Conception collaborative

M. Masmoudi

March 13, 2004

1 Introduction

Il me parait difficile de faire l'état de l'art en matière d'optimisation multi-disciplinaire en quelques pages, alors que la littérature sur le sujet n'est pas, en général, d'une grande clarté. De plus, le nom optimisation multi-disciplinaire est pris au pied de la lettre Personnellement, je lui prèfère le nom conception collaborative. En effet, l'outil d'optimisation n'est qu'une facette, que l'on ne peut pas isoler, du processus global de conception. En plus, dans ce processus, il ne s'agir pas de représenter chaque discipline par un petit bout de programme, il s'agit, à mon avis, plutôt d'une interaction entre des équipes ayant des disciplines différentes.

Pour illustrer ces propos, je n'ai pas vu de publications issues du monde académique mettant en avant le rôle fondamental du choix du paramètres. Ce choix qui, en général est ignoré par les différents auteurs, joue un rôle déterminant sur la solution finale. Ce rôle est tellement important qu'il ne peut pas laisser les ingénieurs des différentes disciplines indifférents. A l'exception de quelques publications issues du monde industriel, les publications académiques ignorent presque totalement cet aspect. Cette affirmation serait injuste si je ne cite pas le travail proposé par J.A. Désidéri qui pour déterminer les paramètres significatifs. De même, une technique de réveil de maillage par volume englobant a été proposée par l'auteur.

Le point commun entre les publications est de considérer deux types de paramètres, les paramètres privés et les paramètres public. Le point commun entre elles est de compliquer la présentation en trainant les paramètres privés alors que l'on peut supposer que chaque discipline rend une réponse où les paramètres privés ont été choisi d'une manière optimale et transparente pour les autres disciplines.

Dans ce qui suit, je présente les idées de base de l'état de l'art, mais la forme de la présentation est personnelle.

2 Comparaison des méthodes de conception collaboratives

2.1 La méthode intégrée (FIO: Fully Integrated Optimization)

Pour simplifier cet exposé, on se limite à deux disciplines (et on ignore les paramètres privés). Les équations d'état sont données sous forme explicite

$$a_1 = A_1(x, a_2) \tag{1}$$
$$a_2 = A_2(x, a_1) \tag{2}$$

où x représente les paramètres de conception. On fait abstraction de l'état, on ne s'intéresse qu'à la sortie (output). On note a_i la sortie de la discipline i. Dans la première équation, a_2 représente la contribution de la discipline 2 au calcul de a_1.

On cherche à minimiser le problème d'optimisation

$$\min_x f(x, a_1(x), a_2(x))$$
$$g(x, a_1(x), a_2(x)) \leq 0 \tag{3}$$

Nous ignorons dans cette présentation, les contraintes internes à la discipline: pour nous elle font partie de l'équation d'état.

Dans l'approche FIO, le problème d'optimisation et les équations d'état sont traités simultanément. Aucune autonomie n'est laissée aux disciplines.

2.2 La méthode MDA (MultiDisciplinary Analysis)

A chaque itération de l'algorithme d'optimisation, les équations d'état sont satisfaites. C'est la méthode naturelle. Un algorithme d'optimisation centralisé qui demande des services d'analyse (et peut etre de calcul de gradient) à chaque discipline.

Elle me parait très saine. A mon avis le seul reproche que l'on peut faire à cette méthode est de ne pas permettre de faire beaucoup de publications.

2.3 La méthode AAO (All-At-Once)

Il s'agit de la méthode qui s'appelle également "one shot", SAND ou SAD (Simultaneous Analysis and Design) [8, 6, 12]. On résout simultanément les équations d'état et le problème d'optimisation. Les équations d'état ne sont pas forcément satisfaites à chaque itération de l'algorithme d'optimisation. Elles le sont à la convergence de l'algorithme d'optimisation. Les équations d'état sont considérées comme des contraintes d'égalité.

Il s'agit d'une méthode efficace. Sa convergence dans le cas où l'une des équations d'état est fortement non linéaire peut poser quelques problèmes.

Chaque discipline doit offrir des garanties quant à la qualité des outils mis à la disposition des autres partenaires. Chaque discipline a ses propres réglages pour converger les codes qui lui sont propres. Dans un contexte AAO, il n'y a plus rien qui garantit la convergence de chaque solveur et encore moins la convergence du processus global.

2.4 La méthode DAO (Disciplinary Analysis Optimization)

Ici, on ne confie pas aux disciplines la moindre initiative d'optimisation. Cependant, chaque discipline agit sur les paramètres pour obtenir un point admissible. Nous avions dit que les

contraintes privés sont cachées dans les équations d'état. Ces contraintes sont généralement satisfaites en ajustant les variables privés. Mais rien ne garatit qu'il existe une solution pour le vecteur x représentant les variables publiques proposé. Pour cette raison, on propose de relaxer x. Chaque discipline a la liberté de choisir le z_i qui permet de satisfaire les contraintes privés. A la convergence, les trois z_i coincident grâce aux trois contraintes ajoutées :

$$\begin{cases} \min_x f_{DAO}(x, b_1, b_2) \\ g(z_0, b_1, b_2) \leq 0 \\ b_1 = a_1(z_1, b_2), \\ b_2 = a_2(z_2, b_1), \\ z_0 = x \\ z_1 = x \\ z_2 = x \end{cases}$$

2.5 La méthode CO (Collaborative Optimization)

Ici, on demande aux disciplines de participer à l'amélioration des performances.Pour cela, il suffit d'écrire le problème Nous reprenons les notations du paragraphe DAO.

Le problème d'optimisation de la discipline 1 est

$$\min_{z_1, b_1} \frac{1}{2}[\|z_1 - x\|^2 + \|b_1 - a_1(z_1, b_2)\|^2]$$

où a_1 est la solution de l'équation d'état (1).

Le problème d'optimisation de la discipline 2 est

$$\min_{z_2, b_2} \frac{1}{2}[\|z_2 - x\|^2 + \|b_2 - a_2(z_2, b_1)\|^2]$$

où a_1 est la solution de l'équation d'état (2).

Le problème d'optimisation de niveau global ou niveau système est

$$\begin{cases} \min_{z_0, \xi_1, \xi_1} \frac{1}{2}[\|z_0 - x\|^2 + \|\xi_1 - b_1\|^2 + \|\xi_2 - b_2\|^2] \\ g(z_0, \xi_1, \xi_2) \leq 0 \end{cases}$$

On peut dire que le but de ce problème d'optimisation est de donner des objectifs à atteindre au sens des moindres carrés aux problèmes disciplinaires. La norme est à prendre au sens d'une semi-norme.

2.6 La méthode CSSO (Collaborative SubSpace Optimization)

Nous renvoyons à [28, ?, 48, ?], pour une présentation détaillée de ces méthodes. Les paramètres de conception sont partagés entre disciplines. On alloue à chaque discipline quelques paramètres à optimiser.

C'est un peu une adaption de la méthode des sous-domaines à ces problèmes d'optimisation avec une différence de taille: dans la résolution d'équations aux dérivées partielles, une inconnue peut être facilement attachée à une équation. C'est-à-dire qu'un domaine contrôle un certain nombre de variables ainsi que les équations qui leurs sont attachées.

En optimisation multidisciplinaire, il n'est pas évident d'associer un paramètre public à une discipline. Par définition, il est partagé entre les disciplines.

Nous continuons à utiliser les mêmes notations qu'au paragraphes précédents. Dans ce paragraphe, dans la discipline 1, on fait appel à une approcimation de la discipline 2 et dans la discipline 2, on fait appel à une approximation de la discipline 1.

On rappelle le problème initial

$$a_1 = A_1(x, a_2) \tag{4}$$

$$a_2 = A_2(x, a_1) \tag{5}$$

$$\min_x f(x, a_1(x), a_2(x))$$
$$g(x, a_1(x), a_2(x)) \leq 0 \tag{6}$$

et on pose $x = (x_1, x_2)$ où x_i est le tableau de variables affectés à la discipline i. Ensuite, on pose $\widetilde{a}_i$ l'approximation de l'analyse et $a_{i.}$ Nous reviendrons plus tard sur cet aspect; D'une manière naturelle, on peut écrire la tâche de la discipline 1

$$a_1 = A_1(x, \widetilde{a}_2) \tag{7}$$
$$\widetilde{a}_2 = \widetilde{A}_2(x, a_1) \tag{8}$$

$$\min_{x_1} f(x_1, x_2, a_1(x), \widetilde{a}_2(x))$$
$$g(x, a_1(x), \widetilde{a}_2(x)) \leq 0 \tag{9}$$
$$(1 - \Delta)x_{1cour} \leq x_1 \leq (1 - \Delta)x_{1cour} \tag{10}$$

Cette dernière contrainte est liée au domaine de validité de l'approximation de a_2. Ainsi x_{1cour} est le point où cette approximation a été faite et Δ donne la région de confiance de cette approximation.

De même la tâche associée à la discipline 2 peut être écrite

$$\widetilde{a}_1 = \widetilde{A}_1(x, a_2) \tag{11}$$
$$a_2 = A_2(x, \widetilde{a}_1) \tag{12}$$

$$\min_{x_2} f(x_1, x_2, \widetilde{a}_1(x), a_2(x))$$
$$g(x, \widetilde{a}_1(x), a_2(x)) \leq 0 \tag{13}$$
$$(1 - \Delta)x_{2cour} \leq x_2 \leq (1 - \Delta)x_{2cour} \tag{14}$$

Ici toutes les recettes classiques utilisées en décomposition de domaines sont applicables. On peut associer à chaque discipline un processus. Le processus i communique aux autres processus, l'état de la variable courante x_{icour}, l'approximation $\widetilde{a}_i$ de a_i et son domaine de validité . On trouve dans la littérature de nombreuses méthodes de paramétrisation qui ont pour but d'élargir le domaine de validité de l'approximation et d'augmenter l'autonomie des disciplines.

En décomposition de domaines, il existe une manière naturelle d'associer une inconnue à une équation et il y a autant d'équations que de données. Cette association rend la mise en oeuvre plus simple. En conception collaborative, on associe un sous-espace à une discipline d'une manière plus ou moins arbitraire, il faut donc communiquer au processus voisin l'état des variables locales ainsi que le modèle approché local.

2.6.1 Approximation avec des dérivées d'ordre 1

La paramétrisation la plus simple consiste à utiliser le gradient, mais son domaine de définition est petit. Ainsi nous avons

$$\widetilde{a}_1 = \partial_x A_1(x_{cour}, a_2(x_{cour}))(x - x_{cour}) + \partial_{a_2} A_1(x_{cour}, a_2(x_{cour}))(a_2(x) - a_2(x_{cour}))$$

et

$$\widetilde{a}_2 = \partial_x A_2(x_{cour}, a_1(x_{cour}))(x - x_{cour}) + \partial_{a_2} A_2(x_{cour}, a_1(x_{cour}))(a_1(x) - a_1(x_{cour})).$$

Les méthodes adjointes (dérivation en mode inverse) sont particulièrement adaptées à ce contexte: on peut avoir un grand nombre de paramètres, les tableaux x_i et a_i peuvent avoir une grande dimension..

2.6.2 Approximation par les méthodes de paramétrisation

Ces méthodes sont particulièrement intéressantes lorsque les disciplines sont faiblement couplées. Plus exactement, on se met dans le cas où l'une des disciplines ne dépend pas des autres:

$$a_1 = A_1(x) \tag{15}$$
$$a_2 = A_2(x, a_1). \tag{16}$$

La discipline 1 ne dépend pas de la discipline 2.

On peut citer le cas très important de l'aéroélasticité. Dans ce cas a_1 représente la réponse de la structure pour une sollicitation dynamique (en utilisant la réponse modale). Une bonne approximation consiste à utiliser la même base modale lorsque les paramètres de conception x varient. Nous obtenons ainsi $\widetilde{a}_1$.

3 La théorie des jeux

La théorie des jeux a été considérée dans le cadre de l'optimisation multicritère [20, 22, 21, 33] Elle a été évoquée par J. A. Désidéri comme un moyen possible pour traiter les problèmes d'optimisation multidisciplinaires. Elle me semble mieux appropriée à la conception multidisciplinaire que les méthodes présentées au paragraphe 2. En particulier, les critères de deux disciplines différentes sont de nature différentes: le max de la contrainte de von Mises, la trainée, le déplacement maximal, la consommation de carburant sont des critères de nature différentes. Dans le paragraphe 2, on se contente de pondérer ces criètres. Toute opération de pondération a un caractère arbitraire. L'avantage de la théorie des jeux est de travailler dans un contexte multi-critères, chaque discipline conserve son propre critère.

Son rôle pourrait être important dans un contexte multidisciplinaire. En fait, chaque partenaire peut être considéré comme un joueur.

On considère deux fonctions coûts de $A \times B$ à valeur dans $\Re$:

$$\begin{array}{rrcl} f_A: & A \times B & \rightarrow & \Re \\ & (x, y) & \mapsto & f_A(x, y) \end{array}$$

et

$$\begin{array}{rrcl} f_B: & A \times B & \rightarrow & \Re \\ & (x, y) & \mapsto & f_B(x, y). \end{array}$$

où $A \times B$ sont deux parties respectives de $\Re^n$ et $\Re^m$. Le joueur A maximise f_A en agissant sur x. Le joueur B minimise f_B en agissant sur y.

3.1 Equilibre de Pareto

3.2 Définition

On dit que (x^*, y^*) est optimal au sens de Pareto s'il n'existe aucun couple $(x, y) \in A \times B$, tel que

$$\begin{cases} f_A(x,y) < f_A(x^*,y^*) \\ f_B(x,y) < f_B(x^*,y^*). \end{cases}$$

Cela veut dire qu'il est impossible d'améliorer f_A et f_B en même temps. Si l'on amétliore f_A alors f_B se dégrade et réciproquement.

L'ensemble des points où l'équilibre de Pareto est satisfait, s'appelle front de Pareto.

Lorsque le front de Pareto est convexe, on peut trouver tous ses points en minimisant $f_\lambda = \lambda f_A + (1-\lambda) f_B$ pour tout $0 < \lambda < 1$. Pour tout λ, on trouve un point d'équilibre au sens de Pareto.

Le front de Pareto donne l'ensemble des solutions qui ne sont pas dominées (on ne peut pas améliorer un critère sans dégrader l'autre). L'ensemble de ces solutions peut fournir une base de discussion entre les responsables des deux disciplines.

3.3 Exemple

$$\begin{cases} f_A = (x-1)^2 + (x-y)^2 \\ f_B = (x-3)^2 + (x-y)^2. \end{cases}$$

Les points d'équilibre de Pareto, sont obtenus, en combinant les deux critères

$$f_\lambda = \lambda f_A + (1-\lambda) f_B$$

et en minimiasant pour tout λ. Dans notre cas

$$f_\lambda = \lambda(x-1)^2 + (1-\lambda)(y-3)^2 + (x-y)^2$$

Pour tout λ, nous avons

$$\begin{aligned} \partial_x f_\lambda &= 2[\lambda(x-1) + (x-y)] = 0 \\ \partial_y f_\lambda &= 2[(1-\lambda)(y-3) - (x-y)] = 0. \end{aligned}$$

Ce qui donne

$$\begin{cases} (1+\lambda)x - y = \lambda \\ -x + (2-\lambda)y = 3(1-\lambda). \end{cases} \tag{17}$$

3.4 Equilibre de Nash

3.5 Définition

On dit que le couple $(x^*, y^*) \in A \times B$ est un point d'équilibre de Nash, si et seulement si

$$\begin{cases} f_A(x^*,y^*) = \inf_x f_A(x,y^*) \\ f_B(x^*,y^*) = \inf_y f_B(x^*,y) \end{cases}$$

Il est clair, dans ce contexte, que cet équilibre doit satisfaire les conditions d'optimalité

$$\begin{cases} \nabla_x f_A(x^*,y^*) = 0 \\ \nabla_y f_B(x^*,y^*) = 0. \end{cases}$$

3.6 Exemple

On considère le problème

$$\begin{cases} f_A = (x-1)^2 + (x-y)^2 \\ f_B = (y-3)^2 + (x-y)^2. \end{cases}$$

En applicant la condition d'optimalité, on obtient,

$$\begin{array}{c} \partial_x f_A = 0 \\ \partial_y f_B = 0 \end{array} \iff \begin{array}{c} y = 2x - 1 \\ x = 2y - 3 \end{array} \iff \begin{array}{c} x = 5/3 \\ y = 7/3. \end{array}$$

L'équlibre de Nash est le meilleur moyen pour déterminer une solution optimale sans chercher de compromis. Cependant, en optimisation multi-disciplinaire, le véritable compromis est caché dans le choix des paramètres de conception en général. Dans ce cas particulier, le choix du sous-espace attribué à la discipline 1 et de son complémentaire, attribué à la discipline 2, représente un compromis caché.

Remarquons que la solution obtenue ici est exactement celle que l'on obtient avec $\lambda = 1$ dans la première équation de 17 et $\lambda = 0$ dans la deuxième équation de 17.

3.7 Equilibre de Stackelberg

3.8 Définition

Dans ce cas, l'un des deux joueurs mène le jeux. Disons que A mène le jeu. Le problème de minimisation devient

$$\min_x \; f_A(x, y_x)$$

$$\text{avec}$$

$$y_x = \arg\min_y \; f_B(x, y).$$

Cela rappelle parfaitement les paramètres privés. Il s'agit d'une technique d'élimination de variable. D'une certaine manière, il s'agit de minimiser $j(x) := f_A(x, y_x)$.

Si l'on suppose que f_A est un critère aérodynamique, tel que la traînée et que f_B désigne la masse d'un avion. Le vecteur y représente les paramètres privés de la mécanique de structures (nombre de longerons, position, épaisseur, ...). A chaque évaluation de f_B, on résout un problème d'optimisation en mécanique de structures.

3.9 exemple

Dans l'exemple donné au paragraphe précédent,

$$\begin{cases} f_A = (x-1)^2 + (x-y)^2 \\ f_B = (y-3)^2 + (x-y)^2. \end{cases}$$

L'équilibre de Stackelberg donne le même résultat que l'équilibre de Nash

$$y_x = \frac{(x+3)}{2},$$

$$j(x) = (x-1)^2 + (\frac{x-3}{2})^2.$$

D'où $x = 5/3$.

3.10 Application de la théorie des jeux à la conception multi-disciplinaire

Il suffit de considérer le problème 12 et 3

$$a_1 = A_1(x, a_2) \tag{18}$$
$$a_2 = A_2(x, a_1) \tag{19}$$

$$\min_x f(x, a_1(x), a_2(x))$$
$$g(x, a_1(x), a_2(x)) \leq 0 \tag{20}$$

où l'on remplace le critère pondéré $f(x, a_1(x), a_2(x))$ par $f_1(x, a_1(x), a_2(x))$ pour la discipline 1 et $f_2(x, a_1(x), a_2(x))$ pour la discipline 2. Par exemple, si la discipline 1 est la mécanique des structures, f_1 représente le poids de la structure. Si la discipline 2 est l'aérodynamique alors f_2 es la portance (avec une contrainte sur la trainée).

Ensuite, on considère la décomposition effectuée dans le paragraphe 2.6 où l'on remplace la fonction coût commune aux deux disciplines par la fonction coût de la discipline. La partie jouée par la discipline 1 est

$$a_1 = A_1(x, \widetilde{a}_2) \tag{21}$$
$$\widetilde{a}_2 = \widetilde{A}_2(x, a_1) \tag{22}$$

$$\min_{x_1} f_1(x_1, x_2, a_1(x), \widetilde{a}_2(x))$$
$$g(x, a_1(x), \widetilde{a}_2(x)) \leq 0 \tag{23}$$
$$(1-\Delta)x_{1cour} \leq x_1 \leq (1-\Delta)x_{1cour}. \tag{24}$$

La partie jouée par la discipline 2 est

$$\widetilde{a}_1 = \widetilde{A}_1(x, a_2) \tag{25}$$
$$a_2 = A_2(x, \widetilde{a}_1) \tag{26}$$

$$\min_{x_2} f_2(x_1, x_2, \widetilde{a}_1(x), a_2(x))$$
$$g(x, \widetilde{a}_1(x), a_2(x)) \leq 0 \tag{27}$$
$$(1-\Delta)x_{2cour} \leq x_2 \leq (1-\Delta)x_{2cour} \tag{28}$$

Les contraintes peuvent être intégrées dans la fonction coût pour se ramener exactement au contexte de la théorie des jeux.

Le front de Pareto apporte offre une réponse fondamentale aux problèmes d'optimisation multi-disciplinaires. Parmi toutes les solutions offertes par le front de Pareto, les concepteurs peuvent choisir la solution la plus pertinente en tenant compte de critères "non écrits", tels que l'esthétique, la fabricabilité, ...

L'équilibre de Nash permet d'éviter le douleureux problèmes des compromis.

L'équilibre de Stackelberg correspond exactement à l'optimisation de variables (privées) internes à la discipline.

La théorie des jeux est très peu appliquée dans le contexte de l'optimisation multidisciplinaire. A différent niveaux, elle peut offrir des réponses intéressantes:

- l'équilibre de Pareto permet d'avoir une gamme de solutions pertinentes. Parmi ces solutions, on peut choisir celles qui répondent à des critères non écrits,

- l'équilibre de Nash, permet d'éviter la notion de compromis, même si le véritable compromis réside dans le choix des sous-espaces de la discipline 1 et de la discipline 2.

- l'équilibre de Stackelberg est une approche particulièrement adaptée à l'optimisation par rapport à des variables privées.

References

[1] N. Alexandrov. Optimization of engineering systems governed by differential equations: Editor's introduction. In *SIAM Activity Group on Optimization*.

[2] N. Alexandrov. Comparative properties of collaborative optimization and other approaches to mdo. Technical report, ICASE, 1999.

[3] N. Alexandrov. Comparative properties of collaborative optimization. Technical report, NASA, 2000.

[4] N. Alexandrov and S. Kodiyalam. Initial results of an mdo method evaluation study. Technical report, AIAA, 1998.

[5] N. Alexandrov and R. Lewis. Analytical and computational properties of distributed approaches to mdo. Technical report, AIAA, 2000.

[6] N. M. Alexandrov and R. M. Lewis. Algorithmic perspectives on problem formulations in mdo. Technical report, 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary analysis and optimization, 2000.

[7] A. Aussem. *Théorie et application des réseaux de neurones récurrents et dynamiques à la prédiction, à la modélisation et au contrôle aptatif des processus dynamiques*. PhD thesis, Université René Descartes - Paris V, 1995.

[8] R. Braun, P. Gage, I. Kroo, and I. Sobieski. Implementation and performance issues in collaborative optimization. Technical report, AIAA, 1996.

[9] M. Cormery. *De l'optimisation aérodynamique vers l'optimisation multidisciplinaire dans un contexte industriel*. PhD thesis, Université Paul Sabatier, 2000.

[10] David and B. Serafini. *A framework for managing models in nonlinear optimization of computationally expensive functions*. PhD thesis, Rice University, 1998.

[11] S. Grihon and M. Cormery. Multidisciplinary optimisation applied to a 500 seat aircraft. In *Proceedings 4th ECCOMAS Conference*.

[12] C. Gumbert, J. Hou, and P. Newman. Simultaneous aerodunamic and structural design optimization (sasdo) for a 2-d wing. Technical report, AIAA, 2001.

[13] I. Kroo and V. Manning. Collaborative optimisation status and directions. Technical report, AIAA, 2000.

[14] K. KUBOTA. A preprocessor for fast automatic differentiation - applications and difficulties on practical problems.

[15] S. Lawrence, L. G. C., and A. C. Tsoi. Lessons in neural network training : overfitting may be harder than expected. In *Fourteenth Conferenceon Artificial Intelligence*.

[16] S. Lawrence, A. C. Tsoi, and D. B. Andrew. Function approximation with neural networks and local methods : bias, variance and smoothness. In *Australian Conference on Neural Networks*.

[17] J. Markish and K. Willcox. Multidisciplinary techniques for commercial aircraft system design. Technical report, 2002.

[18] D. W. Monell and W. M. Piland. Aerospace systems design in nasa's collaborative engineering development. In *50th Intrenational Astronautical Congress*.

[19] S. Padula, J. Korte, H. Dunn, and A. Salas. Multidisciplinary optimiszation branch experience using isight software. Technical report, NASA, 1999.

[20] J. Périaux. *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science: Recent Advances and Industrial Applications*. John Wiley & Son Ltd, 1998.

[21] J. P. R. MÄKINEN, P. NEITTAANMÄKI and J. TOIVANEN. A genetic algorithm for multiobjective design optimization in aerodynamics and electromagnetics. In *Computational Fluid Dynamics '98, Proceedings of the ECCOMAS 98 Conference*, volume 2, pages 418–422, Athens, Greece, 1998. K. D. Papailiou et al. (eds), Wiley.

[22] R. G. RAMOS, A. M. and J. PERIAUX. Nash equilibria for the multiobjective control of linear partial differential equations. *Journal of Optimization Theory and Applications*, 112(3):457–498, 2002.

[23] J. F. Rodriguez, J. E. Renaud, B. A. Wujek, and R. V. Trappeta. Trust region model management in multidisciplinary design optimization. *Journal of Computationnal and Applied Mathematics*, 2000.

[24] A. Salas and J. Rogers. A web-based system for monitoring and controlling multidisciplinary design. Technical report, NASA, 1997.

[25] A. Salas and J. Townsend. Framework requirement for mdo application development. Technical report, AIAA, 1998.

[26] D. Sandy, Balki, and K. J. L. Denis. A neural network approach to response surface mathodology. *Communication in Statistics - Theory and Methods*, 2000.

[27] D. Sandy, Balkin, and K. L. Dennis. A neural approach to response surface methodology. *Communications in Statistics - Theory and Methods*, 2000.

[28] J. Shankar, C. Ribbens, R. Haftka, and W. L. Computational study of nonhierarchical decomposition algorithm. *Compuational Optimization and applications*, 1993.

[29] I. P. Sobieski and I. M. Kroo. Collaborative optimization using response surface estimation. *AIAA Journal*, 2000.

[30] V. STRASSEN. Algebraic complexity theory. In A. Elsevier, editor, *in J. van Leeuwen (Editeur), Handbook of Theoritical Computer Science*, volume A, 1990.

[31] V. Torczon and M. W. Trosset. Using approximations to accelerate engineering design optimization. Technical report, NASA, 1998.

[32] J. Walsh, J. Townsend, A. Salas, J. Samareh, V. Mukhopadhyay, and J. Barthelemy. Multidisciplinary high-fidelety analysis and optimization of aerospace vehicles. Technical report, AIAA, 2000.

[33] J. F. Wang. *Optimisation distribuée multicritère par Algorithmes génétiques et Théorie des jeux. Application à la simulation numérique de problèmes d'hypersustentation en aérodynamique*. PhD thesis, Université Paris VI, 1992.

[34] J. Young, R. Anderson, and R. Yurkovich. A description of the f/a-18e/f design and design process. Technical report, 1998.

[35] A. Aussem, *Théorie et application des réseaux de neurones récurrents et dynamiques à la prédiction, à la modélisation et au contrôle adaptatif des processus dynamiques*, Thèse de l'université René Descartes - Paris V, 1995.

[36] S. D. Balkin, D. K. Lin, *A neural network approach to response surface methodology*, Communications in Statistics - Theory and Methods, vol. 29, 9-10, pp 2215-2227, 2000.

[37] R. H. Byrd, P. Lu, J. Nocedal, C. Zhu, *A Limited Memory Algorithm for Bound Constrained Optimization*, Technical Report NAM-08, 1994.

[38] J. C. Gilbert, G. Le Vey, J. Masse, *La différentiation automatique de fonctions représentées par des programmes*, rapport de recherche INRIA, 1997.

[39] F. Girosi, M. Jones, T. Poggio, *Regularization Theory and Neural Networks Architectures*, in Neural Computation, pp 219-269, 1995.

[40] S. Haykin, *Neural Networks, a comprehensive foundation*, Prentice Hall International, 1999.

[41] K. Hornik, M. Stinchcombe, H. White, *Multilayer Feedforward Networks are Universal Approximators*, Neural Networks, vol. 2, pp 359-366, 1989.

[42] J.-F. Jodouin, *Les Réseaux de Neurones. Principes et définitions*, Hermes, 1994.

[43] C. T. Kelley, *Iterative methods for Optimization*, Frontiers in Applied Mathematics, SIAM, 1999.

[44] S. Lawrence, A. C. Tsoi, A. D. Back, *Function approximation with neural networks and local methods : bias, variance and smoothness*, in Australian Conference on Neural Networks, Peter Barlett, Anthony Burkitt, Robert Williamson, eds., pp 16-21, 1996.

[45] S. Lawrence, C. L. Giles, A. C. Tsoi, *Lessons in neural network training : overfitting may be harder than expected*, in Fourteenth Conference on Artificial Intelligence, AIAA Press, pp 540-545, 1997.

[46] E. Soria, A. J. Serrano, *Redes Neuronales: Una Breve Introduccion*, EMESIS.

[47] R. Tappeta, S. Nagendra, JE. Renaud, K. Badhrinath, *Concurrent Sub-Space Optimization (CSSO) MDO Algorithms in iSIGHT. CSSO in iSIGHT: Validation and Testing*, in Technical Information Series, GE Research and Development Center, 87CRD186, 1998.

[48] R. Tappeta, S. Nagendra, JE. Renaud, K. Badhrinath, *Concurrent Sub-Space Optimization (CSSO) Code Usage in iSIGHT*, in Technical Information Series, GE Research and Development Center, 87CRD188, 1998.

[49] D. Padmanabhan, S. M. Batill, *Reliability based optimization using approximations with applications to multi-disciplinary system design*, AIAA-2002-0449, 2002.