Optimisation des conditions aux frontières par algorithmes génétiques

Jean-Luc Marcelin

Laboratoire sols, solides, structures
URA CNRS 1511
BP 53
38041 Grenoble cedex 9

RÉSUMÉ. *Une stratégie d'optimisation des conditions aux frontières pour divers problèmes de mécanique des structures à géométrie fixe est présentée. Elle consiste à coupler un programme général d'analyse par éléments finis avec un programme d'optimisation par algorithmes génétiques (reposant sur les lois de la sélection naturelle). Du fait des spécificités de la méthode des éléments finis (raideur calculée une fois pour toutes), nous montrons sur des exemples variés que la stratégie présentée peut se révéler intéressante.*

ABSTRACT. *This work examines the possibility of using a stochastic method, called the genetic algorithm for optimization of boundary conditions in finite elements calculations. The examples show that using genetic algorithms in order to optimize boundary conditions is an efficient way.*

MOTS-CLÉS : *optimisation, conditions aux limites, algorithmes génétiques.*
KEY WORDS : *optimization, boundary conditions, genetic algorithms.*

1. Introduction

Dans de nombreux cas d'application (prise de pièce en usinage, vibrations d'une structure mécanique par exemple), l'optimisation des conditions aux frontières (localisation et nature des conditions limites) peut apporter une amélioration intéressante du comportement mécanique de la structure étudiée. Ces structures sont le plus souvent calculées par la méthode des éléments finis, et dans ce cadre le calcul des sensibilités par rapport aux conditions aux limites demeure un exercice assez délicat du fait de la nature discrète du problème [KEU 92] et [SON 93], à un tel point qu'en optimisation de forme par exemple le problème des conditions aux limites est le plus souvent éludé. De plus, les méthodes déterministes d'optimisation dites méthodes de gradient nécessitent un calcul fiable de ces sensibilités. Actuellement sont en vogue un certain nombre d'autres méthodes d'optimisation dites stochastiques ou probabilistes comme la méthode du recuit simulé ou celle des algorithmes génétiques dont les principaux avantages sont une convergence assurée sans utilisation de dérivées et pour des fonctions éventuellement à variables discrètes et non dérivables. Les détracteurs de ces méthodes mettent en avant non sans raison le volume considérable des calculs, surtout dans le cas d'une méthode d'analyse de type élément fini. En restant pour le moment dans le cadre de géométries fixes, nous allons cependant démontrer que le problème de l'optimisation des conditions aux frontières en couplant un algorithme génétique à la méthode des éléments finis est tout à fait réalisable (le problème de l'optimisation des conditions aux limites en optimisation de forme fera l'objet de travaux ultérieurs). En effet, dans le cas d'une forme fixe et du fait des spécificités de la méthode des éléments finis, le volume des calculs peut être considérablement réduit. Les raisons essentielles que nous développerons par la suite en sont les suivantes: la matrice raideur est calculée et assemblée une fois pour toutes; par ailleurs dans le cas d'une structure pour laquelle certaines conditions aux limites peuvent être fixes et d'autres variables (c'est-à-dire rentrant dans le cadre de l'optimisation), il serait possible de triangulariser la matrice raideur une fois pour toutes, et de tenir compte des conditions aux limites variables grâce à une procédure de pénalisation de la fonctionnelle énergétique à minimiser par les conditions aux limites. Dans ces conditions, même si le nombre d'analyses reste important, les temps de calcul resteront raisonnables, car il ne s'agira pas d'analyses complètes à chaque fois. Nous montrons au travers d'exemples variés que la procédure mise en oeuvre permet d'optimiser les conditions aux frontières avec une efficacité certaine.

2. Les algorithmes génétiques

2.1. Le cadre de l'étude

A plusieurs reprises la méthode des algorithmes génétiques a été utilisée dans le cadre de problèmes divers de mécanique. Ces algorithmes se sont révélés très efficaces, comme par exemple dans le cas de la maximisation de l'amortissement de poutres ou plaques composites [MAR 94] et [MAR 95] ou de problèmes un peu différents [TRO94]. L'intérêt de ces algorithmes a été aussi montré dans le cas difficile de l'optimisation des engrenages [MAR 95]. On trouve dans la suite quelques éléments de compréhension de la méthode. Pour des compléments, on peut consulter les références [GOL 89] et [HOL 92].

2.2. Que sont les algorithmes génétiques ?

Les algorithmes génétiques sont des algorithmes d'optimisation. Ils cherchent les solutions optimales d'un problème donné en simulant le processus d'évolution et d'adaptation des organismes vivants. Le problème est traduit en termes de maximisation d'une fonction objectif dans un espace à N dimensions. Le point de départ est une population d'individus choisis au hasard qui sont codés sous forme de nombres binaires (par exemple) appelés chromosomes. Après ce départ, l'algorithme génétique génère de façon plus ou moins aléatoire de nouvelles populations, formées d'individus de plus en plus aptes à s'adapter à un environnement bien défini, en utilisant trois opérateurs différents qui sont :

- la reproduction,
- le croisement,
- la mutation.

La figure 1 donne une vue d'ensemble de la méthode pour des chromosomes comprenant par exemple des 0, 1 et par extension des 2.

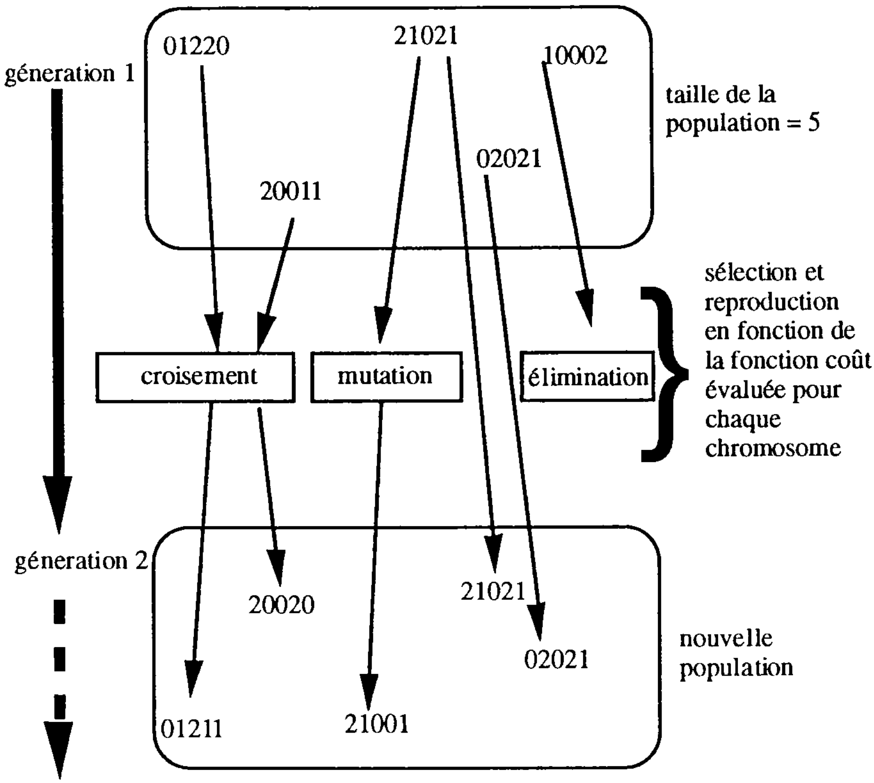

Figure 1. *Représentation schématique de l'algorithme génétique simple*

2.3. La reproduction

La reproduction est un processus dans lequel chaque individu de la population est reproduit en fonction de son aptitude à maximiser l'objectif. Leurs chaînes codées ont plus de chance d'être copiées dans les générations suivantes, et de subir les autres opérations : le croisement et la mutation. La probabilité de la sélection d'un individu pour la reproduction augmente avec son coût. Elle peut être définie par exemple en comparant le coût d'un individu à ceux des autres individus de la même population ; pour un individu i dont le coût est fi, la probabilité de sélection est

$$P \text{ sélection } i = fi / \sum fi$$

Exemple : pour une fonction coût : $f(x) = x^2$ dans $[0, 31]$, et une population de quatre individus, les probabilités de sélection de chaque individu sont présentées dans le tableau ci dessous.

N°	codage	x	coût	P sélection
1	01101	13	169	0.144
2	11000	24	576	0.492
3	01000	8	64	0.055
4	10011	19	361	0.309

Dans la reproduction de cette population, on choisit de garder les deux meilleurs individus, mais il faut savoir que les meilleurs n'ont qu'une grande probabilité d'être sélectionnés et peuvent disparaître d'une génération à l'autre

A = 11000
B = 10011

2.4. Le croisement

Cet opérateur est appliqué à deux individus sélectionnés pour la reproduction, pour en créer d'autres. Le processus s'effectue en deux étapes :
D'abord, un rang K est choisi aléatoirement entre 1 et L, avec L la longueur des chaînes codées.
Ensuite, la première partie d'une chaîne (de longueur K) est collée à la deuxième partie de l'autre chaîne (de longueur L, K).

Exemple : pour les deux individus sélectionnés précédemment :

A = 11000
B = 10011

Si la valeur de k = 4 (L = 5) alors les individus résultants sont :

A' = 11001
B' = 10010

2.5. La mutation

La reproduction et le croisement sont les exploitations des "bonnes" qualités des générations passées. Ils peuvent bloquer le processus d'optimisation sur les maxima locaux si des opérateurs d'exploration de l'espace solution ne sont pas considérés. C'est le cas de la mutation qui consiste simplement à effectuer une permutation sur la chaîne codée. La probabilité de la mutation est prise dans nos calculs plus faible car le procédé est moins fréquent que le croisement et la reproduction (dans l'évolution des espèces naturelles).

Par exemple le digit A = 11001 est transformé en A' =11010

2.6. Différences entre les algorithmes génétiques et les techniques classiques d'optimisation

Ces algorithmes sont différents des autres méthodes sur plusieurs points

- Ils utilisent des codages des variables et non les variables elles mêmes,
- Ils travaillent sur la population, c'est-à-dire sur un ensemble de solutions,
- Ils n'ont besoin que de la fonction objectif, et non des dérivées,
- Ils utilisent des règles probabilistes et non déterministes.

2.7. Avantages des algorithmes génétiques

Contrairement à ce que l'on pourrait penser, la méthode des algorithmes génétiques n'a rien de magique. Elle fait partie de la classe des méthodes dites "stochastiques". La plus connue d'entre elles est la méthode déjà ancienne du recuit simulé. L'avantage essentiel de ces méthodes est qu'elles opèrent simultanément sur un échantillon de l'espace des solutions. La méthode génétique diffère du recuit simulé par les opérateurs qu'elle utilise pour faire évoluer cet échantillon de population. En tout cas, et cela est démontré dans [GOL 89], la convergence est toujours assurée vers un extrèmum qui n'est pas forcément l'extrèmum absolu, mais qui a plus de chances de l'être que si l'on utilise une méthode classique de gradient. En effet, une méthode stochastique explore l'espace des solutions plus largement.

3. L'optimisation des conditions aux frontières

L'optimisation consiste à coupler un programme de calcul par élément finis standard (appelé dans la suite programme d'analyse) avec l'algorithme génétique.

3.1. Les programme d'analyse et d'optimisation

Le programme d'analyse est un code standard EF. Il suffit simplement d'appeler ce code chaque fois que l'algorithme génétique doit évaluer la fonction coût pour un chromosome donné. Ceci est réalisé pour chaque individu de la population; c'est ainsi, par exemple, que pour 20 individus et 30 générations il y aura 600 semi-analyses par éléments finis (rappelons que la raideur totale est calculée une fois pour toutes), ce qui est relativement peu vis-à-vis des 2^{20} solutions possibles. Par contre, pour les différents tests effectués, en particulier ceux présentés plus loin, il n'a pas été nécessaire de mettre en oeuvre une stratégie de pénalisation de la fonctionnelle "énergie potentielle totale" par les conditions aux limites imposées, car la convergence a été suffisamment rapide. Le travail du programmeur consiste simplement à écrire un programme "pré-analyse" capable de décoder le chromosome en question et de modifier automatiquement le fichier de données éléments finis en conséquence, puis un programme "après-analyse" capable d'extraire la fonction coût du fichier résultat éléments finis. Ces deux programmes, ainsi que l'appel du code éléments finis, sont implantés dans le programme génétique qui pilote le processus. Dans notre cas, nous avons deux versions de l'algorithme génétique: l'une écrite en Pascal et qui fonctionne sur plateforme PC sous DOS (le code élément fini appelé est un logiciel simple sur PC), l'autre écrite en langage C et qui fonctionne sur station SILICON GRAPHICS sous UNIX (le code élément fini utilisé est le code ANSYS). Suivant la nature du problème et sa dimension, on utilise l'un ou l'autre des programmes. Ces deux programmes sont des algorithmes génétiques de base, c'est-à-dire utilisant les opérateurs élémentaires décrits dans la partie 2.

3.2. Choix du codage et de la fonction objectif

La difficulté du problème est double : il s'agit d'abord de mettre en place un codage de la solution sous forme de chromosome qui soit simple et efficace; ensuite puis de mettre au point une fonction objectif. Le codage le plus généralement utilisé est simple et naturel (on en verra des variantes dans les exemples): il peut reprendre le codage communément utilisé pour les conditions aux limites dans les programmes éléments finis, à savoir 0 pour un degré de liberté libre et 1 pour un degré de liberté fixe. Les différents codes des noeuds concernés sont rangés bout à bout dans un chromosome qui est donc constitué de n digits binaires correspondant aux n degrés de liberté pouvant être fixés. Quant à la fonction objectif, elle dépend du problème que l'on se pose. Dans les deux premiers exemples, on est en statique et on cherche à minimiser le déplacement maximum, ou bien à minimiser une déformation ou une contrainte; dans le troisième exemple, on est en dynamique et on cherche à maximiser la première fréquence propre; on peut aussi essayer d'écarter deux fréquences de résonance. Bien d'autres choix peuvent être effectués; on peut prendre des fonctions multi-objectifs ou faire de la pénalisation de la fonction objectif par des limitations comme indiqué dans [MAR 95].

4. Les résultats obtenus

Avant chaque exécution, l'algorithme génétique demande à l'utilisateur de préciser les valeurs des paramètres suivants :

- le nombre d'individus d'une population,
- le nombre maximal de générations,
- la longueur des chromosomes,
- la probabilité de croisement,
- la probabilité de mutation.

Il est clair que l'algorithme donne des résultats meilleurs quand les valeurs choisies pour les deux premiers paramètres sont grandes (ceci dans les limites de capacité du matériel informatique utilisé). En pratique, le nombre d'individus d'une population sera de l'ordre de 1 à 5 fois le nombre de digits d'un chromosome.
Cependant, les probabilités de croisement et de mutation sont plus difficiles à choisir. Nous avons vu auparavant que la mutation est un phénomène beaucoup moins fréquent que le croisement ; dans [GOL 89], GOLDBERG conseille les valeurs suivantes :

$$P_{croisement} = 0{,}60$$
$$P_{mutation} = 0{,}03$$

Ces valeurs conseillées sont issues d'une expérimentation numérique sur de nombreux exemples. En tout état de cause, la probabilité de croisement doit être nettement supérieure à la probabilité de mutation car la mutation est moins fréquente [GOL 89]. Si on supprime par exemple toute mutation, l'algorithme converge néanmoins vers un extrèmum mais il y a peu de chances que ce soit l'extrèmum absolu. En théorie, la convergence est obtenue quand toutes les valeurs des coûts d'une population se stabilisent autour d'une valeur maximale. En pratique, la convergence est assez lente avec des hauts et des bas dûs à la nature même de l'algorithme. L'utilisateur se contente d'arrêter le processus quand la valeur maximale du coût d'une population n'évolue plus ; il sélectionne alors manuellement le ou les individus les plus intéressants de la population finale afin d'en comparer les mérites.

4.1 Test 1: test de qualification de la stratégie mise en oeuvre

Ce premier test très simple en statique pour la pièce axisymétrique (d'axe de symétrie z) représentée sur la figure 2 est destiné à vérifier et à fiabiliser la mise en oeuvre des techniques utilisées. Pour ce test on a calculé la raideur une fois pour toutes mais on n'a pas fait de pénalisation des conditions aux limites; d'ailleurs, les calculs étant suffisamment rapides pour les tests présentés ici, cette procédure n'a jamais été appliquée.

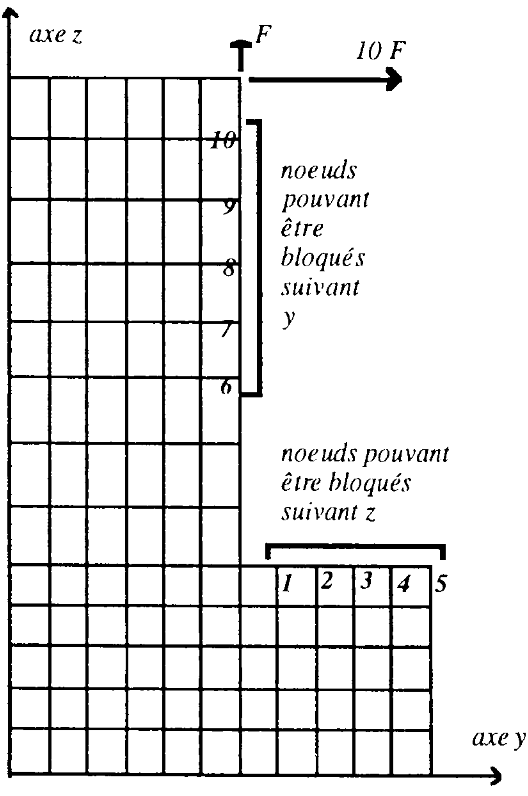

Figure 2. *Maillage, efforts appliqués et conditions limites pour le test 1*

Pour ce test, le chromosome est une chaîne de 10 digits binaires, les 5 premiers digits sont les codes des conditions aux limites des 5 noeuds pouvant être bloqués suivant z, et les 5 digits suivants sont les codes des conditions aux limites des 5 noeuds pouvant être bloqués suivant y; ainsi, le chromosome 1011001000 correspond aux conditions aux limites appliquées sur les noeuds 1, 3, 4 et 7. Notons qu'il y a 2^{10} possibilités. L'objectif est de minimiser le déplacement d du noeud en lequel on applique les forces. Comme l'algorithme génétique cherche en fait le maximum d'une fonction objectif, l'objectif choisi est de maximiser la fonction 1-d. L'intérêt de ce test est que la solution optimale est connue: il s'agit évidemment du chromosome 1111111111, mais ce test permet de valider le processus et surtout de voir en combien d'étapes de calcul l'algorithme génétique parvient à cette solution. La réponse est donnée sur la figure 3 qui donne le maximum de la fonction coût de chaque population (40 individus par population) ainsi que le chromosome correspondant en fonction du nombre de générations. En général, le nombre d'individus par population est pris de l'ordre de 1 à 5 fois la taille du chromosome (ici le nombre de digits). On voit que le maximum est atteint en 5 générations seulement (pour les probabilités de croisement et de mutation conseillées dans la partie 4), ce qui correspond à 200 semi-analyses ou un peu moins (car une solution qui apparaît plusieurs fois au cours du processus n'est calculée qu'une fois pour toutes), ce qui est peu comparé aux 2^{10} combinaisons possibles. La convergence non uniforme est caractéristique des algorithmes génétiques car le meilleur individu de chaque population a une faible probabilité d'être éliminé; d'ailleurs si on force l'algorithme à ne garder que les meilleurs, on enlève à la méthode son caractère probabiliste et l'algorithme peut être plus efficace ou diverger dans certains cas. De plus, si on relance l'optimisation avec les mêmes paramètres, la convergence obtenue n'est pas du tout la même, car le processus est totalement aléatoire.

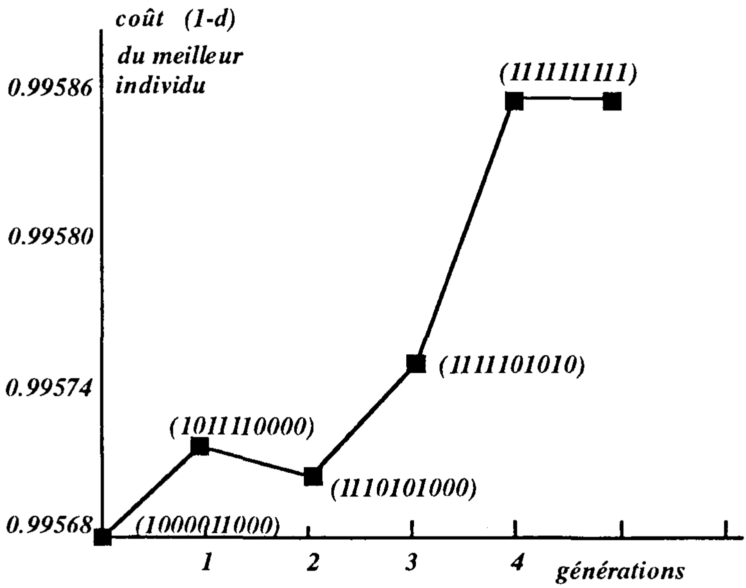

Figure 3. *Coût maximum pour chaque population au cours des générations*

4.2 Test 2: optimisation des conditions aux frontières (tournage)

Ce test montre une première application industrielle dans le cadre de la prise de pièce en usinage, mais toujours avec des données simples permettant de valider et de vérifier la stratégie mise en oeuvre. La qualité de la pièce à usiner dépend notamment des déformations engendrées lors de l'usinage, que ce soit à cause du phénomène d'usinage lui-même ou à cause du maintien en position de la pièce sur son montage. La pièce choisie est une pièce de révolution d'axe z représentée sur la figure 4. Il y a 3 surfaces à usiner notées S1, S2 et S3. Pour ce test, les calculs ne sont effectués que pour la surface S2.

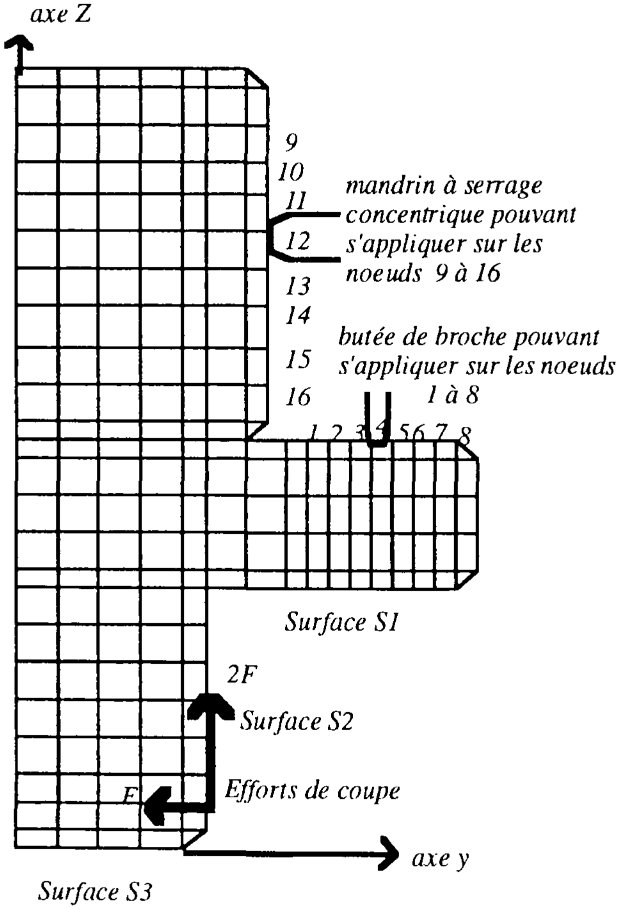

Figure 4. *Maillage, efforts appliqués et conditions limites pour le test 2*

Contrairement au test précédent où les dix noeuds sélectionnés pouvaient être bloqués, ici le mandrin de serrage ne peut s'appliquer que sur l'un des 8 noeuds possibles (noeuds 9 à 16); la butée de broche peut s'appliquer sur l'un des 8 noeuds numérotés 1 à 8. Ce test reste simple et permet de contrôler ainsi le comportement de l'algorithme génétique car la solution optimale est prévisible et le nombre de solutions possibles est limité; il n'en serait pas de même pour un maillage plus fin. On peut reprendre le même type de codage que dans le test précédent; autrement dit les 8 premiers digits du chromosome concerneront les noeuds 1 à 8, mais dans l'algorithme le nombre de 1 possible dans cette partie du chromosome sera limité à 1; de même, les 8 digits suivants concerneront les noeuds 9 à 16 mais tout chromosome ayant un nombre de 1 supérieur à un dans cette partie sera écarté du processus. Par exemple 1000000001000000 est un chromosome admissible. Ce type de codage n'a pas été retenu pour cet exemple car il conduit à des chromosomes de longueur 16 digits et suppose une modification de l'algorithme génétique. Un autre type de codage possible est de construire un chromosome de longueur 2 en décimal; par exemple 29 signifie que les noeuds 2 et 9 sont soumis à des conditions aux limites; le premier digit varie entre 1 et 8, et le deuxième digit varie entre 9 et 16. Ce type de codage décimal a été testé avec succès dans [MAR 95] pour optimiser la séquence d'empilement des matériaux dans un composite multicouche pour lequel les matériaux étaient choisis dans un catalogue de 10 matériaux. Finalement, le codage qui a été retenu pour cet exemple est un codage utilisant 6 digits en binaire, par exemple 100011. Le décodage se fait de la manière suivante (rappelons que le programme de décodage et de modification du fichier de données éléments finis est à concevoir par l'utilisateur pour chaque nouvel exemple et doit être placé immédiatement avant l'analyse): les 3 premiers digits donnent le code du noeud 1 à 8 contraint suivant la correspondance suivante: 000 (noeud 1), 001 (noeud 2), 010 (noeud 3), 011 (noeud 4), 100 (noeud 5), 101 (noeud 6), 110 (noeud 7), 111 (noeud 8), et les 3 digits suivants donnent le code du noeud 9 à 16 contraint suivant le même type de correspondance; c'est ainsi que pour notre exemple 100011 correspond aux noeuds 5 et 12 contraints. Bien sûr, cet exemple est encore un test simple car il n'y a que 64 combinaisons possibles que l'on peut toutes calculer pour avoir l'optimum du problème. L'objectif ici est de minimiser la déformation équivalente maximale ou bien la contrainte équivalente de Von Misès qui se produit au point d'application des efforts. La meilleure solution trouvée par l'algorithme génétique est la combinaison des noeuds 8 et 16 pour laquelle la contrainte de Misès vaut 17.009 daN/mm^2. Ce résultat est trouvé au prix d'une douzaine de calculs par éléments finis (et ceci dès la deuxième génération pour une population de 6 individus). A l'opposé, on peut faire déterminer à l'algorithme génétique la moins bonne solution en relançant le programme avec comme objectif cette fois-ci la maximisation de la contrainte maximale de Von Misès; il s'avère alors que la moins bonne solution est la combinaison des noeuds 1 et 9 pour laquelle la contrainte de Misès vaut 17.195 daN/mm^2. Ce test reste néanmoins simple car le maillage est limité. On pourrait le compliquer avec un maillage plus fin et en faisant déterminer à l'algorithme génétique une solution de compromis valable pour l'usinage des surfaces S1, S2 et S3.

4.3 Test 3: optimisation des conditions aux limites en vibrations

Ce test reprend le test en dynamique proposé dans [SON 93] et permet une fois de plus sur un cas simple de valider la stratégie mise en oeuvre. On considère une plaque carrée de côté 30.5 cm et d'épaisseur 0.328 cm en vibrations de flexion (module d'Young 73.1 GPa, masse volumique 2821 kg/m^3). Cette plaque est en appui sur 4 points situés symétriquement sur les diagonales (figure 5). L'objectif est de trouver la position optimale des appuis avec l'objectif de maximiser la première fréquence fondamentale. Dans [SON 93], ce problème est résolu par une méthode classique de gradient à partir d'un calcul des sensibilités des fréquences par rapport à la position des conditions aux limites. Comme il s'agit du premier mode symétrique de flexion, on ne maille ici qu'un quart de la plaque. [SON 93] trouve deux points optimaux équivalents notés A et B sur la figure 5 et correspondant à des fréquences respectives de 169.46 Hz et 169.67 Hz. En fait [SON 93] n'utilise qu'un maillage de 36 éléments pour la totalité de la plaque et une étude avec des maillages plus fins montre que l'optimum se trouve en réalité entre les points A et B. Ce test est souvent utilisé dans la littérature ([PIT 92]); tous les auteurs trouvent que le point optimal est situé entre A et B.

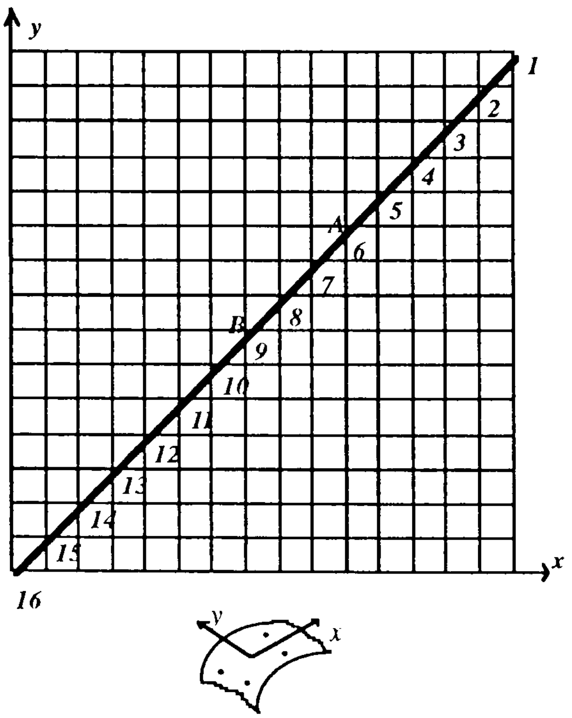

Figure 5. *Maillage du 1/4 de la plaque sur 4 appuis situés sur les diagonales. 1er mode symétrique de flexion*

La mise en oeuvre d'une stratégie à base d'algorithme génétique suppose d'abord le codage sous forme de chromosome de la position du point d'appui sur la diagonale principale du quart de la plaque. Avec le maillage choisi, 15 X 15 éléments, il y a seulement 16 possibilités que l'on peut toutes calculer pour avoir la solution de référence qui correspond effectivement aux points 7 ou 8 avec des fréquences de l'ordre de 205 Hz. Le codage des 16 points possibles est simplement un codage binaire, c'est ainsi que 0000 correspond au noeud 1, 0001 au noeud 2, 0010 au noeud et ainsi de suite jusqu'au noeud 16 (1111). Rappelons que l'objectif est la maximisation de la première fréquence. Pour une population de 4 individus, on obtient au bout de la 3ème génération une population contenant deux individus 0110 (noeud 7) et un individu 0111 (noeud 8), ce qui montre la convergence de l'algorithme génétique mais son efficacité est plus probante sur des exemples avec des chromosomes plus longs (comme dans l'exemple 1).

5. Conclusion

Cette étude a démontré l'efficacité des algorithmes génétiques pour répondre au problème de l'optimisation des conditions aux frontières en éléments finis. Cette étude est avant tout une étude de faisabilité et sera complétée dans un avenir proche par des exemples de type industriel. Elle peut facilement être étendue à d'autres domaines que la mécanique; par exemple en thermique il serait facile de concevoir des chromosomes contenant non seulement l'information sur le type de condition limite, mais aussi l'information sur la valeur à optimiser de cette condition (valeur du flux, valeur du coefficient de convection); des applications peuvent être aussi envisagées en mécanique des fluides. L'efficacité peut être encore améliorée dans le cas de calculs importants (optimisation de forme par exemple), en utilisant des réseaux neuronaux [JOD 94] pour l'analyse du problème à la place d'une analyse par éléments finis classique. En effet, l'utilisation de réseaux de neurones pour la modélisation de structures mécaniques semble donner de bons résultats [BER 92] et [SZE 93]. On peut fort bien imaginer de faire l'apprentissage d'un réseau neuromimétique en parallèle des premières générations qui seraient calculées par éléments finis (c'est-à-dire en utilisant les résultats des analyses par éléments finis). Une fois l'apprentissage terminé, le réseau neuronal remplacerait complètement les calculs par éléments finis, d'où des calculs beaucoup plus rapides, d'autant plus que la méthode des algorithmes génétiques, au contraire des méthodes déterministes, n'exige pas de calculs extrêmement précis de la fonction objectif.

BIBLIOGRAPHIE

[BER 92] BERKE L., HAJELA P. "Applications of artificial neural nets in structural mechanics". *Structural Optimization, Vol 4,* 1992, p. 90-98.

[GOL 89] GOLDBERG D. E. *Genetic Algorithm in Search, Optimization and Machine Learning* Addison - Wesley, 1989.

[HOL 92] HOLLAND J. " Les algorithmes génétiques " *Pour la science,* N° 179 Septembre 1992.

[JOD 94] JODOUIN J.F. *Les réseaux neuromimétiques* Hermès, 1994.

[KEU 92] KEUM D. J., KWAK B. M. "Calculation of Stress Intensity Factors by Sensitivity Analysis with Respect to Change of Boundary Conditions". *Computers and Structures, Vol. 44, n° 1/2,* 1992, p. 63-69.

[MAR 94] MARCELIN J.L., TROMPETTE P. "Optimal location of plate damped parts by use of a genetic algorithm". *Shock and Vibration, Vol. 1, n° 6,* 1994, p. 541-547.

[MAR 94] MARCELIN J.L., TROMPETTE P. "Amortissement optimal de plaques et coques contraintes par algorithmes génétiques", *Colloque international CNES/ESA " structures des véhicules spatiaux" 21-24 juin 1994 -PARIS. Actes publiés par les éditions Cépaduès* , p. 125-134

[MAR 95] MARCELIN J.L. "CAO d'engrenages par algorithmes génétiques", *Revue Internationale de CFAO et d'Infographie,* Hermès, Accepté, à paraître en 1995.

[MAR 95] MARCELIN J.L., TROMPETTE P., DORNBERGER R. "Optimisation of Composite Beam Structures Using a Genetic Algorithm". *Structural Optimization,* Accepté, à paraître en 1995.

[MAR 95] MARCELIN J.L., TROMPETTE P., DORNBERGER R. "Optimal Structural Damping of Skis Using a Genetic Algorithm". *Structural Optimization,* Accepté, à paraître en 1995.

[PIT 92] PITARRESI J.M., KUNZ R.J. "A Simple Technique for the Rapid Estimation of the Optimal Support Locations for a Vibrating Plate". *Journal of Vibration and Acoustics Vol. 114, n° 1,* 1992, p. 112-118.

[SON 93] SON J. H., KWAK B. M. "Optimization of Boundary Conditions for Maximum Fundamental Frequency of Vibrating Structures". *AIAA Journal Vol. 31, n° 12,* 1993, p. 2351-2357.

[SZE 93] SZEWCZYK Z., HAJELA P. "Neural network approximations in a simulated annealing based optimal structural design".*Structural Optimization, Vol 5,* 1993, p. 159-165.

[TRO94] TROMPETTE P., MARCELIN J.L. "The Use of a Genetic Algorithm to Optimize the Damping of Viscoelastic Constrained Beams and Plates". *NATO WR Nafplio Grèce* 26-27 août 1994.

Article reçu le 5 janvier 1995.
Version révisée le 30 mars 1995.