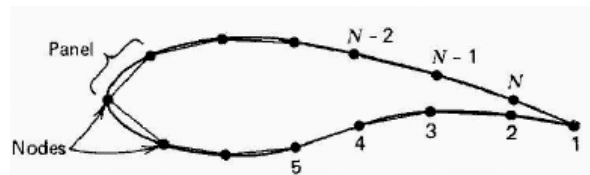


Figure 3.5 : Notation des panneaux sur le profil.

Afin de bien représenter le profil aérodynamique, on doit utiliser un grand nombre de panneaux au bord d'attaque et au bord de fuite, c'est à dire aux endroits où la courbure est importante. Pour ce faire, on utilise une distribution cosinusoidal des panneaux où $x_i = c/2 \cdot (1 - \cos \theta_i)$, $\theta_i = i \cdot \Delta \theta$ l'incrément $\Delta \theta$ est fonction du nombre des panneaux ($\Delta \theta = 2\pi / N$).

Sur chaque panneau, on distribue des sources et des tourbillons. Si on utilise les équations 3.1 et 3.2, le potentiel résultant en un point (x, y) devient :

$$\Phi = V_\infty (x \cos \alpha + y \sin \alpha) + \sum_{i=1}^N \int_{\text{panneau } i} \left(\frac{q(s)}{2\pi} \ln r - \frac{\gamma}{2\pi} \theta \right) ds \quad (3.4)$$

Il s'agit donc de récupérer un système de $N+1$ équations à $N+1$ inconnues, ce qui laisse la possibilité de choisir arbitrairement les distributions de sources et de tourbillons. D'après Hess et Smith [78], l'intensité de la source est constante sur chaque panneau, mais elle varie d'un panneau à l'autre. Ainsi $q(s) = q_i$ sur le panneau i ; $\gamma(s) = \gamma$ sur tout le profil. Cette méthode constitue un bon choix lorsqu'on veut limiter le nombre d'inconnues. Les inconnues du problème sont les suivantes :

$q_i = N$ valeurs pour les sources aux points de contrôle

$\gamma =$ une valeur pour le tourbillon aux points de contrôle

Les points de contrôle se trouvent au milieu de chaque panneau. Il faut déterminer l'expression pour ces points en regard du panneau i . Pour la condition de Kutta, on va poser que les valeurs des vitesses sur les panneaux N et 1 sont égales.

Coefficient de pression : avec la vitesse tangentielle sur chaque panneau, on peut calculer le coefficient de pression C_p à l'aide l'expression :

$$C_p = 1 - \left(\frac{U_e}{V_\infty} \right)^2 \quad (3.5)$$

- Calcul des coefficients aérodynamiques

Nous recherchons le coefficient de portance C_l , le coefficient de moment de tangage C_m qui représente le moment des forces aérodynamique par rapport le point des coordonnées $(c/4, 0)$

$$C_l = -\cos \alpha \cdot \int C_p dx - \sin \alpha \cdot \int C_p dy$$

$$C_{m,c/4} = \int (x - 1/4) \cdot C_p \cdot dx + \int y \cdot C_p \cdot dy \quad (3.6)$$

- Validation du modèle d'écoulement non visqueux

Les résultats sont alors comparés à la solution fournie par Xfoil, qui emploie exactement la même formulation et a été validé par les données expérimentales [74]

Le coefficient de portance C_l et le coefficient de moment de tangage C_m sont calculés par les deux programmes sont comparés dans plusieurs cas d'essai en respectant l'angle d'attaque et les paramètres de forme, sont donné dans la figure (3.6).

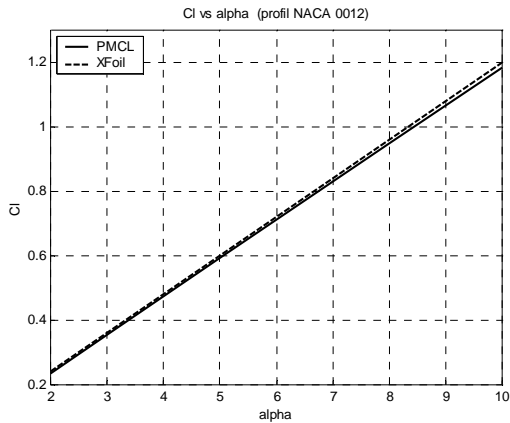


Figure 3.6.a : Coefficient de portance en fonction d'angle d'incidence de profil NACA 0012.

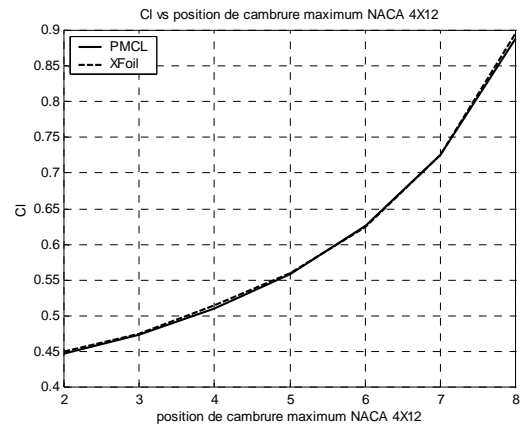


Figure 3.6.b : Coefficient de portance en fonction de position de cambrure maximum de profil NACA 4X12 avec angle d'incidence nul

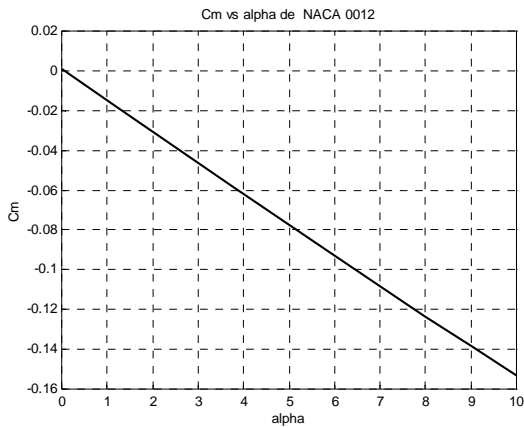


Figure 3.6.c : Coefficient de moment de tangage en fonction de l'angle d'incidence de profil NACA 0012

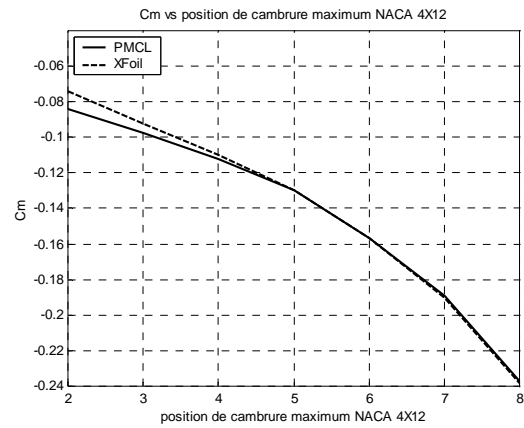


Figure 3.6.d : Coefficient de moment de tangage en fonction de position de cambrure maximum de profil NACA 4X12 avec angle d'incidence nul.

4.2.3. Modèle de couche limite

- Hypothèses :

En faisant l'hypothèse que la pression statique à travers la couche limite est constante et que la composante normale de la vitesse y est faible, ces équations peuvent être simplifiées sous la forme des équations de couche limite dérivées par Prandtl. Parmi les modèles les plus simples, on retrouve les méthodes intégrales basées sur l'équation de Von Karman dérivée à partir d'hypothèses simplificatrices et des équations de couche limite. Cette équation est applicable pour des couches limites laminaires ou turbulentes, bidimensionnelles et avec gradients de pression. Pour une couche limite laminaire ou turbulente se développant sur un corps de révolution soumis à un écoulement incompressible, 2-D, en régime permanent et où les forces de volume extérieures sont négligeables.

Le modèle employé : la partie laminaire de l'écoulement est calculée avec modèle de Twaites [77], et la partie turbulente est résolue avec le modèle de Head [34]. critère de Michel [77] pour localiser le point de transition. Le coefficient de traînée est calculé en utilisant la formule de Young- Squire [34].

3.2.3.1. Travail préliminaire

Avant utiliser le modèle de couche limite, il est recommandé de suivre :

- la distribution des vitesses et les dimensions du profil d'aile doivent être adimensionnel. Ainsi, la vitesse tangente est divisée sur la vitesse à l'infinie d'écoulement et les coordonnées de point de profil d'aile sont divisées par la longueur de corde de profil d'aile.
- La couche limite devrait commencer au point d'arrêt, et suit l'écoulement le long de l'intrados ou l'extrados vers le bord de fuite. Ceci exige identifier le point d'arrêt dans la solution non visqueuse et employer la discrétisation de panneau pour calculer la longueur d'arc le long de la surface à partir point d'arrêt, qui est la longueur x dans les équations de couche limite.

- La discrétisation de chaque côté du profil peut être donnée directement par les panneaux utilisés pour le calcul d'écoulement. Mais pour résoudre la couche limite exactement sans devoir employer beaucoup de panneaux pour la solution d'écoulement non visqueuse, il est plus commode d'établir une discrétisation uniforme de longueur d'arc de chaque côté. Cette discrétisation est basée sur une spline d'interpolation des points des panneaux.
- Le modèle de couche limite emploie la vitesse tangentielle, aussi bien que sa première dérivée spatiale. La valeur de la vitesse à chaque point de discrétisation est obtenue en utilisant une spline d'interpolation du champ de vitesse connu aux noeuds de panneau. Le dérivé de la vitesse est obtenu en différenciant le spline d'interpolation.

3.2.3.2. La couche limite laminaire

- Équation théorique de l'écoulement

En analysant et en comparant l'ordre de grandeur des termes des équations incompressible de Navier-Stokes, nous pouvons dériver les équations suivantes de la couche limite de Prandtl (des détails des calculs et des approximations peuvent être trouvés dans [34]) :

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (3.7)$$

$$\rho \left(u \frac{\partial u}{\partial x} + v \frac{\partial v}{\partial y} \right) = -\frac{\partial p}{\partial x} + \mu \frac{\partial^2 u}{\partial y^2} \quad (3.8)$$

$$\frac{\partial p}{\partial y} = 0 \quad (3.9)$$

Note : x et y ne sont pas des coordonnées cartésiennes. x est mesuré le long de la surface de l'aile avec $x=0$ localisant le point d'arrêt, et y est mesuré normal sur la surface.

Les hypothèses les plus importantes utilisées dans la dérivation sont :

- L'épaisseur de couche limite est très petite comparée à c (corde) pour les grands nombres de Reynolds,

- La vitesse tangentielle u est beaucoup plus grande que le composant normal v ,
- La pression est essentiellement constante à travers la couche limite (dans la direction normale de y).

Ces résultats mènent naturellement au concept d'une formulation de couche limite, où on suppose que les propriétés dépendent seulement de x .

▪ Équation intégrale de quantité de mouvement

En combinant les équations de couche limite, par l'intégration des expressions résultante de zéro à l'infini de y , on obtient l'équation de quantité de mouvement de Von Karman :

$$\frac{d\theta}{dx} + \frac{\theta}{U_e} \left(2 + \frac{\delta^*}{\theta} \right) \frac{dU_e}{dx} = \frac{1}{2} C_f \quad (3.10)$$

Les épaisseurs de déplacement δ^* , quantité de mouvement θ le coefficient du frottement C_f définis comme suit :

$$\delta^* = \int_0^{\infty} \left(1 - \frac{u}{U_e} \right) dy$$

$$\theta = \int_0^{\infty} \frac{u}{U_e} \left(1 - \frac{u}{U_e} \right) dy$$

$$C_f = \frac{\tau_w}{\frac{1}{2} \cdot \rho U_e^2}, \text{ ou } \tau_w = \mu \left. \frac{\partial u}{\partial y} \right|_{y=0}$$

Note : L'épaisseur de déplacement peut être vue comme le déplacement nécessaire de profil d'aile pour construire un profil de vitesse constant par la couche limite ayant le même volume d'écoulement que dans le vrai cas.

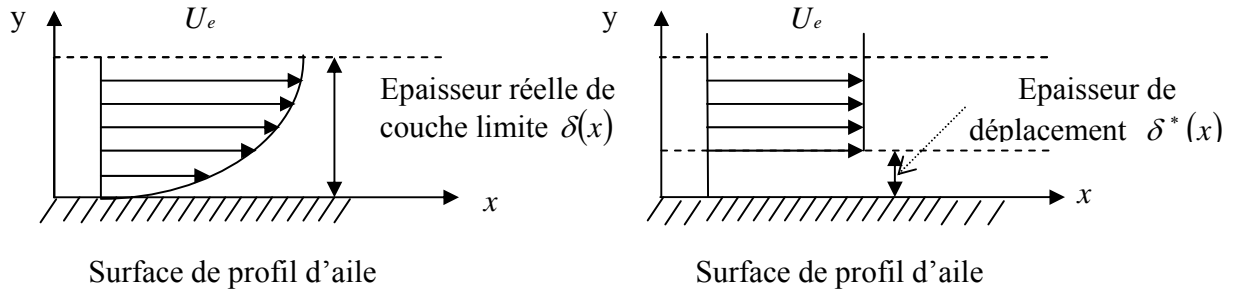


Figure 3.7 : Équivalence en débit d'écoulement.

D'après la figure (3.7), cette équivalence en débit d'écoulement peut être analytiquement exprimée comme :

$$\int_0^{\infty} (U_e - u) dy = \delta^* \cdot U_e \quad (3.11)$$

En présentant le facteur de forme $H = \delta^*/\theta$, l'équation de quantité de mouvement peut être écrite comme suit :

$$\frac{d\theta}{dx} + \frac{\theta}{U_e} (2 + H) \frac{dU_e}{dx} = \frac{1}{2} C_f \quad (3.12)$$

- Équation intégrale d'énergie cinétique

Si l'équation quantité de mouvement est multipliée par u puis intégrée, l'équation d'énergie cinétique résulte est :

$$\frac{d\theta^*}{dx} + 3 \frac{\theta^*}{U_e} \frac{dU_e}{dx} = 2C_D, \quad (3.13)$$

L'épaisseur d'énergie cinétique θ^* et le coefficient de dissipation C_D sont définis par :

$$\theta^* = \int_0^{\infty} \left(1 - \left(\frac{u}{U_e} \right)^2 \right) \frac{u}{U_e} dy \quad \text{et} \quad C_D = \frac{1}{\rho U_e^3} \int_0^{\infty} \mu \frac{\partial^2 u}{\partial y^2} dy$$

Puis, en introduisant le deuxième paramètre de forme $H^* = \theta^*/\theta$ et soustrayant l'équation (3.12) de l'équation (3.13), l'équation d'énergie cinétique écrite comme suit :

$$\theta \frac{dH^*}{dx} + [H^*(H-1)] \frac{\theta}{U_e} \frac{dU_e}{dx} = 2C_D - H^* \frac{C_f}{2} \quad (3.14)$$

- Le modèle de Thwaites

Ce modèle simple se fonde sur l'équation intégrale de quantité de mouvement et devient comme suit :

Multipliant l'équation (3.12) par le nombre de Reynolds basé sur l'épaisseur de quantité de mouvement

$Re_\theta = Re \theta U_e$ Nous obtenons :

$$Re U_e \frac{d\theta^2}{dx} = 2[L - (2 + H)\lambda],$$

$$\text{où : } L = Re \theta U_e \frac{C_f}{2} \text{ et : } \lambda = Re \theta^2 \frac{dU_e}{dx},$$

Puis, Thwaites a constaté que le côté droit peut être rapproché par la formule linéaire :

$$2[1 - (2 + H)\lambda] \approx 0.45 - 6\lambda$$

ça et la définition de λ sont substitués dans l'équation, nous obtenons :

$$Re U_e \frac{d\theta^2}{dx} = 0.45 - 6 Re \theta^2 \frac{dU_e}{dx}$$

Ce qui mène à l'équation :

$$Re \frac{d}{dx} (\theta^2 U_e^6) = 0.45 U_e^5$$

La valeur de θ au point d'arrêt est :

$$\theta(x=0) = \sqrt{\frac{0.075}{\operatorname{Re} \frac{dU_e}{dx}(0)}}$$

À partir du là, l'intégration est effectuée comme suit :

$$\operatorname{Re}(\theta^2 U_e^6) \Big|_{i-1}^i = 0.45 \int_{x_{i-1}}^{x_i} U_e^5 dx$$

L'intégrale est évaluée en utilisant la quadrature du gauss d'ordre 5.

$$\int_{x_{i-1}}^{x_i} U_e^5(x) dx \approx \frac{dx}{18} \left[5U_e^5 \left(x_m - \sqrt{\frac{3}{5}} \frac{dx}{2} \right) + 8U_e^5(x_m) + 5U_e^5 \left(x_m + \sqrt{\frac{3}{5}} \frac{dx}{2} \right) \right],$$

$$\text{où : } dx = x_i - x_{i-1} \quad \text{et} \quad x_m = (x_i + x_{i-1})/2$$

Une fois que θ est connu, λ peut être calculé. Le facteur de forme H et le coefficient de frottement C_f sont alors calculés à partir des formules semi empiriques:

$$H(\lambda) = 2.61 - 3.75\lambda + 5.24\lambda^2 \quad \text{pour} \quad 0 < \lambda < 0.1$$

$$= 2.088 + \frac{0.0731}{\lambda + 0.14} \quad \text{pour} \quad -0.1 < \lambda < 0$$

$$\text{et} \quad C_f = \frac{2L(\lambda)}{\operatorname{Re} V_e \theta},$$

$$\text{Avec} \quad L(\lambda) = 0.22 + 1.57\lambda - 1.8\lambda^2 \quad \text{pour} \quad 0 < \lambda < 0.1$$

$$= 0.22 + 1.402\lambda + \frac{0.018\lambda}{\lambda + 0.107} \quad \text{Pour} \quad -0.1 < \lambda < 0$$

3.2.3.3. Localisation le point de transition laminaire-turbulente

La transition de l'écoulement laminaire à turbulent est un résultat de la croissance des perturbations qui rendent la couche limite laminaire instable. La prévision précise de la région de transition est un point crucial pour obtenir une bonne évaluation de la traînée puisqu'elle sépare la région laminaire d'écoulement, où la traînée de frottement est basse, à la région turbulente d'écoulement, où la traînée de frottement augmente considérablement.

- Le critère de Michel

Ce critère est basé sur l'idée que la transition commence à un nombre de Reynolds spécifique basé sur la distance x dès le début de la couche limite. La valeur du nombre de Reynolds de transition dépend de plusieurs facteurs, le plus important est le gradient de pression imposé à la couche limite par l'écoulement non visqueux et la rugosité de l'aile.

Pour des écoulements incompressibles sans transfert thermique, Michel a examiné une variété de données et a conclu que, pour les applications de profil d'aile, la transition devrait être prévue quand :

$$\text{Re}_\theta > \text{Re}_{\theta_{\max}} = 1.174 \left(1 + \frac{22.4}{\text{Re}_x} \right) (\text{Re}_x)^{0.46} \quad (3.15)$$

où:

$$\text{Re}_\theta = \text{Re} U_e \theta$$

$$\text{Re}_x = \text{Re} U_e x$$

La formule (3.15) explique l'effet du gradient de pression, parce que l'épaisseur de quantité de mouvement augmente plus rapidement dans un gradient positif de pression. Cependant, elle n'inclut pas l'effet de la rugosité de surface, mais étant basé sur des données prises sur des ailes, il devrait être bon pour l'analyse d'aile.

3.2.3.4. La couche limite Turbulente

Pour un assez grand nombre de Reynolds, la couche limite peut devenir turbulente, on utilise le modèle de Head [77]

- Modèle de Head

Le modèle de Head est décrit en détail dans [34]. C'est une méthode intégrale, où certains des procédures analytiques ont été suivis avant que le problème numérique soit posé.

Le modèle utilise l'équation de Von Karman, et comme dans le cas laminaire, une partie des relations semi empirique pour fermer le système. La méthode a été dérivée comme suit.

Nous considérons le débit de volume dans la couche limite à une position x :

$$Q(x) = \int_0^{\delta(x)} u dy, \quad (3.16)$$

Où $\delta(x)$ est l'épaisseur de couche limite. Combinaison de ceci avec la définition de l'épaisseur de déplacement, nous trouvons

$$\delta^* = \delta - \frac{Q}{U_e}$$

Nous présentons la vitesse d'entraînement :

$$E = \frac{dQ}{dx} = \frac{d}{dx} U_e (\delta - \delta^*) \quad (3.17)$$

Ce que nous écrivons :

$$E = \frac{d}{dx} (U_e \theta H_1)$$

Avec.

$$H_1 = \frac{\delta - \delta^*}{\theta}$$

Head a supposé que la vitesse sans dimensions E/U_e d'entraînement dépend seulement de H_1 , et H_1 est une fonction de H .

Les équations différentielles aux valeurs initiales sont résolues numériquement par la méthode de Runge-Kutta d'ordre 4.

$$\frac{1}{U_e} \frac{d}{dx} (U_e \theta H_1) = 0.0306 (H_1 - 3)^{-0.6169}$$

et

$$H_1 = k(H) = 3.3 + 0.8234(H - 1.1)^{-1.287} \quad \text{pour } H \leq 1.6$$

$$= 3.3 + 1.5501(H - 0.6778)^{-3.064} \quad \text{pour } H > 1.6$$

Une quatrième équation employée pour résoudre pour les inconnus θ , le H , le H_1 et le C_f est la loi de frottement de Ludwig-Tillman

$$C_f = 0.246 \cdot (10^{-0.678H}) \text{Re}_\theta^{-0.268} \quad (3.18)$$

La valeur pour H à la séparation souvent présentée dans la littérature [34] est $H = 2.4$

La figure (3.8) illustre le schéma de fonctionnement du modèle de couche limite utilisé pour les différents codes développés.

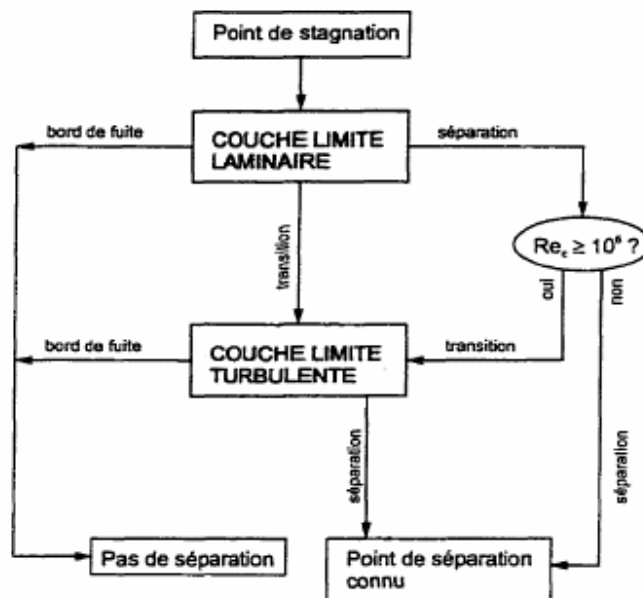


Figure 3.8 : Modèle de couche limite laminaire-turbulente utilisé pour les différents codes

3.2.4. Calcul le coefficient de la traînée

La formule de Young-Squire fournit des moyens de prévoir la traînée de profil en reliant le déficit de quantité de mouvement loin en aval aux valeurs du champ d'écoulement donné au bord de fuite.

Etant donné les grandeurs adimensionnel de l'épaisseur quantité de mouvement θ , le facteur de forme H et la vitesse U_e sur l'extrados et l'intrados au bord de fuite, la traînée trouvée est:

$$C_d = \left[2\theta_{TE} \cdot (U_e)_{TE}^{\frac{H_{TE}+5}{2}} \right]_{UP} + \left[2\theta_{TE} \cdot (U_e)_{TE}^{\frac{H_{TE}+5}{2}} \right]_{LOW} \quad (3.19)$$

Cette formule donne directement toute la traînée de profil (c.-à-d. la somme de la traînée due au frottement et de pression) en fonction des valeurs de U_e , θ et H au bord de fuite, calculé sur les deux côtés de profil d'aile.

La figure (3.9) illustre le coefficient de traînée Cd est calculé par notre programme et comparé par le code de calcul Xfoil dans plusieurs cas d'essai en respectant l'angle d'attaque et les paramètres de forme.

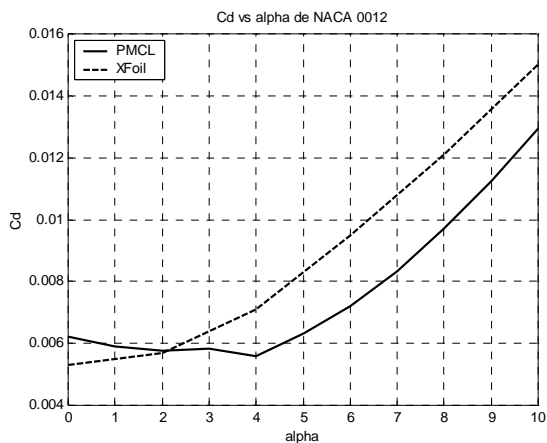


Figure 3.9.a : Coefficient de traînée en fonction d'angle d'incidence de profil NACA 0012.

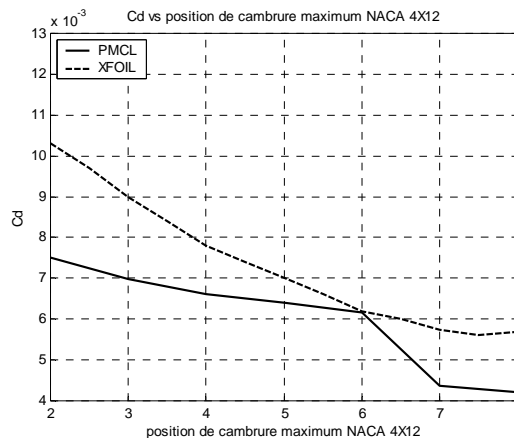


Figure 3.9.b : Coefficient de traînée en fonction de position de cambrure maximum de profil NACA 4X12 avec angle d'incidence nul

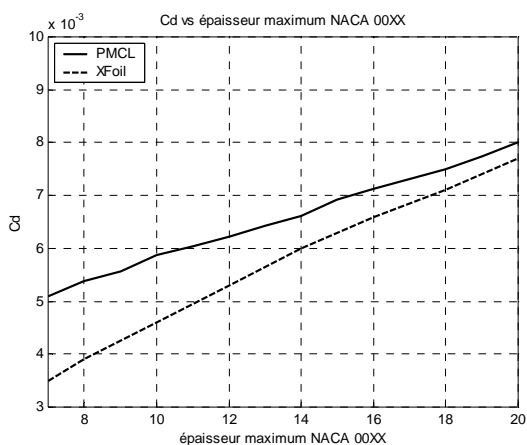


Figure 3.9.c : Coefficient de traînée en fonction d'épaisseur maximum de profil NACA 00XX avec angle d'incidence nul.

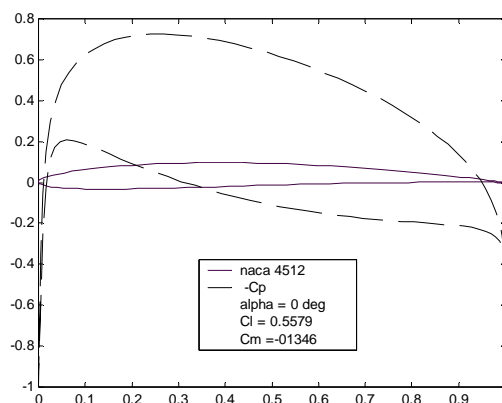


Figure 3.9.d : Distribution de coefficient de pression pour NACA 4512.

3.3. Paramétrisation du profil

La représentation d'une courbe du plan ou d'une surface de l'espace est un problème qui est à la base de la géométrie. Nous allons nous intéresser uniquement aux techniques bidimensionnelles, mais il est important de noter que ces techniques peuvent toutes s'appliquer pour modéliser une surface de l'espace tridimensionnel. On peut distinguer quatre méthodes principales permettant la représentation d'une courbe :

Les transformations conformes permettant d'obtenir les profils de Joukovski sont aujourd'hui rarement utilisées mais pas complètement abandonnées [50]. Les profils NACA à 4 ou 5 chiffres dont la géométrie est définie par des fonctions analytiques sont encore très utilisés, sans doute à cause de l'énorme quantité de données expérimentales et numériques disponibles sur ces profils.

Les courbes de Bézier ont introduit la notion de point de contrôle permettant une interaction simple et visuelle avec la courbe. De plus, la simplicité de leur définition les rend très faciles à utiliser mais représente aussi un inconvénient dans le sens qu'elle limite un peu trop le champ des courbes représentables. C'est pourquoi les courbes de Bézier sont de moins en moins utilisées [79] au profit des B-splines qui se sont imposées ces dernières années comme la méthode d'interpolation de référence. Ce sont en effet elles qui sont utilisées par la plupart des auteurs [80] [81] car elles possèdent de nombreux avantages.

3.3.1. Représentation de NACA à 4 chiffres

La forme d'un profil d'aile est définie en utilisant plusieurs paramètres ; épaisseur maximum, distribution de cambrure, rayon de bord d'attaque, etc.....

Après la deuxième guerre mondiale les scientifiques ont développé une théorie de combiner la ligne moyenne et la distribution d'épaisseur pour obtenir le profil d'aile de NACA désiré. Cette théorie a permis pour obtenir des formes d'aile en utilisant seulement trois paramètres.

1. Épaisseur maximum,
2. Cambrure maximum
3. position de cambrure maximum.

La connaissance de ces trois paramètres avec l'angle d'attaque est suffisante pour obtenir l'efficacité aérodynamique d'un profil d'aile particulière.

La forme du profil d'aile peut être choisie dans les profils NACA à 4 chiffres [82]. La forme de ce type de profil est exprimée analytiquement en fonction de trois paramètres : la cambrure maximum, l'endroit de cambrure maximum, et épaisseur maximum de l'aile. Une grande variété d'ailes peut être obtenue en changeant les trois paramètres.

Un système de numération est employé pour définir des sections d'aile de NACA 4-chiffre. Le premier chiffre indique la valeur de cambrure maximum en pourcentage de la corde. Le deuxième nombre entier indique la distance du principal bord à l'endroit de la cambrure maximum dans les dizaines de la corde. Les deux derniers nombres entiers indiquent l'épaisseur de section en pourcentage de la corde. Ainsi, le NACA 4512 a la cambrure maximum de 4% située à 50 % de la corde à partir du bord principal, et 12% épais maximum.

Nous notons que ces chiffres ne doivent pas vraiment être des nombres entiers. Par la prolongation de la notation, a NACA 4.23 2.21 17.2 aurait la cambrure maximum de 4.23% située à 22.1% de la corde à partir du bord principal, et soit 17.2% épais maximum [34].

Le profil d'aile est obtenu en combinant la ligne de cambrure et la distribution d'épaisseur comme montrée sur la figure suivante:

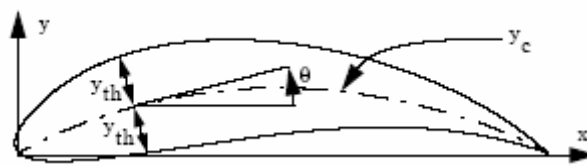


Figure 3.10 : Représentation d'un profil d'aile type NACA

$$\begin{cases} x_U = x - y_{th} \cos \theta \\ y_U = y_c + y_{th} \sin \theta \end{cases}$$

et

$$\begin{cases} x_U = x + y_{th} \cos \theta \\ x_U = y_c - y_{th} \sin \theta \end{cases}$$

Où (x_U, y_U) et (x_L, y_L) sont les points sur l'extrados et l'intrados respectivement.

La distribution d'épaisseur et la ligne de cambrure sont données par :

$$y_{th} = 5\tau \cdot c \left(0.2969 \sqrt{\frac{x}{c}} - 0.126 \frac{x}{c} - 0.3537 \left(\frac{x}{c} \right)^2 + 0.2843 \left(\frac{x}{c} \right)^3 - 0.1015 \left(\frac{x}{c} \right)^4 \right) \quad (3.20)$$

et

$$y_c = \frac{m}{p^2} \left(2p \frac{x}{c} - \left(\frac{x}{c} \right)^2 \right) \quad \text{pour } \frac{x}{c} \leq p$$

$$= \frac{m}{(1-p)^2} \left(1 - 2p + 2p \frac{x}{c} - \left(\frac{x}{c} \right)^2 \right) \quad \text{pour } \frac{x}{c} \geq p$$

Dans ces expressions, c est la longueur de corde de profil d'aile, m est la cambrure maximum, p est l'endroit de cambrure maximum, et τ est l'épaisseur maximum.

3.3.2. Représentation en B-spline

Le profil d'aile est représenté par la courbe de B-spline d'ordre 6 [80] pour les deux surfaces de profil. Les coordonnées des points de contrôle sont sur l'intrados et l'extrados avec bord de fuite fermé.

On définit les fonctions de bases de la courbe de B-spline [83] façon suivante :

Soit $U = \{u_0, \dots, u_m\}$ une suite croissante de nombres réels appelés *noeuds* où U est le vecteur nodal. La $i^{\text{ème}}$ fonction de base de degré d , correspondant à une B-spline d'ordre $d+1$, notée $N_i^d(u)$ est définie pour $0 \leq i \leq m-d-1$ comme suit :

$$N_i^0(u) = \begin{cases} 1 & \text{si } u_i \leq u \leq u_{i+1} \\ 0 & \text{si non} \end{cases}$$

$$N_i^d(u) = \frac{u - u_i}{u_{i+d} - u_i} N_i^{d-1}(u) + \frac{u_{i+d+1} - u}{u_{i+d+1} - u_{i+1}} N_{i+1}^{d-1}(u) \quad (3.21)$$

De cette définition, il ressort clairement que les fonctions N_i^p sont nulles en dehors de l'intervalle $[u_i, u_{i+d+1}]$ ce qui confère leur caractère local aux B-splines. Le choix du vecteur

nodal permet une très grande variété de fonctions de base. Dans le cas particulier où ce vecteur est de la forme :

$$\left\{ \underbrace{0, \dots, 0}_d, u_{d+1}, \dots, u_{m-d+1}, \underbrace{1, \dots, 1}_d \right\} \quad (3.22)$$

Où les noeuds entre 0 et 1 sont régulièrement espacés, alors la courbe est dite uniforme. Le degré d des fonctions utilisées joue un rôle essentiel sur la représentation géométrique de la courbe. Plus le degré est bas, plus la courbe "suit" les points de contrôle et plus le domaine d'influence d'un point est petit, mais moins elle est régulière. Cette propriété est illustrée par la figure (3.11). Dans le cas qui nous intéresse, le degré 6 est un bon compromis car il assure la continuité de la courbe, de ses tangentes et de sa courbure tout en lui laissant suffisamment de liberté pour bien refléter les détails d'un profil d'aile, comme représenté dans la figure (3.11).

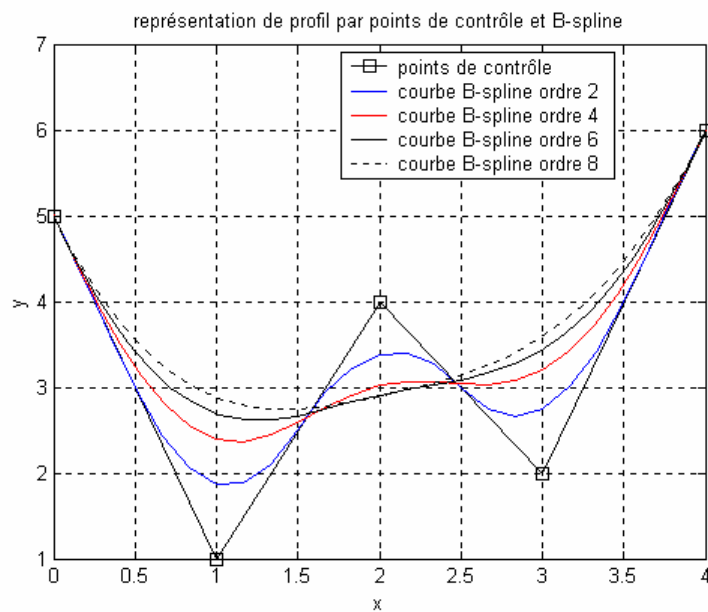


Figure 3.11 : L'influence de l'ordre sur une courbe B-spline à 5 points de contrôle.

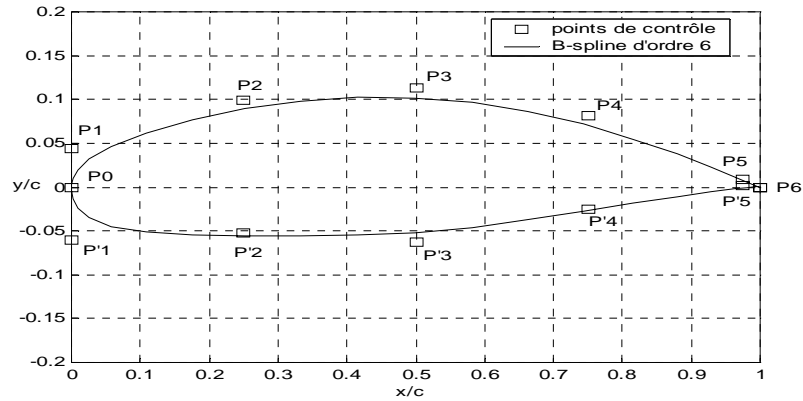


Figure 3.12 : Représentation B-spline et points de contrôles de profil d'aile

3.4 Spécificités de l'Algorithme génétique

3.4.1. Le Codage

Les algorithmes génétiques diffèrent des autres méthodes d'optimisation car ils utilisent un codage de ses variables, plutôt que les variables elles-mêmes. Le codage désigne le processus qui transforme les variables en un chromosome

$$x_c = [(x_c)_1, (x_c)_2, \dots, (x_c)_n] \in R^n \rightarrow [a_1, a_2, \dots, a_n] \quad (3.23)$$

Les éléments constituant le chromosome sont les gènes. Ils appartiennent à un ensemble appelé *alphabet*. L'algorithme historique emploie un codage binaire qui s'appuie sur l'alphabet de $\{0,1\}$

Le codage binaire suppose en pratique la définition d'un intervalle admissible pour les variables de contrôle et une discrétisation de cet intervalle. Pour chaque variable de contrôle réelle $(x_c)_i$, on définit une limite inférieure l_i^{inf} et une limite supérieure l_i^{sup} au domaine de variation. En supposant que la discrétisation fournisse une précision de N_i^{cs} chiffres significatifs, le nombre de gènes nécessaire au codage de la variable $(x_c)_i$ est le plus petit entier l_i vérifiant :

$$(l_i^{\text{sup}} - l_i^{\text{inf}}) 10^{N_i^{\text{cs}}} \leq 2^{l_i} - 1 \quad (3.24)$$

Un chromosome étant constitué des gènes associés à chaque variable de contrôle mis bout à bout, le nombre total de gènes d'un chromosome, caractérisant un individu et un point dans l'espace de recherche est alors :

$$l = \sum_{i=1}^n l_i \quad (3.25)$$

La transformation d'un chromosome en variables réelles peut se faire de manière naturelle par la relation :

$$(x_c)_i = l_i^{\text{inf}} + \frac{l_i^{\text{sup}} - l_i^{\text{inf}}}{2^{l_i} - 1} \sum_{i_{bit}=1}^{l_i} 2^{i_{bit}-1} Ch(i_{bit}) \quad (3.26)$$

où $Ch(i_{bit})$ représente la valeur binaire, 0 ou 1, du gène i_{bit} associé à la variable $(x_c)_i$.

- Codage Gray

Le codage, bien qu'il soit une simple conversion du système de variables, est important dans la mesure où il influe sur le comportement de l'algorithme. Il est difficile, voire impossible de connaître a priori les qualités d'un codage donné pour un problème spécifique.

Néanmoins, le codage binaire choisit pour l'algorithme historique est généralement un choix médiocre. Pour s'en convaincre, il suffit de constater que deux chromosomes variant d'un unique gène peuvent fournir deux valeurs des variables de contrôle très éloignées, cette caractéristique est généralement néfaste à la résolution d'un problème présentant certaines propriétés de régularité.

On peut facilement améliorer le codage binaire en introduisant un codage de type Gray. Une caractéristique de ce codage est que deux valeurs adjacentes ne diffèrent que d'un seul gène. L'utilisation d'un codage Gray est généralement bénéfique, pour la plupart des problèmes dont les variables de contrôle sont réelles.

3.4.2. L'espace de recherche :

Pour cette approche d'optimisation, nous devons limiter les paramètres à optimiser, de sorte que l'optimisation soit exécutée dans un espace de recherche raisonnable. Nous choisissons les valeurs minimales et maximales suivantes:

- Profil d'aile type NACA

Tableau 3.1: le domaine de l'espace de recherche pour profil type NACA

Les paramètres	Limite minimum	Limite maximum
cambrure maximum	0%	9%
position de cambrure maximum	20%	80%
épaisseur maximum	07%	20%
angles d'attaque	0 degré	3 degré

- Profil d'aile type B-spline

Nous ne permettrons aux points de contrôle de varier que dans la direction verticale, c'est-à-dire que les x_i sont fixés, aux valeurs suivantes:

$x_0 = x_1 = 0.00$, $x_2 = 0.25$, $x_3 = 0.5$, $x_4 = 0.75$, $x_5 = 0.97$, $x_6 = 1.00$, Les 2 premiers points de contrôle sont tous deux d'abscisse nulle ($x_0 = x_1 = 0$), ce qui contraint le profil à avoir une tangente verticale à l'avant.

En effet les courbes de B-spline sont tangentes aux segments formés par les 2 premiers et les 2 derniers points de contrôle, le premier et le dernier point de contrôle appartiennent à la courbe. Ces 2 points (les extrémités) sont fixes, donc les ordonnées y_0 et y_6 sont fixes: $y_0 = y_6 = 0.00$.

Ainsi nous allons jouer sur 10 paramètres pour déformer le profil: ($y_1; y_2; y_3; y_4; y_5$) pour l'extrados et ($y'_5; y'_4; y'_3; y'_2; y'_1$) pour l'intrados. D'après la figure (3.11), on fixe dans le tableau (3.2) les intervalles de variation suivants (pour une longueur de profil unité)

Tableau 3. 2 : le domaine de l'espace de recherche pour profil type B-spline

	Valeurs min	Valeurs max
y_0, y_6	0.00	0.00
y_1	0.01	0.025
y_2, y_3, y_4	0.01	0.10
y_5	0.00	0.01
y'_1	-0.06	-0.03
y'_2, y'_3, y'_4	-0.07	0.02
y'_5	-0.002	0.002

3.4.3. La fonction objective (fitness)

Pour un problème d'optimisation mono-objectif, une solution optimale unique peut exister. Cependant, pour le problème d'optimisation multi-objectif, il existe un ensemble de solutions compromises connu, les solutions Pareto-optimales ou solutions non-dominées.

Le but de l'optimisation multi-objective est de trouver un ensemble de solutions. Une fois que les solutions sont obtenues, le concepteur peut pouvoir choisir une conception finale avec d'autres considérations.

Une approche bien connue pour résoudre des problèmes multi objectifs est la méthode de la somme des coefficients de pondération .En employant cette méthode, le problème est transformé en problème d'optimisation mono-objectif en combinant des objectifs multiples en objectif simple. Par exemple, pour réduire au minimum de plusieurs fonctions objectifs, une fonction scalaire F peut être définie comme suit :

$$F = w_1 obj_1 + w_2 obj_2 + \dots + w_n obj_n \quad (3.27)$$

avec $\sum_{i=1}^n w_i = 1$

Pour chaque valeur de w_i , cette approche trouve une des solutions Pareto-optimales.

Dans le présent travail les valeurs de w_i ont été prise de même valeur, les objectifs sont de la même importance. Par conséquent, le problème multi objectif réduit à un problème mono objectif [80] [81].

On utilise la fonction objective selon le problème sur lequel on travaille.

3.4.4. L'opérateur de sélection

Il existe plusieurs méthodes pour sélectionner les individus les mieux adaptés. Pour des raisonnements cité auparavant on utilise la sélection par échantillonnage stochastique de la partie restante sans remplacement (*stochastic remainder without replacement selection*).

Un exemple de cette méthode se trouve dans l' APPENDICE B

3.4.5. Croisement et mutation :

3.4.5.1 L'opérateur de croisement

En binaire la méthode la plus courante consiste à prendre un nombre aléatoire k compris entre 1 et la longueur l de la chaîne. A partir des chaînes de caractères de deux individus parents, on crée deux nouvelles chaînes par permutation de tous les caractères compris entre les positions $k+1$ et l des chaînes.

La figure (3.13) illustre un couple de profil d'aile avant l'application de l'opérateur de croisement.

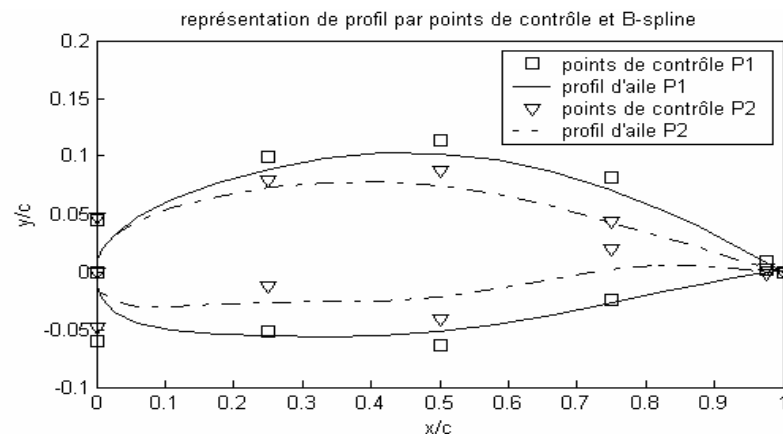


Figure 3.13 : Les profils d'aile avant croisement

La figure (3.14) illustre le couple de profil d'aile après l'application de l'opérateur de croisement.

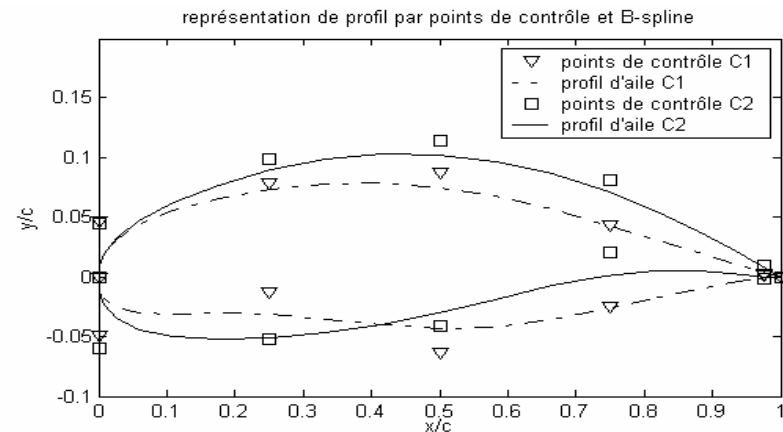


Figure 3.14 : Les profils d'aile après croisement

3.4.5.2 L'opérateur de mutation

Il vise à augmenter la diversité de la population pour éviter les optimums locaux, en modifiant de façon aléatoire et occasionnelle une partie de la population (voir figure 3.15). Le principe est de choisir une valeur de remplacement aléatoire pour l'un des gènes des individus de la population concernée. Lors d'un codage binaire des chaînes, cela

revient à changer un 1 en 0 et vice versa. Ceci permet à la fois de pouvoir explorer le domaine complet de variation des variables et également d'éviter un appauvrissement de la population qui pourrait entraîner, en particulier, la convergence vers un extremum local. Pour converger plus régulièrement et plus rapidement, la procédure peut être mise en place

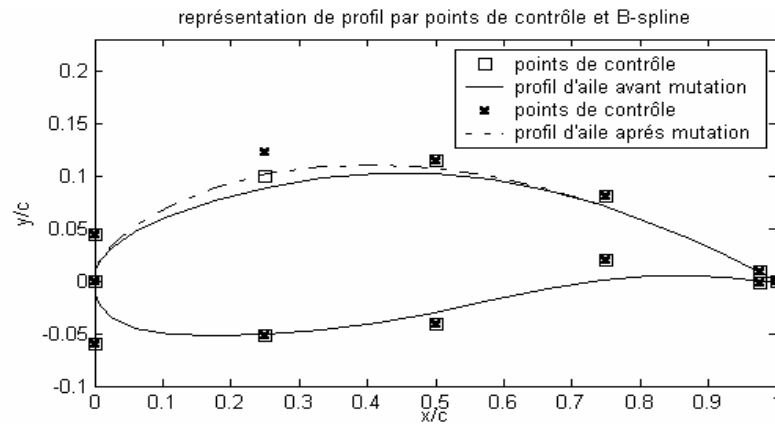


Figure 3.15 : Mutation simple des bits

3.4.6. Procédure élitiste :

On travaille de façon à ce que la fonction fitness ne puisse que croître. Si à une génération i , on a trouvé un meilleur chromosome on le stocke pour les générations suivantes jusqu'à ce que l'on en trouve un meilleur.

3.4.7. Les paramètres d'algorithme génétique

A chaque fois que nous utilisons l'algorithme génétique, nous gardons fixées un certain nombre de constantes et de procédures. En effet, des études ont été menées auparavant [79], afin de déterminer certains paramètres comme les probabilité de croisement et de mutation afin que la convergence de l'algorithme génétique soit optimale. Nous avons utilisé les résultats obtenus, c'est-à-dire:

Type de sélection : sélection par échantillonnage stochastique de la partie restante sans remplacement (*stochastic remainder without replacement selection*)

Type de croisement : croisement simple

Probabilité de croisement : 0.85

Type de mutation : mutation binaire

Probabilité de mutation : 0.02

Procédure élitiste

Codage de Gray des paramètres, précision: 1e-4

3.5. Organigramme

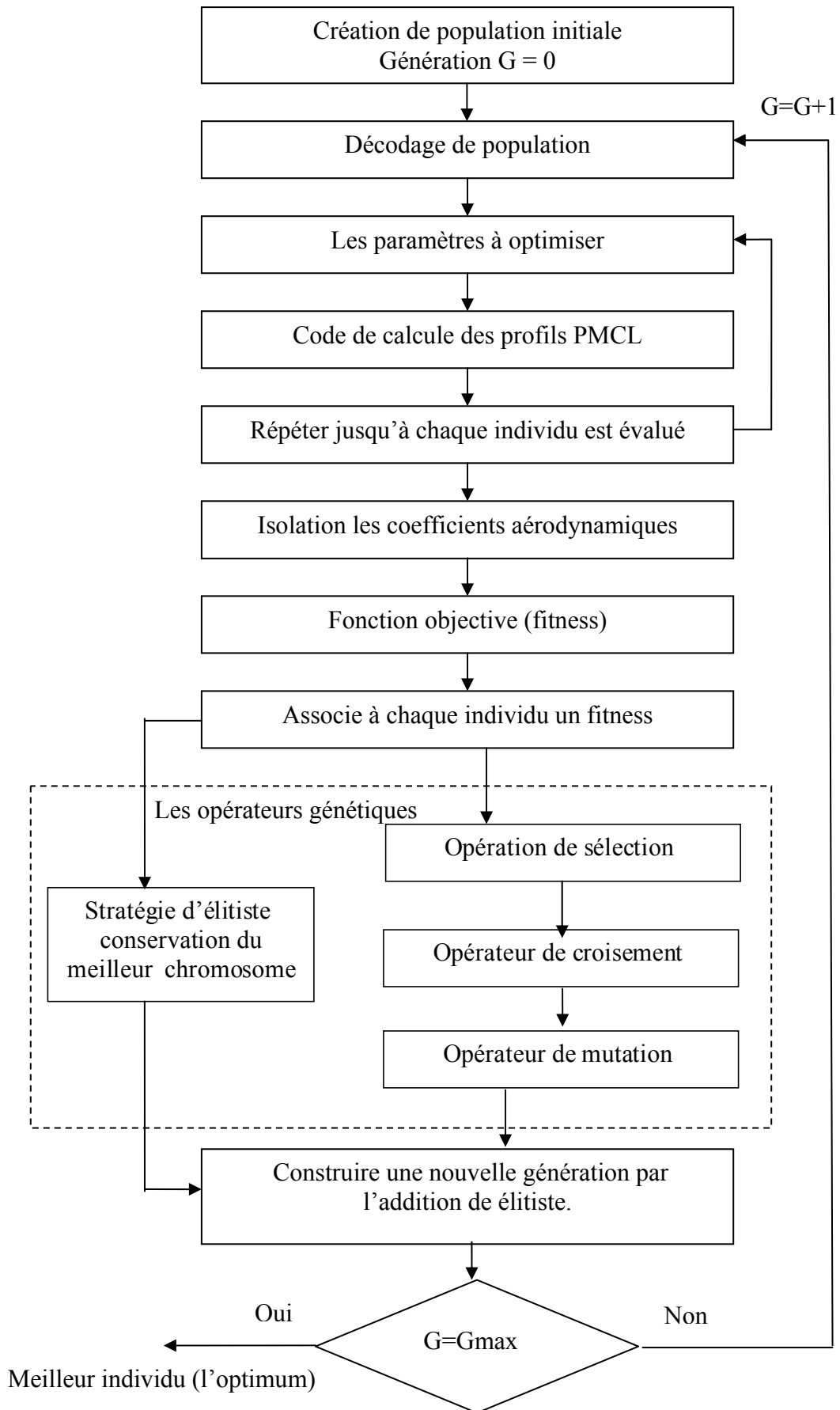


Figure 3.16 : Organigramme de calcul.

CHAPITRE 4

RESULTATS ET DISCUSSION

Nous présentons dans le présent chapitre les résultats obtenus par la mise en œuvre du programme élaboré sous Matlab dans lequel on a implémenté un algorithme génétique pour l'obtention d'un profil d'aile optimal.

4.1. Convergence de l'algorithme génétique

Cette première étude a pour but de déterminer de quelle façon l'algorithme génétique converge en fonction du nombre d'individus et du nombre de générations, afin de prédire par exemple s'il est plus intéressant de travailler avec plus d'individus ou plus de générations pour un même nombre d'évaluations. Pour cela, il suffit de comparer les valeurs de la fonction fitness, par exemple lorsque l'on a travaillé avec 30 individus et 50 générations puis avec 50 individus et 30 générations.

La fonction fitness utilisée est donnée par l'expression suivante:

$$fitness = \frac{1}{\frac{Cd}{Cl} + (Cl - Clt)^2 + (Cm - Cmt)^2}$$

Dans la figure (4.1) on a travaillé avec des nombres d'individus variant entre 10 et 50.

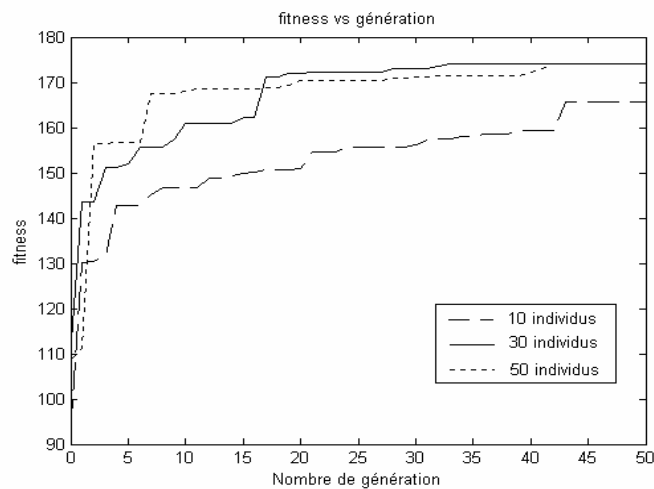


Figure 4.1 Variation de fitness en fonction de nombre d'individus de la population.

On constate tout d'abord que pour moins de 40 générations et 50 individus, l'optimum n'est pas atteint. Par contre il est atteint au bout de 33 génération lorsque on a 30 individus, puisque le temps de calcul (le nombre d'évaluations) est inférieur au premier cas. Le choix de la méthode de recherche stochastique peut expliquer cette différence.

On constate aussi que les individus sont effectivement de mieux en mieux adaptés au milieu puisque les valeurs des fonctions fitness croissent avec les générations (procédure élitiste). On remarque aussi que les résultats sont moins bons avec 50 individus qu'avec 30 puisque le nombre de calculs est plus élevé avec 50 individus, ceci est dû la aussi à la méthode de recherche stochastique.

Un nombre d'individus réduit joue en faveur de la convergence de l'algorithme génétique, néanmoins il semble qu'il est plus avantageux de mettre un grand nombre de générations au détriment du nombre d'individus.

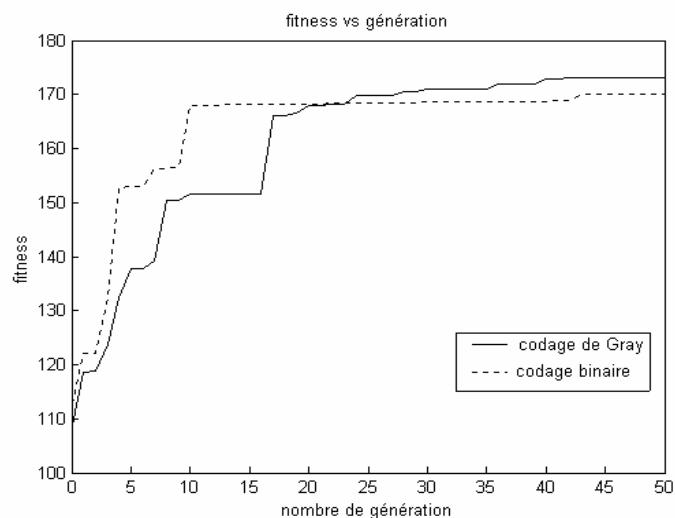


Figure 4.2 : Influence de type de codage.

La figure (4.2) montre l'importance de codage de type Gray qui favorise l'exploitation c'est-à-dire il cherche à améliorer les individus lors de l'application des opérateurs génétiques. Cependant, le codage binaire favorise plutôt l'exploration de l'espace de recherche grâce au risque d'obtention d'un individu très distant de ses parents lors d'un croisement ou d'une mutation.

4. 2. L'approche inverse

Les possibilités de l'algorithme utilisé sont démontrées en considérant un problème de conception inverse. Le but ici est d'employer l'algorithme d'optimisation pour trouver

les paramètres d'un profil d'aile. L'algorithme d'optimisation devrait trouver des valeurs pour ces paramètres près de celui donné comme profil cible tel que ceux dans le tableau (4.1).

D'abord, une première population de 20 individus a été créée aléatoirement. Notons que l'algorithme génétique trouve toujours le maximum de la fonction fitness. Par conséquent, la fonction objective pour l'algorithme génétique a été définie comme l'inverse de la fonction à minimiser. Le programme fonctionne pour 50 générations.

L'algorithme génétique qu'on a choisi pour procéder à l'optimisation mono objectif de la fonction coût (fitness), comporte une population d'individus où chaque individu se compose de 4 paramètres à optimiser.

La fonction fitness utilisée est :

$$fitness = \frac{1}{\sum_i (Cp(i) - Cpt(i))^2}$$

avec Cpt : distribution de pression cible pour le profil NACA 0012 à un angle d'incidence $\alpha = 2$ degré.

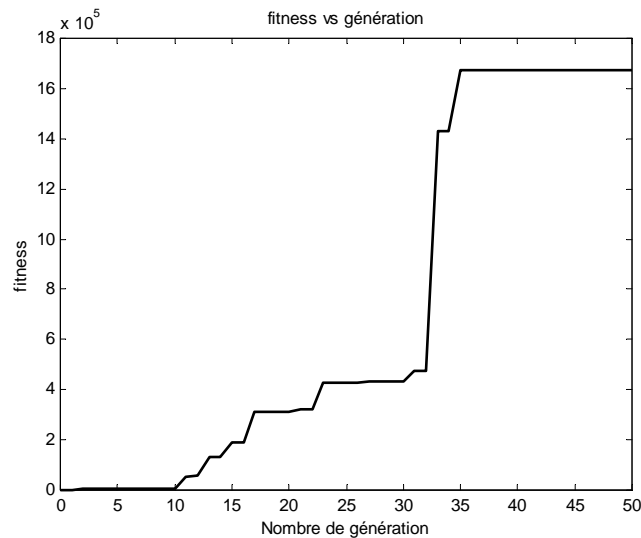


Figure 4.3 : L'évolution du fitness en fonction des générations.

La figure (4.3) montre l'évolution du meilleur fitness en fonction des générations, on constate que au bout de 35 générations l'optimum est atteint.

L'évolution de la convergence du calcul est montrée dans la figure (4.2). Il faut souligner que la valeur maximale de fitness correspond au meilleur individu dans chaque génération. La tendance de la fonction fitness sur cette figure prouve bien que l'on s'approche de l'optimum d'une génération à l'autre, ce qui montre la fiabilité de l'algorithme génétique utilisé.

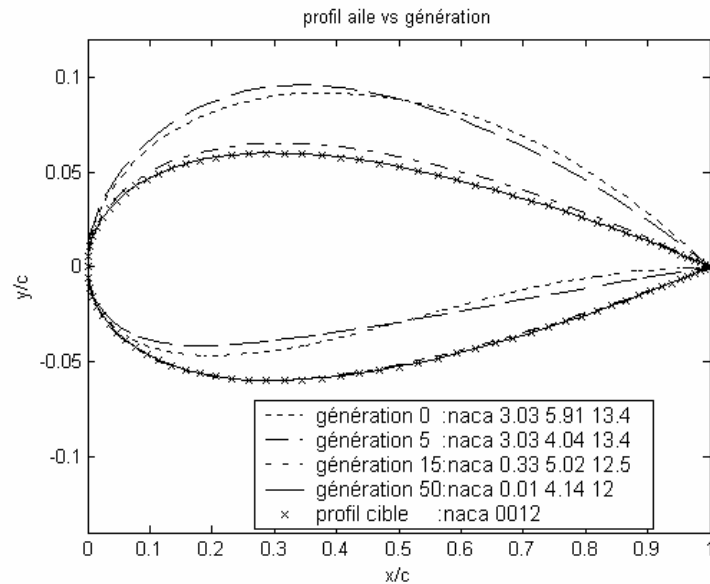


Figure 4.4 : Evolution de la forme du profil d'aile en fonction des générations.

D'après la figure (4.4), on constate que la forme optimale (trait continu) est pratiquement identique à la forme du profil cible (croix).

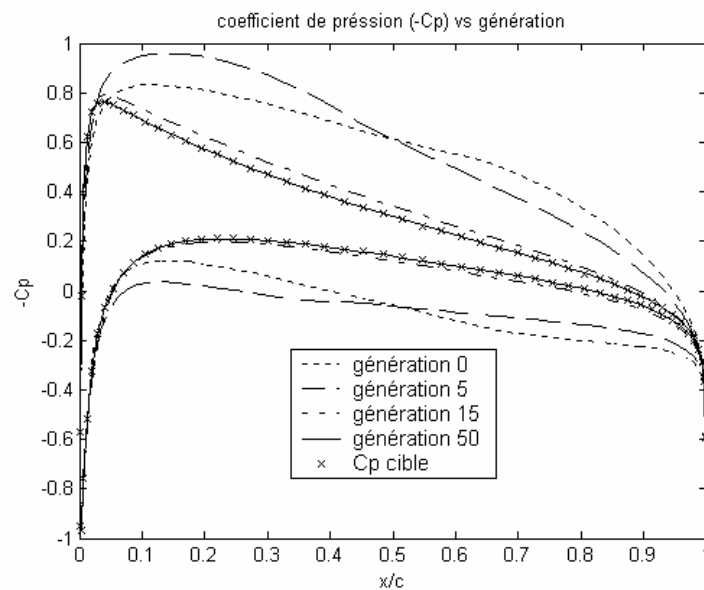


Figure 4.5 : Evolution du coefficient de pression en fonction des générations.

On a tracé sur la figure (4.5) les distributions de pression associées aux individus de la meilleure fitness au cours des générations, on constate que le meilleur individu que l'on obtient est celui en trait continu correspondant au profil NACA 0012.

Tableau 4.1 : les valeurs calculées et cibles pour l'approche inverse.

	Calculé	Cible
C_l	0.238	0.236
C_m	-0.0313	-0.0307
C_d	0.0058	0.0058

4.3. L'approche directe

L'algorithme utilisé est un algorithme génétique mono objectif de type minimisation (on cherche à minimiser la fonction objectif (fitness)). On utilise la fonction objective (fitness) selon le cas de minimisation du coefficient de traînée:

4.3.1. Représentation NACA :

On considère que le régime de croisière est atteint pour un nombre de Reynolds de l'ordre de 10^6 , condition qui correspond à de nombreux cas tests [34]. Dans ce cas précis, le concepteur cherche à diminuer la consommation de carburant. Celle-ci étant proportionnelle à la traînée, on prend comme fonction fitness (on est dans le cas d'une minimisation).

Cette représentation a été plus facile à mettre en place et ce en se référant à des études déjà faites [80] [81] [8].

$$fitness = \frac{1}{\frac{C_d}{C_l} + (C_l - C_{lt})^2 + (C_m - C_{mt})^2}$$

On a fixé le coefficient de portance cible à $C_{lt} = 0.65$ et coefficient de moment de tangage cible à $C_{mt} = -0.16$ comme des contraintes qui incluent directement dans la fonction d'adaptation (fitness). Nous avons travaillé avec 20 individus et 50 générations. L'épaisseur maximale ne devait pas être inférieure à 07 ; c'est à dire NACA XX07 (sinon on aurait obtenu un profil très fin).

La population initiale est produite aléatoirement, avec des paramètres de forme de profil d'aile et angle d'attaque d'écoulement choisis en utilisant des nombres aléatoires. Les profils d'ailes de la population initiale ont satisfait les contraintes. Par conséquent, elles peuvent être facilement considérées comme population diversifiée. Mais il serait également intéressant de voir l'exécution de l'algorithme quand la population initiale est construite à l'aide de quelques profils d'ailes connus. Pour ces raisons, une population de 20 profils d'ailes est lancées en utilisant les profils type NACA donnés dans le tableau (4.2), avec un angle d'attaque d'écoulement aléatoirement choisi.

Tableau 4.2 : profils d'ailes initiales type NACA

NACA			Angle d'incidence
4.19	4.12	07.8	2.71
3.12	6.05	15.6	1.00
7.91	4.44	19.2	1.36
4.63	6.89	11.2	2.83
.....		
.....		
4.53	5.97	13.6	1.89
1.58	7.13	14.8	0.57
5.85	5.51	10.9	1.67
5.26	4.86	18.0	1.73

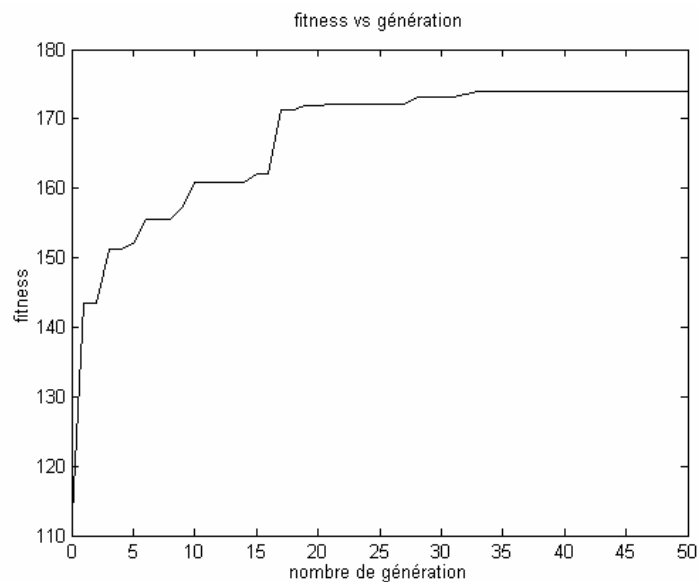


Figure 4.6 : Evolution de la fitness en fonction des générations.

La figure (4.6) montre l'évolution du meilleur fitness c'est à dire l'élite ; on constate qu'au bout de 35 générations l'optimum est atteint.

La figure (4.6) montre l'augmentation de la fitness de 113 pour le profil d'aile initial à 175 pour le profil d'aile optimal, ce qui représente une augmentation de 55 %. Il est également évident d'après le tableau (4.2) que les contraintes ont été également satisfaites. Puisque l'algorithme utilise la stratégie élitiste, ce qui donne à la courbe de meilleure fitness la propriété de croissance monotone.

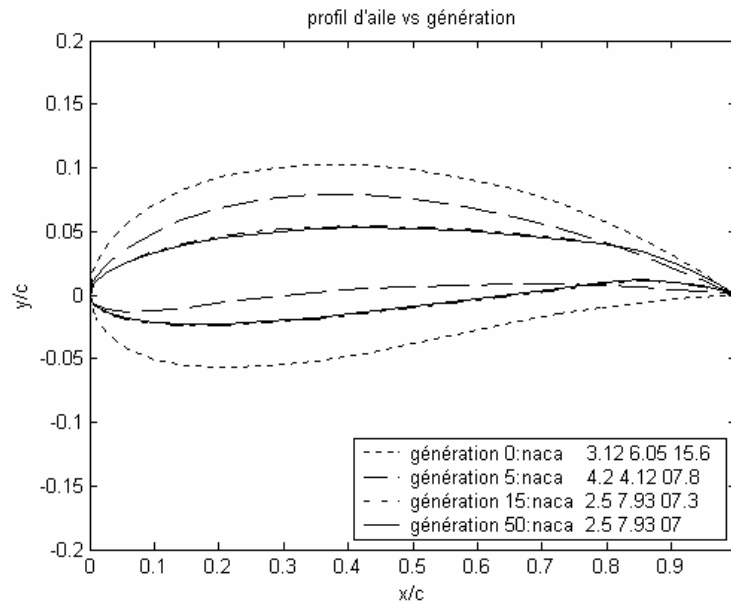


Figure 4.7.a : Les formes des profils d'aile pour différentes générations

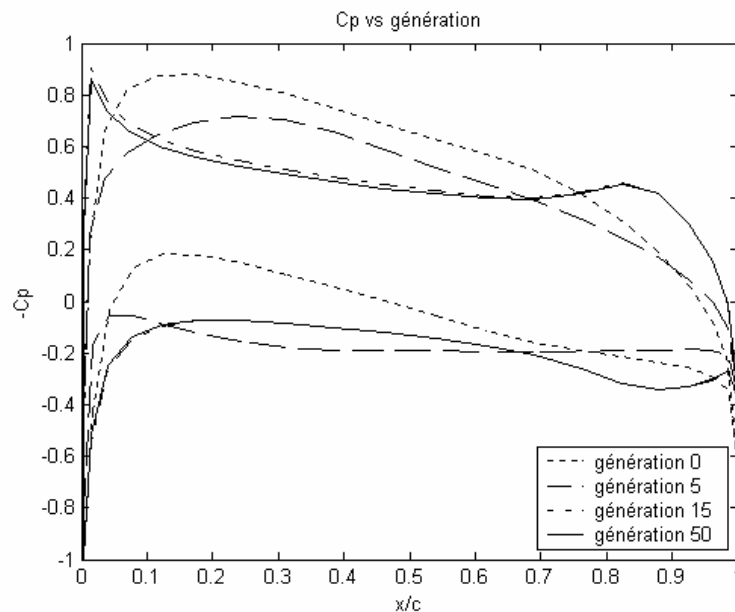


Figure 4.7.b : Les coefficients de pression des profils d'aile pour différentes générations

La figure (4.7) donne les formes des profils d'aile et leurs distributions de pression pour différentes générations.

D'après la figure (4.8.a), en isolant la forme du profil d'aile optimal trouvé, on constate qu'il a pratiquement la même forme que le profil obtenu par une autre méthode tel que illustré par la figure (4.8.b), à savoir l'optimisation d'un profil d'aile par différenciation automatique élaborée par C. Wauquiez [34] pour le même cas test (conditions de vol)

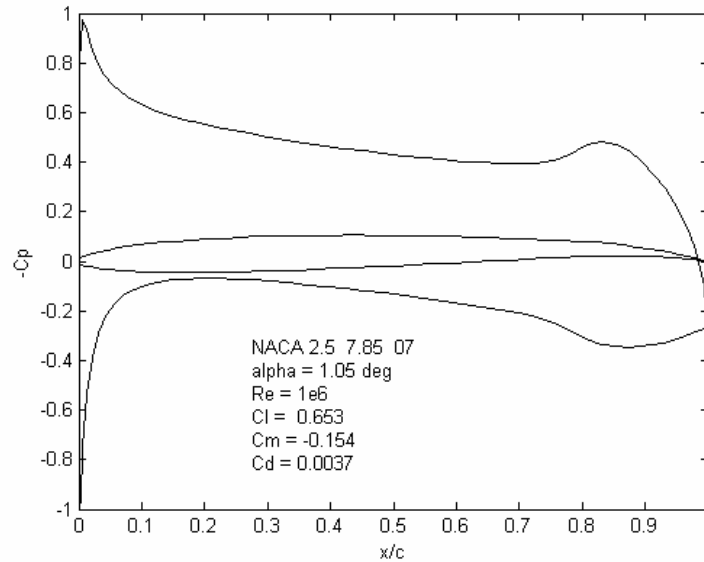


Figure 4.8.a : Forme du profil d'aile optimal obtenu et sa distribution de pression.

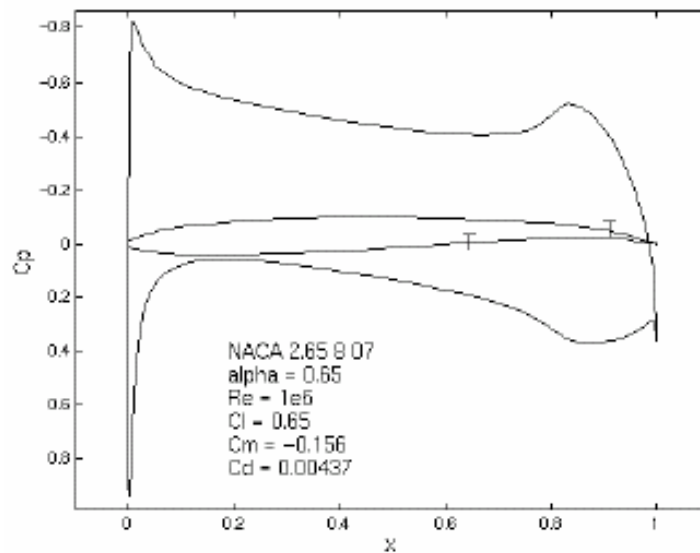


Figure 4.8.b : Forme du profil d'aile optimal et sa distribution de pression obtenu par différenciation automatique (C.Wauquiez [34]).

La figure (4.8.a) montre la forme du profil d'aile et la distribution de pression, les coefficients aérodynamiques correspondants et l'angle d'incidence optimal et ce pour une

fonction fitness donnée pour le cas d'une population de 30 individus sur 50 générations ou 1500 évaluations.

On voit que l'épaisseur maximum du profil est égale à sa valeur minimale, ce qui donne un faible coefficient de traînée. Le coefficient de portance est obtenu par à un peu de cambrure, localisé à l'arrière de profil. Cette région à l'arrière de la cambrure donne un faible coefficient de traînée.

Les valeurs de C_l , C_m et C_d qui correspondent à l'optimum sont mentionnés dans le tableau (4.3)

Tableau 4.3 : les valeurs calculées et cibles

	Calculé	cible
C_l	0.653	0.65
C_m	-0.154	-0.16
C_d	0.0037	N/A

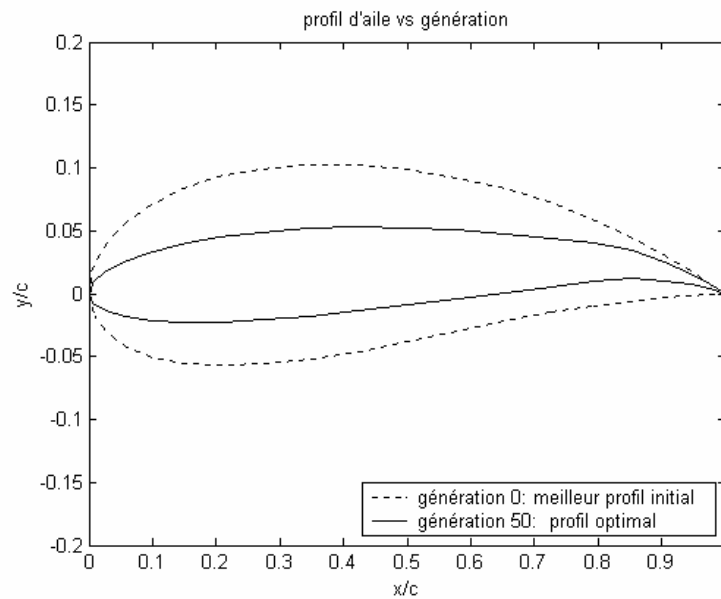


Figure 4.9.a : Formes des meilleurs profils d'aile initiale et optimal.

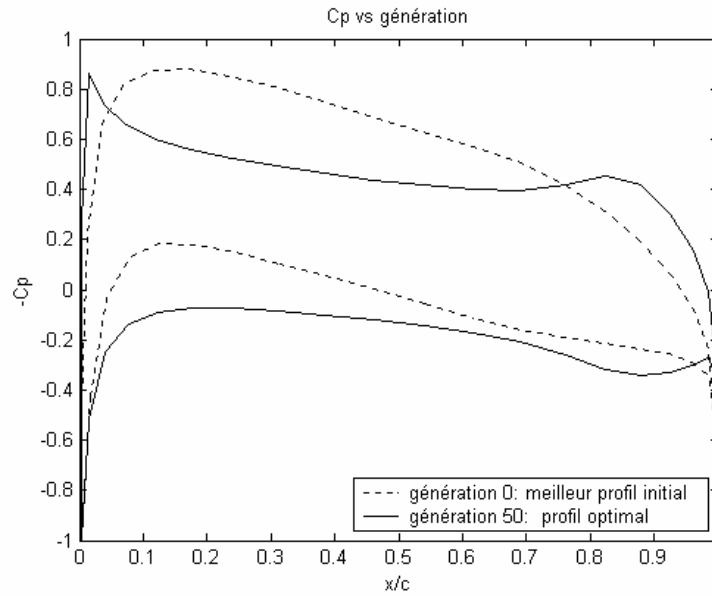


Figure 4.9.b : Distributions de pression des meilleurs profils d'aile initial et optimal.

Les meilleurs individus des populations initiales et finales et leur distribution de pression sont montrés sur les figures (4.9), qui montre que l'aire entre la partie supérieure et inférieure de la courbe de coefficient de pression donne le coefficient de portance d'un profil d'aile. Pour le meilleur profil d'aile de la population initiale plus de portance est produite près du bord d'attaque, alors qu'elle est répartie uniformément dans le cas du profil optimal obtenu. Ceci diminue nettement le coefficient de traînée.

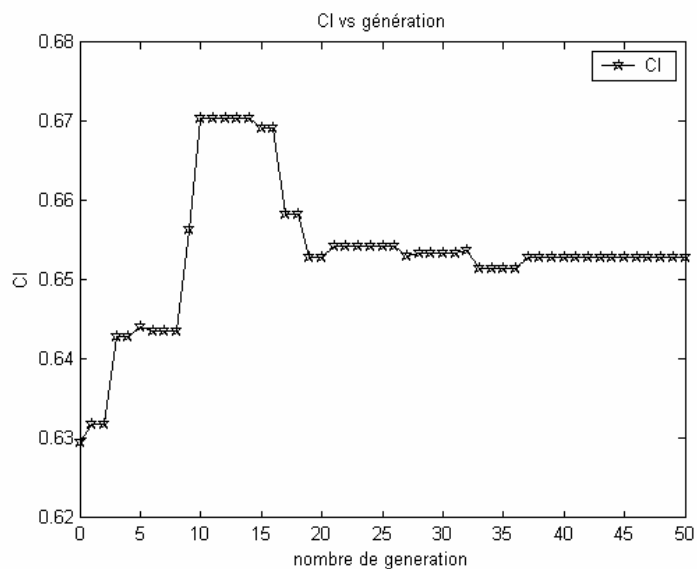


Figure 4.10 : Variation du coefficient de portance en fonction du nombre de générations

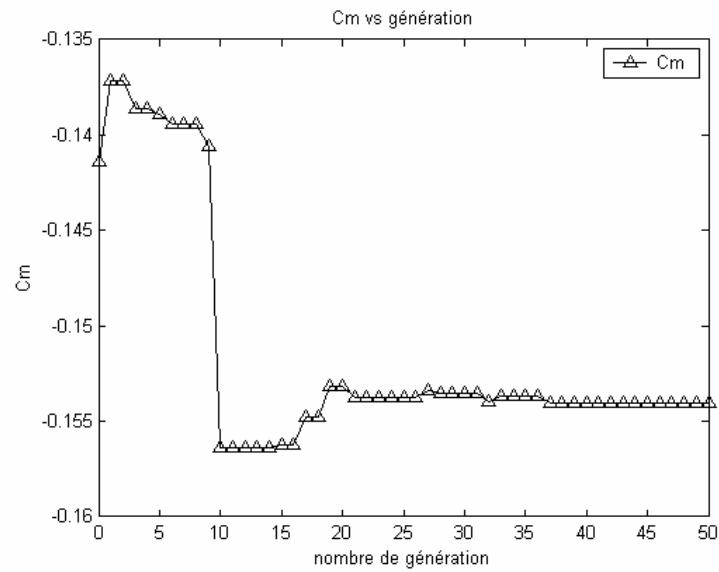


Figure 4.11 : Variation du coefficient de moment tangage en fonction du nombre de générations

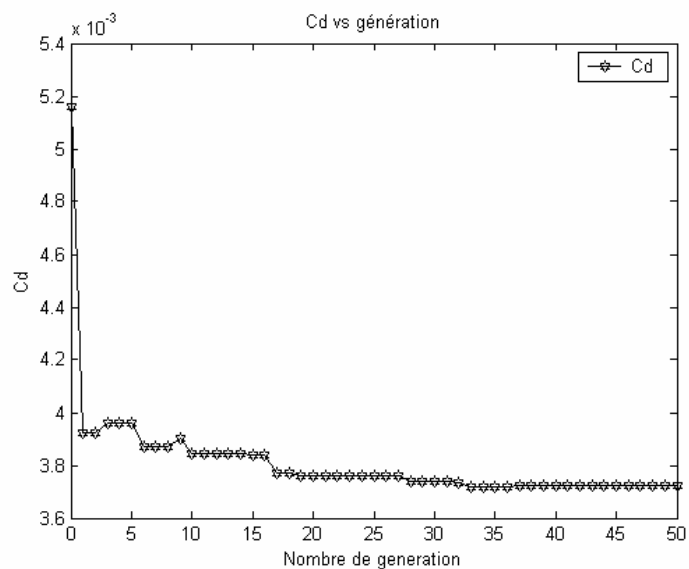


Figure 4.12 : Variation du coefficient de traînée vs le nombre de générations.

La variation des coefficients de portance, le coefficient de moment tangage et le coefficient de traînée pendant l'évolution sont montrées sur les figures (4.10), (4.11) et (4.12) respectivement.

Les figures (4.10), (4.11) et (4.12) montrent clairement que les contraintes sont satisfaites en même temps. Le profil d'aile optimal trouvé après l'évolution possède un coefficient de traînée minimal.

4.3.2. Représentation à l'aide de B-Splines

La paramétrisation du profil d'aile se fait dans ce cas par utilisation de courbes B-splines. Les courbes sont définies à l'aide de 12 points de contrôle sur le profil, 6 pour représenter l'extrados et 6 pour l'intrados, avec l'ordre $d = 6$.

On considère un écoulement dans les mêmes conditions que les précédents, ici la population initiale est produite en choisissant aléatoirement les variables de décision.

Puisque le nombre d'évaluations varie de manière exponentiel avec le nombre de paramètres à optimiser c'est pourquoi on l'augmente ; nous avons travaillé avec 50 individus et 150 générations.

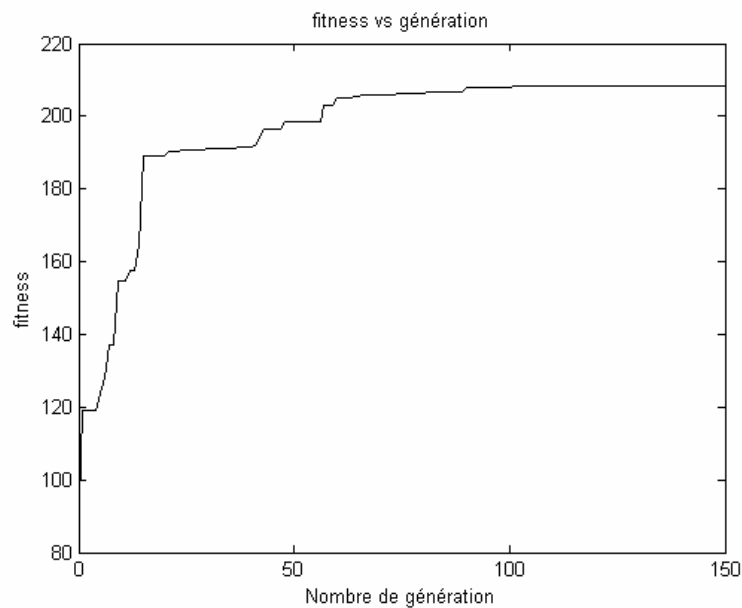


Figure 4.13 : Evolution de la fitness en fonction des générations.

La figure (4.13) représente l'évolution de la fitness en fonction du nombre de générations. Il y a convergence vers une valeur de 208 à partir de la 100^{ème} génération.

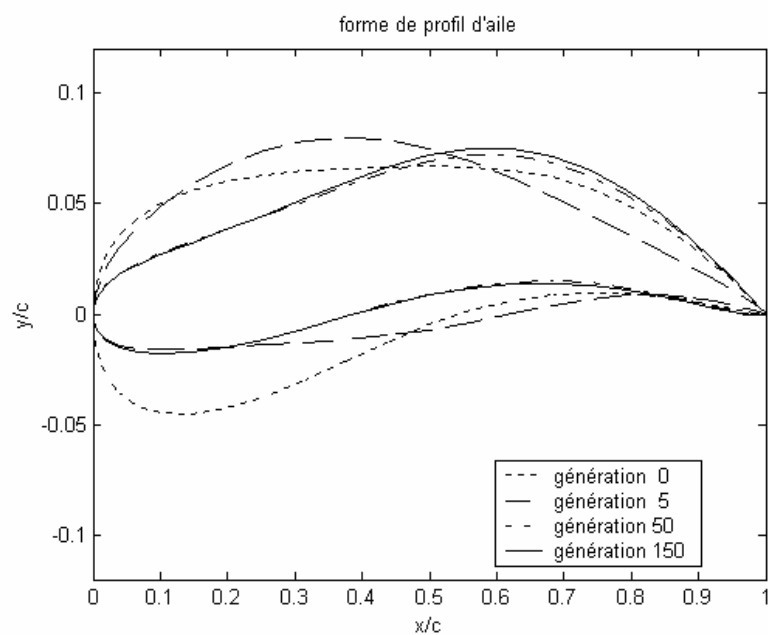


Figure 4.14.a : Formes des profils d'aile pour différentes générations.

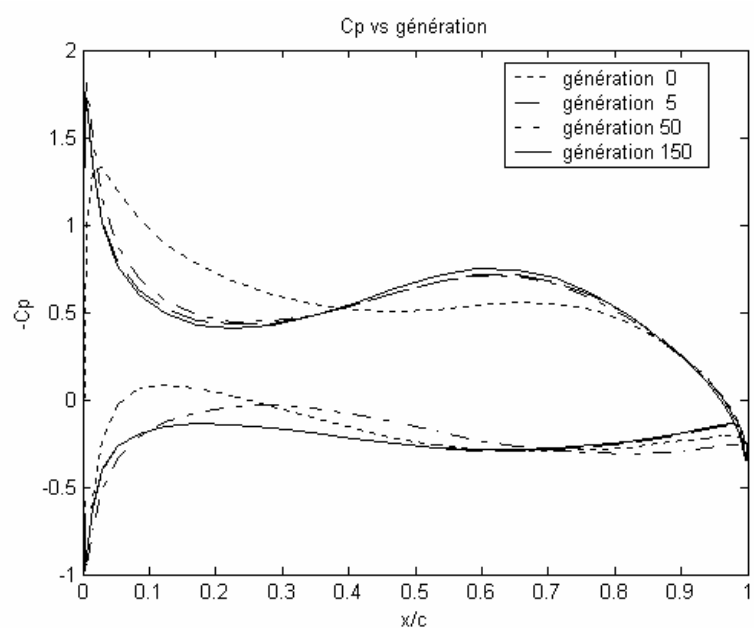


Figure 4.14.b : Coefficients de pression des profils d'aile pour différentes générations

La figure (4.14.a) montre les formes des profil d'aile et la distribution de pression figure (4.14.b) pour différente générations, angle d'incidence optimal pour une même fonction fitness que la précédente avec une population de 50 individus pendant 150 générations ou 7500 évaluations.

On a obtenu un profil dont l'épaisseur est réduite, ce qui donne un faible coefficient de traînée. Le coefficient de portance est obtenu par à un peu de cambrure, localisé à l'arrière de profil. Cet endroit arrière de la cambrure est donne une basse coefficient de traînée.

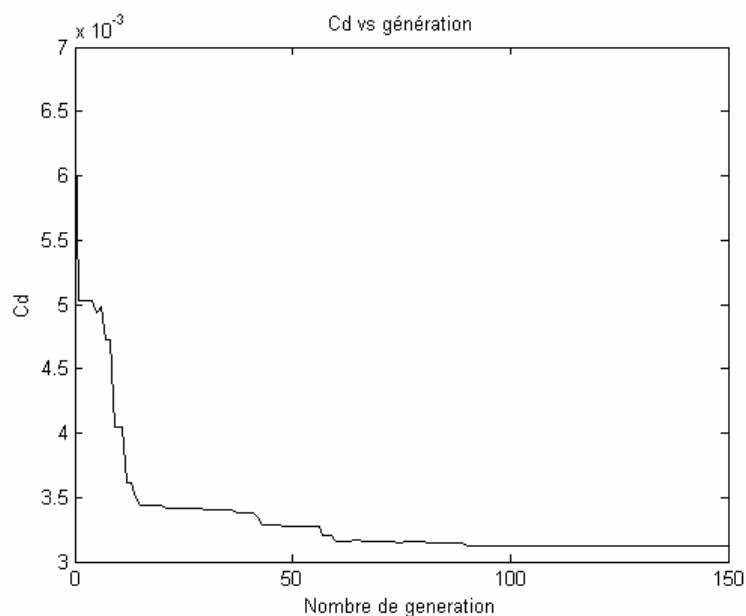


Figure 4.15: Variation du coefficient de traînée vs. le nombre de génération.

D'après la figure (4.15), la diminution de coefficient de traînée de 0.0060 pour le profil d'aile initial à 0.0032 pour le profil l'aile optimale, ce qui représente une réduction 47%.

CONCLUSION

Dans cette étude, on a réalisé l'optimisation de forme de profil d'aile selon le critère de minimisation de la traînée en régime subsonique. Cette optimisation a été réalisée à l'aide des algorithmes génétiques.

Les approches des méthodes stochastiques sont aujourd'hui utilisées dans de nombreux domaines. Elles sont intéressantes lorsque les méthodes classiques utilisant le gradient de la fonction de coût ne peuvent être utilisées ou fournissent des résultats médiocres (convergence locale). Les méthodes stochastiques sont caractérisées par une grande robustesse et la capacité de réaliser une optimisation globale. Elles ont une structure leur permettant d'utiliser les architectures parallèles, ce qui leur permet de rester compétitives en terme de temps de calcul. Parmi celles-ci, les algorithmes génétiques représentent une philosophie de recherche très générale, offrant la possibilité de résoudre des problèmes avec de nombreux optima locaux ou multicritères. Malgré les modifications apportées à l'algorithme classique, ils ont encore des difficultés à s'adapter à des problèmes réels, notamment ceux présentant de nombreuses contraintes. Un autre inconvénient pratique des algorithmes classiques est l'espace de conception qui doit être régulier. Cette nouvelle approche, qui tente de dépasser les limitations des algorithmes classiques en introduisant d'avantage de flexibilité dans leur définition, a obtenu des résultats prometteurs.

Nous avons obtenu des profils d'ailes optimaux en adoptant une stratégie évolutionniste à partir d'une population initiale sur la base de représentations de profils NACA puis en utilisant des courbes B-splines. Pour les conditions de vol choisies les profils optimaux ont été obtenus après 50 et 150 générations respectivement par minimisation du coefficient de traînée et en prenant comme fonction fitness la relation (4.1). Les profils optimaux obtenus concordent avec ceux de C. Wauquiez [34]

APPENDICE A

LISTE DES SYMBOLES ET DES ABREVIATIONS

- c : Longueur de la corde du profil d'aile.
 Cd : Coefficient de traînée
 C_D : Coefficient de dissipation.
 Cl : Coefficient de portance.
 Cl_t : Coefficient de portance cible.
 Cm : Coefficient de moment de tangage.
 Cm_t : Coefficient de moment de tangage cible.
 Cp : Coefficient de pression à la surface de profil d'aile.
 C_{pt} : Coefficient de pression à la surface de l'aile cible.
 d : Degré des fonctions de base B-splines.
 E : Vitesse d'entraînement.
 F : Fonction coût de l'optimisation (fitness).
 H : Facteur de forme
 n : Nombre de points de discrétisation.
 N : Nombre des panneaux.
 N_i^d : La $i^{\text{ième}}$ fonction de base B-Splinaire de degré d .
 q : Distribution de sources par unité de longueur.
 Q : Débit de volume dans la couche limite.
 Re : Nombre de Reynolds basé sur la corde.
 Re_θ : Nombre de Reynolds basé sur l'épaisseur de quantité de mouvement.
 u : Projection de la vitesse sur l'axe.
 u_i : Le $i^{\text{ième}}$ terme du vecteur nodal.
 u_θ : Vitesse tangentielle.
 U : Vecteur nodal de B-Splines.
 U_e : Vitesse extérieure à la couche limite.
 u_r : Vitesse radiale.
 v : Projection de la vitesse sur l'axe y .

- V_∞ : Vitesse de l'écoulement uniforme incident.
 w : Coefficient de pondération.
 x : Abscisse d'un point.
 x_c : Variable de contrôle.
 y : Ordonnée d'un point.
 α : Angle d'incidence (corps profilés soumis à un écoulement uniforme).
 δ : L'épaisseur réelle de couche limite.
 δ^* : Épaisseur de déplacement.
 Φ_∞ : Potentiel de l'écoulement uniforme.
 Φ_v : Potentiel d'une distribution de tourbillons d'intensité $\gamma(s)$.
 Φ_s : Potentiel d'une distribution de sources $q(s)$ par unité de longueur.
 ∇^2 : Laplacien.
 ∇ : Opérateur gradient.
 $\nabla \cdot$: Opérateur divergence.
 μ : Viscosité dynamique de fluide.
 ρ : Densité du fluide.
 γ : Intensité d'un tourbillon distribué.
 λ : Intensité d'une source distribuée.
 λ : Paramètre adimensionnel de gradient de pression (couche limite) Constante empirique.
 θ : Épaisseur de quantité de mouvement.
 θ^* : L'épaisseur d'énergie cinétique.
 τ_w : Contrainte de cisaillement à la paroi.

APPENDICE B

Méthode de sélection par échantillonnage stochastique de la partie restante sans
remplacement (*Stochastic remainder without replacement selection*)

On considère une population de $N = 10$ individus. On classe les individus dans un tableau T.

Indices tableau	1	2	3	4	5	6	7	8	9	10
Individus	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	I_{10}
Fitness	3.2	0.5	0.2	1.5	2.5	0.3	0.2	0.4	1.5	0.3

On construit ensuite un tableau T' de dimension K ($K < N$) contenant les indices dont le nombre de représentants est donné par la partie entière du rapport de la fitness à la moyenne des fitness F_m (ici $F_m = 1.06$); ainsi l'individu I_1 aura 3 représentations ($F_1 / F_m = 3.01$). Pour notre exemple on obtient un tableau T' de dimension $K = 7$:

Indices tableau	1	2	3	4	5	6	7
Représentants	i_1	i_1	i_1	I_4	i_5	i_5	i_9

La création du tableau étant purement déterministe on y ajoute un principe aléatoire pour ne pas éliminer complètement les mauvais individus. On étend alors le tableau T' à la dimension N de la manière suivante :

En parcourant la population, on calcule pour chacun des individus la partie fractionnaire du rapport de sa fitness à la moyenne. On obtient donc un nombre L entre 0 et 1. On effectue ensuite un tirage aléatoire (uniforme entre 0 et 1) qui permet de sélectionner l'individu considéré avec la probabilité L. Quand on arrive à l'élément N on repasse à l'élément 1.

Indices tableau	1	2	3	4	5	6	7	8	9	10
Représentants	i_1	i_1	i_1	i_4	i_5	i_5	i_9	i_1	i_3	i_6

Ensuite pour assurer une certaine hétérogénéité de la population, on fait un brassage aléatoire du tableau.

REFERENCE

1. R. Duvigneau et M. Visonneau. « Shape optimization strategies for complex applications in Computational Fluid Dynamics ». Hamburg, Germany, 14-17 May 2003.
2. Lei Xie. «Gradient-Based Optimum Aerodynamic Design Using Adjoint Methods». Blacksburg, Virginia, April 12, 2002.
3. A. Taylor et A. Oloso. «Aerodynamic design sensitivities by automatic differentiation ». AIAA 98-2536, Jun. 1998.
4. Terry, L. Hoist and Thomas, H. Pulliam1. « Aerodynamic shape optimization using a real-number-encoded genetic algorithm ». NASA, CA 94035, AIAA 2001-2473.
5. Quagliarella, D. and Delia, Cioppa, A. « Genetic Algorithms Applied to the Aerodynamic Design of Transonic Airfoils». AIAA Paper 94-1896-CP, 1994.
6. Jean-Marc Alliot. « Techniques d'optimisation stochastique appliquées à certains problèmes du trafic aérien ». Thèse doctorat 27 octobre 1998.
7. S. Obayashi and S. Takayashi. « Genetic optimization of target pressure distribution for inverse design method ». AIAA, CA, Jun 19-22 1995.
8. S. Ziaei-Rad , A.R. Pishevar. «Wing Planform Design Using Stochastic Optimization Techniques». DME, Isfahan University of Technology, Isfahan, Iran.
9. Nelder, J. A. and Mead, R. «A simplex method for function minimization». Comput. J. 7:308-13, 1965.
10. Roland Tapiéro. «Optimisation numérique ». Cours, Université Claude Bernard, F69622 Villeurbanne Cedex, France, 2003.
11. Flannery, B.P., S.A. Teukolsky, & W.T. Vetterling. « Numerical Recipes ». W.H. Press, Cambridge University Press, 32 East 57th St., New York, NY 10022. 1992.
12. N. Jussien et C. Guéret. « Utilisation du backtrack intelligent dans un branch-and-bound Application au problème d'Open-Shop ». EMN, Nantes Cedex 03 France, 1998.
13. L. Siebenmann, L. Guillou et H. Hähl. « Les voisinages ouverts réguliers: critères homotopiques d'existence ». ASENSA, Sér. 4, 7 N° 3 (1974), p. 431-461,1974.

14. Jin-Kao Hao et al. « Méthaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes ». LERIA, U.F.R. Sciences, Revue d'Intelligence Artificielle Vol : No. 1999
15. A. MILLET. «Méthodes de Monte-Carlo». LPMA, 75634 Paris Cedex 13, France.
16. S. Kirkpatrick, C.D. Gelatt, P.M. Vecchi. «Optimization by simulated annealing ». Science 220: 671-680, 1983.
17. F. Glover. «Future paths for integer programming and links to artificial intelligence ». Computers and Operations Research 13: 533-549, 1986.
18. I. N. Vladimir, C. P. Georg. « A Branch and Bound Method for Stochastic Global Optimization ». IIASA, Austria. 1996.
19. T. Bäck, F. Hoffmeister, H-P. Schwefel. «A Survey of evolutionary strategies». Proc. of 4th Intl. (ICGA'91), R. Belew, L. Booker (Eds.), Morgan Kaufmann, p. 2-9, 1991.
20. D.B. Fogel. «Evolutionary programming: an introduction and some current directions». Statistics and Computing 4: 11-129, 1994.
21. John R. Koza. «Genetic Programming». Computer Science Department Stanford University, 258 Gates Building, Stanford, California 94305 USA, 1997.
22. Anderson, W. K. and Venkatakrishnan. « Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation ». AIAA Paper 97-0643, 1997.
23. Nielsen, E. J. and Anderson, W. K. « Aerodynamic Design Optimization on Unstructured Meshes Using the Navier-Stokes Equations ». AIAA, Vol. 37, pp. 1411-1419, No. 11, 1999.
24. Elliott, J.P. « Practical 3D aerodynamic design and optimization using unstructured grids ». AIAA 96-4170,1996.
25. M. Nemec and D. W. Zingg. « Towards Efficient Aerodynamic Shape Optimization Based on the Navier-Stokes Equations ». Toronto, ON, M3H 5T6, Canada.
26. Pueyo, A. and Zingg, D. « Efficient Newton-Krylov Solver for Aerodynamic Computations ». AIAA Journal, Vol. 36, No. 11, 1998, pp. 1991-1997.
27. T. Turgeon and D. Pelletier. « A continuous sensitivity equation approach to optimal design in mixed convection ». AIAA 99-3625,1999.
28. P. Wong and D.W. Zingg. « Aerodynamic Optimization Using the Flow Sensitivity Approach ». Downsview, Ontario, Canada, M3H 5T6.
29. W. K. Anderson and V. Venkatakrishnan. «Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation ». AIAA paper 97-0643, Jan 1997.

30. R. M. Hicks and P. A. Henne. « Wing design by numerical optimization». AIAA 77-1247, Aug. 22-24 1977.
31. K. D. Lee and S. Eyi. « Transonic airfoil design by constrained optimization». AIAA 91-3287, Sept. 23-25, 1991.
32. W. K. Anderson, J. C. Newman, D. L. Whitfield, and E. J. Nielsen. « Sensitivity analysis for Navier-Stokes equations on unstructured meshes using complex variables». AIAA Journal 39, n°1, 56-63, 2001.
33. Newman et al. «Multidisciplinary Sensitivity Derivatives Using Complex variables». MSSU-COE-ERC-98-08, July 1998
34. Christian Wauquier. «Shape Optimization of Low Speed Airfoils using MATLAB and Automatic Differentiation ». Licentiate's Thesis Royal Institute of Technology Department of Numerical Analysis and Computing Science, Stockholm, 2000.
35. L. L. Green, P. A. Newman, and K. J. Haigler, « Sensitivity derivatives for advanced CFD algorithm and viscous modeling parameters via automatic differentiation» J. Comput, Phys 125, 31-3324. 1996
36. P. Sundaram, S. Agrawal, and J. O. Hager. « Aerospace vehicle MDO shape optimization using ADIFOR 3.0 gradients». AIAA 2000-4733, Sept. 6-8 2000.
37. M. M. Putko, A. C. Taylor, P. A. Newman, and L. L. Green, « Some advanced concepts in discrete aerodynamic sensitivity analysis ». AIAA 2001-2529, CA, Jun. 2001.
38. Quagliarella, D. and Cioppa, A. D. « Genetic Algorithms applied to the Aerodynamic Design of Transonic Airfoils». AIAA-94-1896-CP, June 1994.
39. Yamamoto, K. and Inoue, O.«Applications of Genetic Algorithm to Aerodynamic Shape Optimization ». AIAA Paper 95-1650-CP, CA, pp. 43-51, June 1995.
40. Poloni, C., Mosetti, G. and Contessi, S. « Multi Objective Optimization by GAs: Application to System and Component Design ». John Wiley & Sons, Ltd, Chichester, U.K., pp. 258-264, 1996.
41. Cao, H. V. and Blom, G. A. « Navier-Stokes/Genetic Optimization of Multi-Element Airfoils ». AIAA 96-2487, June 1996.
42. Obayashi, S. and Oyama, A. «Three-Dimensional Aerodynamic Optimization with Genetic Algorithms». John Wiley & Sons, Ltd, Chichester, U.K., pp.420-424, 1996.
43. Oyama, A. «Multidisciplinary Optimization of Transonic Wing Design Based on evolutionary Algorithms Coupled with CFD solver». ECCOMAS 2000, 2000.
44. Uelschen, M. and Lawrenz, M. « Design of Axial Compressor Airfoils With Artificial Neural Networks and Genetic Algorithms ». AIAA Paper 2000-2546, 2001.

45. Akmandor, S. and Oksuz, O. «Aerodynamic Optimization of Turbomachinery Cascades Using Euler/Boundary-Layer Coupled Genetic Algorithms». AIAA Paper 2001-2577, 2001.
46. Vicini, A. and Quagliarella, D. «Inverse and Direct Airfoil Design Using a Multiobjective Genetic Algorithm». AIAA, Vol. 35, pp. 1499-1505, 1997.
47. Obayashi, S., Yamaguchi, Y. and Nakamura, T. « Multiobjective Genetic Algorithm for Multidisciplinary Design of Transonic Wing Planform». J. of Aircraft, Vol. 34, 1997.
48. Sasaki, D., Obayashi, S., Sawada, K. and Himeno, R. « Multiobjective Aerodynamic Optimization of Supersonic Wings Using Navier-Stokes Equations ». ECCOMAS 2000, Barcelona, Spain, Sept. 11-14, 2000.
49. Oyama, A. « Multidisciplinary Optimization of Transonic Wing Design Based on Evolutionary Algorithms Coupled with CFD Solver ». ECCOMAS 2000, Barcelona, Spain, Sept. 11-14, 2000.
50. Oyama, A. « Wing Design Using Evolutionary Algorithms ». PhD Thesis, Dept. of Aeronautics and Space Engineering, Tohoku University, Sendai, Japan, March 2000.
51. A. Oyama and S. Obayashi. «Transonic axial-flow blade shape optimization using evolutionary algorithm». AIAA 2002-5642, 2002.
52. Guido Dietz, Ralph Voß, and Roeland De Breuker. «Airfoil optimization based on an evolution strategy with respect to aeroelasticity». Institut für Aeroelastik des DLR in der Helmholtz-Gemeinschaft Bunsenstr. 10, D-37073 Göttingen, Germany.
53. D.J. Doorly, J. Peiro' and J-P. Oesterle, «Optimization of aerodynamic and Coupled Aerodynamic –structurel design using parallel genetic algorithms » AIAA paper 96-4027,1996.
54. I. De Falco, R .Del Balio. «A parallel genetic algorithm for transonic airfoil optimization». IRSIP, CNR, Via P, Castellino 111,80131, Napoly ,Italy.
55. X. Wang and M. Damodaran. «Aerodynamic Shape Optimization Using computational Fluid Dynamics and Parallel Simulated Annealing Algorithms». AIAA Vol. 39, No. 8, Aug 2001.
56. H. Lièvre, J.A. Désidéri et A. Habbal. «Optimisation numérique de profils d'aile par Algorithmes Génétiques et Jeux de Nash». INRIA N° 4275 septembre 2001.
57. Vicini, A. and Quagliarella, D. « Airfoil and wing design through hybrid optimization strategies». CIRA,81043 Cupua,Italy,1998.
58. Marco .N.et Beux .F. « Multilevel optimization: application to shape optimum design with a One-Shot method». INRIA N° 2068, 1993.

59. Kuruvila et Shlomo. «Airfoil Optimization by the One-Shot Method». Von Karman Institute for Fluid Dynamics, Rhode-Saint-Genese, Belgium, April 1994.
60. Christophe Held. «Optimisation de formes en aérodynamique sans gradient par résolution simultanée». INRIA N° 3864. Janvier 2000.
61. J. Holland. «Outline for a logical theory of adaptive systems». Journal of the Association of Computing Machinery, 3, 1962.
62. David Goldberg. «Genetic Algorithms». Addison Wesley, ISBN: 0-201-15767-5, 1989.
63. D.E. Goldberg. «Real-coded genetic algorithms, virtual alphabets and blocking. Complex Systems», 5:139–167, 1991.
64. L.J. Fogel, A.J. Owens, and M.J. Walsh. «Artificial Intelligence Through Simulated Evolution». Wiley and sons. NY, 1966.
65. Z. Michalewicz. «Genetic algorithms + Data Structures = Evolution Programs». Springer-verlag, 1992.
66. X. Yin and N. Gerny. «A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization». In Proceedings of the Artificial Neural Nets and Genetic Algorithms, 1993.
67. L. Ingber and B. Rosen. «Genetic algorithms and very fast simulated re-annealing». Mathematical Computer Modeling, 16(11):87–100, 1992.
68. S.W. Mahfoud and D.E. Goldberg. «Parallel recombinative simulated annealing: A genetic algorithm». Illigal report 92002, University of Illinois, Urbana, IL 61801-2996, April. 1992.
69. J. Horn and N. Nafpliotis. «Multiobjective optimization using the nitched pareto genetic algorithm». Illigal Report 93005, University of Illinois at Urbana, 1993.
70. A. Bertoni and M. Dorigo. «Implicit parallelism in genetic algorithms». Artificial Intelligence, 61(2): 307–314, 1993.
71. H. Muhlenbein. «Parallel genetic algorithms, population genetics and combinatorial optimization». In Proceedings of the Third International Conference on Genetic Algorithms, 1989.
72. A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. «Pvm 3 user's guide and reference manual». Technical report, Oak Ridge National Laboratory, 1994.
73. M. Drela and M. B. Giles. «Viscous-Inviscid Analysis of Transonic and Low Reynolds Number Airfoils». AIAA Journal, 25 (10), Oct. 1987.

74. Nicholas, K. Borer. « Design and Analysis of Low Reynolds Number Airfoils». Final Project, ATH 6514: Industrial Math, December 2002.
75. Ilan Kroo . PANDA. « A Program for Analysis and Design of Airfoils». P.O. Box A-L, Stanford, CA 94 309,1994.
76. E. Alpman. «Airfoil shape optimization using evolutionary algorithms ».University Park, PA, 16802.
77. J. TARDIF. « Modélisation numérique de l'aérodynamique interne d'une roue de ventilateur centrifuge». Maîtrise es sciences appliqué, Sherbrooke, CANADA, Déc 1999.
78. Hess-Smith. «Panel Method». AA200b, Lecture 3, January 13-18, 2005
79. Michele, Andreoli., Ales Janka. and Jean-Antoine, Désidéri. « Free-form-deformation parameterization for multilevel 3D shape optimization in aerodynamics». INRIA, N° 5019 November 2003.
80. A. Mitra. « Design of transonic airfoil using multiobjective genetic algorithms ». DMESU of Stanford, California 94305.
81. F. Zhang, S. Chen and M. Khalid. « Multi-Point Optimization of Transonic Wing by Real-Coded Genetic Algorithm ». IAR,NRC, Ottawa, K1A, Ontario, Canada.
82. Abbot, I. H., Von Doenhoff, A. E. «Theory of Wing Sections ». Dover Publications Inc., ISBN 486-60586-8, 1959.
83. Mehmet Celikbas. « B-splines pour l'optimisation de forme». INRIA, N° 4968, Octobre 2003.