



HAL
open science

Global Optimization of Mechanical Systems

Rodolphe Le Riche

► **To cite this version:**

Rodolphe Le Riche. Global Optimization of Mechanical Systems. Engineering Sciences [physics]. Ecole Nationale Supérieure des Mines de Saint-Etienne, 2008. tel-00476005

HAL Id: tel-00476005

<https://theses.hal.science/tel-00476005v1>

Submitted on 23 Apr 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimisation globale de systèmes mécaniques
Global optimization of mechanical systems

“Habilitation à Diriger les Recherches” manuscript,
defended on the 30th of September 2008 at
ENSM-SE

Rodolphe Le Riche
CNRS UMR 5146 and ENSM-SE
`leriche@emse.fr`

Contents

Résumé étendu en Français	iii
Foreword	xi
Optimizing mechanical systems	xiii
0.1 The sides of an optimization problem	xiii
0.1.1 Optimization basic components	xiii
0.1.2 Engineering optimization: connecting the basic components	xiv
0.2 Bottlenecks when optimizing mechanical systems	xvi
0.3 General versus specific solutions : the No Free Lunch theorem	xviii
0.4 Scope of this dissertation	xix
0.4.1 Our research process	xix
0.4.2 Structure of the dissertation	xix
0.4.3 Scientific contributions	xx
1 General global optimization algorithms	1
1.1 Evolutionary optimization	1
1.1.1 Context	1
1.1.2 Overall structure of an evolutionary algorithm	2
1.1.3 Tuning an EA : the exploration-exploitation compromise	10
1.1.4 An historical standpoint	10
1.1.5 Tools for designing EAs	14
1.1.6 Parametric evolutionary optimizers: EDAs and ES	16
1.2 Handling non-linear inequality constraints	21
1.2.1 Context	21
1.2.2 The segregated genetic algorithm	22
1.2.3 Non-differentiable penalty functions and duality	24
1.3 GBNM	31
1.3.1 A brief presentation of GBNM	31
1.3.2 Application to the design of swimming monofins	32
1.4 A multi-point global optimization criterion	37
1.4.1 Space filling design of experiment for optimization	37
1.4.2 Ordinary kriging	38
1.4.3 Kriging-based optimization criteria	38
1.4.4 The multi-points expected improvement	41
2 Specialized evolutionary optimizers	45
2.1 How to specialize EAs	45
2.2 Evolutionary algorithms for composite structures	46
2.2.1 Specializing the coding and the variation operators	46
2.2.2 The Double Distribution Optimization Algorithm	51
2.3 Evolutionary algorithms for shape optimization	54

3	Perspectives	59
3.1	Optimum design accounting for uncertainties	60
3.2	Identification accounting for uncertainties in solid mechanics	62
3.2.1	Identification from full-field measurements	62
3.2.2	Bayesian identification	66
3.3	Finite-cost optimization	68
3.3.1	A sequential finite-cost algorithm based on kriging	68
3.3.2	A finite-cost, multi-fidelity optimization problem	69
4	Bibliography	71
4.1	External bibliography	71
4.2	Personal bibliography	85
4.2.1	Patents	86
4.2.2	Journals	86
4.2.3	Conferences	89
4.2.4	Technical and contract reports	92
4.2.5	Popularizing texts	94
4.2.6	Seminars	94
5	Other professional activities	97
5.1	Project management	97
5.1.1	The OMD project	97
5.1.2	Other projects in collaboration with private companies	98
5.2	Teaching	100
5.2.1	Classes taught	100
5.2.2	Students advising	103
5.3	Reviewing and other administrative tasks	105
5.3.1	Reviewing	105
5.3.2	Conferences and seminars organization	106
5.3.3	Other administrative tasks	106
5.4	Curriculum Vitae	107
	Acknowledgements	109

Résumé étendu en Français

Eléments d'optimisation en mécanique

Les problèmes d'optimisation, c'est à dire la minimisation ou la maximisation de fonctions ou de fonctionnelles, entrent à de nombreux niveaux dans les activités de modélisation et de conception de systèmes mécaniques. L'identification d'un modèle fait intervenir des minimisations d'écart entre modèles et expériences. La modélisation peut souvent être formulée comme une minimisation d'énergie (par exemple l'énergie potentielle totale pour les systèmes conservatifs). Enfin, la conception optimale est une recherche de solutions maximisant des critères de performance.

Si l'on porte un regard numérique sur les difficultés de la modélisation et, a fortiori, de la conception en mécanique, on constate que nombre de ces difficultés sont liées à la complexité des problèmes d'optimisation sous-jacents. La construction d'un modèle en présence de données expérimentales incomplètes (ce qui est généralement le cas) est un problème d'identification mal posé. La fragilité des logiciels d'analyse par éléments finis non linéaires est due à la grande taille de l'espace dans lequel on minimise les résidus (le nombre de degrés de liberté) et à la non différentiabilité (e.g., plasticité, endommagement), voire à la non convexité (e.g., flambage) des résidus. Au niveau de la conception optimale, les obstacles rencontrés sont le temps de calcul des modèles, qui croît à la même vitesse que la puissance des ordinateurs, l'absence de calcul des sensibilités des modèles par rapport aux variables, la fragilité numérique des modèles et les difficultés d'interfaçage entre modèles et algorithmes d'optimisation.

Pour saisir la problématique de l'**optimisation de systèmes mécaniques**, il est utile de décomposer un problème en étapes génériques :

Modélisation : Définition du modèle numérique¹ du système considéré, y , où y est un vecteur de réponses. Le modèle peut être établi sur une base physique, par exemple la résolution d'équations aux dérivées partielles par éléments finis, ou à partir d'un jeu d'entrées x - sorties y qui sont numériquement apprises. Un même système est souvent décrit par plusieurs modèles, par exemple un modèle d'apprentissage construit à partir de simulations physiques fines (cf. l'optimisation multi-fidélité en section 3.3.2).

Formulation : Ecriture du problème d'optimisation sous la forme d'un vecteur de critères f à minimiser en changeant les variables x dans un espace de recherche

¹Les cas où le modèle et la résolution du problème d'optimisation peuvent être traités de façon analytique ne sont pas l'objet de nos travaux car ils ne sont pas, en général, des problèmes difficiles.

\mathcal{S} , en satisfaisant des contraintes d'inégalité, g ,

$$\begin{cases} \min_{x \in \mathcal{S}} f(x) \\ \text{tel que } g(x) \leq 0 \end{cases} \quad (1)$$

f et g sont des fonctions implicites des variables x et d'autres variables non optimisées e à travers l'exécution du modèle numérique y , par exemple $f(x) \equiv f(x, e, y(x, e))$.

Stratégie de résolution : Une fois la formulation faite, la nature mathématique du problème d'optimisation de l'Equation (1) est précisée. \mathcal{S} est-il continu, discret ou mixte, quel est le nombre de variables, existe-t-il des contraintes, le problème est-il linéaire, quadratique, convexe, la solution du problème est-elle unique, les critères d'optimisation sont-ils bruités ? Cette analyse permet de définir une stratégie de résolution qui pourra également tenir compte d'autre connaissance a priori (e.g., Fully Stressed Design en optimisation de formes, [75]). Il en résulte un algorithme d'optimisation ou optimiseur.

Mise en œuvre : Mise en place informatique du modèle et de son interface avec l'algorithme d'optimisation. Cette étape est en général très consommatrice de temps humain.

Verrous à l'optimisation en mécanique

Toutes ces composantes de l'optimisation appliquée sont traditionnellement abordées de manière séparée dans les disciplines scientifiques établies (la modélisation par les physiciens des domaines concernés, l'optimisation par les mathématiciens appliqués, la mise en œuvre par les informaticiens). Pour autant, une analyse des verrous actuels à l'optimisation de systèmes réels montre que ceux-ci se situent souvent à l'interface entre ces composantes.

Coût des simulations. le temps de calcul de $y(x)$ est l'obstacle le plus communément rencontré en optimisation. Cela restera ainsi à l'avenir car la complexité des modèles numériques croît plus vite que la puissance des ordinateurs. Tous les éléments constitutifs de l'optimisation appliquée sont impliqués dans le contrôle du coût numérique. Tout d'abord, le nombre possible d'analyses oriente le choix de l'optimiseur : sous 1000 analyses, on ne peut pas utiliser une méthode d'optimisation stochastique. Par ailleurs, le coût des simulations peut conduire à la création d'un autre modèle, numériquement plus efficace, qui puisse orienter l'optimisation à moindre coût (c'est le sujet de la multi-fidélité, cf. section 3.3.2). Enfin, un trop grand coût de simulation peut entraîner un changement de formulation, en particulier une décomposition du problème initial. Par exemple, dans [Luersen et al., J2006], la conception optimale d'une monopalme de nage est simplifiée en un problème bidimensionnel dont les solutions sont ensuite transposées en trois dimensions.

La présence d'optima locaux. Les développements de la simulation et de l'optimisation numérique ont été de pair avec de la multiplication des problèmes dont la régularité n'est pas connue et comportant des optima locaux. Les méthodes d'optimisation globales, i.e., capables d'échapper aux optima locaux, qui étaient des sujets de recherches dans les années 80 sont maintenant utilisées dans de nombreux bureaux d'études industriels. Ceci représente un important changement de perspectives, comparable à l'entrée de la non-linéarité dans le domaine de la

modélisation. Or, l'optimisation globale est un problème mathématiquement difficile, qui nécessite souvent l'ajout, aux approches d'optimisation généralistes, de connaissances spécifiques au problème traité. C'est pourquoi l'essentiel de ce document est consacré aux synergies entre visions mécaniciennes et mathématiques de l'optimisation. L'ajout de connaissances a priori peut intervenir à tous les niveaux de l'optimisation. Dans le plus simple des cas, une bonne solution a priori peut servir de point de départ à un optimiseur. Parfois, une solution connue modifie la formulation du problème: elle sert à régulariser un problème mal-posé (cf. [Le Riche and Guyon, T1999]), ou elle permet de fixer des objectifs à l'optimisation (un exemple en optimisation couplée process-structure est donné dans [Le Riche et al., J2003-2]). Dans d'autres cas, la connaissance a priori prend la forme d'une règle d'optimisation qui peut générer ou accélérer un optimiseur global : un exemple est donné avec le fully-stressed-design et l'optimisation évolutionnaire de formes en section 2.3; l'optimisation évolutionnaire de composites, dont nous traitons en section 2.2, constitue un excellent exemple d'optimisation globale intégrant de la mécanique.

Autres verrous. Il y a d'autres difficultés récurrentes en optimisation de systèmes mécaniques.

- La présence de paramètres incertains dans les fonctions optimisées, par exemple des bruits de mesures ou des conditions de fonctionnement mal contrôlées. Pour en tenir compte, il faut traiter deux problèmes imbriqués, l'optimisation du système et l'évaluation de l'effet des incertitudes sur ce système. L'optimisation en présence d'incertitudes est une perspective que nous détaillons en section 3.1.
- Les espaces de recherche non continus, e.g., les espaces mixtes et les espaces non paramétriques. L'optimisation de l'épaisseur de stratifiés composites décrite en section 2.2.1 est un exemple d'optimisation non paramétrique.
- La complexité croissante des logiciels de simulation fait qu'ils sont souvent vus comme des boîtes noires lors de l'optimisation. Ceci a deux effets : les sensibilités aux paramètres d'optimisation sont de plus en plus rarement calculées par les simulateurs et il est courant que les simulations échouent pour certains choix des paramètres.

Positionnement scientifique

Les problèmes d'optimisation qui demeurent d'actualité sont difficiles et méritent d'être abordés à la fois à travers la modélisation, la formulation, les algorithmes d'optimisation et la mise en œuvre informatique. Les travaux de recherche que j'ai réalisés portent sur l'optimisation de systèmes mécaniques, l'optimisation étant vue sous tous ces angles:

- *La modélisation* a été principalement abordée dans le cadre de l'apprentissage statistique [Rakotomamonjy et al., J2007; Rakotomamonjy et al., J2002; Le Riche et al., J2001];
- *La formulation*; dans [Gogu et al., J2008; Molimard et al., J2005; Molimard and Le Riche, J2003], nous avons étudié l'influence de la formulation des distances modèle-expérience à des fins d'identification; [Le Riche et al., J2003-2] est une étude sur la formulations de l'optimisation couplée d'une structure et de son procédé de fabrication; Dans la partie 1.3.2, la conception d'une monopalme est décomposée en deux étapes, une optimisation

en deux dimensions avec prise en compte des couplages fluide-structure et une optimisation structurale en trois dimensions, de manière à rendre le problème calculable; Le brevet [Bay et al., P2005] est essentiellement un travail sur la formulation du contrôle optimal de laminoirs;

- *Les méthodes d'optimisation généralistes* sont discutées dans le chapitre 1;
- *Les méthodes d'optimisation spécialisées* sont présentées dans le chapitre 2; dans l'esprit des théorèmes de No Free Lunch pour l'optimisation (cf. section 0.3), il s'agit de concevoir des algorithmes d'optimisation adaptés aux problèmes traités à savoir, la conception de structures composites et l'optimisation de forme. Des concepts venant de la mécanique sont inclus dans des algorithmes évolutionnaires.
- *La mise en œuvre* informatique est susceptible de faire gagner, ou perdre, beaucoup de temps à un projet d'optimisation. Elle peut aussi donner des idées ou les contraindre : tous les liens imaginables entre simulateurs et optimiseurs devraient pouvoir être réalisés. L'article [Le Riche et al., J2002] est une réflexion sur comment obtenir une grande liberté d'association entre simulateurs et optimiseurs au moyen de programmation orientée objet.

Contributions scientifiques

Les travaux présentés dans cette thèse sont le fruit de collaborations avec mes coauteurs et ont été influencés par des chercheurs du monde entier. Qu'ils soient tous remerciés pour leur aide, leurs idées et leur enthousiasme.

Une liste complète des publications que j'ai coécrites est donnée dans le chapitre 4.2. Une sélection d'article est reproduite en Annexes. Parmi ces publications, nous dégageons maintenant les travaux qui nous paraissent les plus importants. Ces travaux sont classés en contributions méthodologiques et applicatives.

Contributions méthodologiques

Mes coauteurs et moi même avons contribué au développement de méthodes d'optimisation généralistes. Nous mentionnerons en particulier les travaux suivants :

- Une nouvelle stratégie de prise en compte les contraintes d'optimisation basée sur la dualité a été proposée dans [Le Riche and Guyon, J2002]. C'est une méthode de pénalisation minimale, où la faible pénalité est obtenue au prix d'une perte de continuité. Cette méthode est particulièrement adaptée aux algorithmes évolutionnaires, mais elle peut être également utilisée dans tout autre optimiseur d'ordre 0.
- Une autre stratégie de prise en compte des contraintes basée sur la coévolution de solutions faisables et infaisables a été décrite dans [Le Riche et al., C1995-1]. Cette méthode, appelée SeGregated Genetic Algorithm (SGGA), est spécifique aux algorithmes évolutionnaires. A notre connaissance, c'était la première technique coévolutionnaire de traitement des contraintes en optimisation continue ([146, 147] ont développés des concepts similaires mais dans le cadre de la recherche opérationnelle).
- L'algorithme "Globalized and Bounded Nelder-Mead" (GBNM, cf. [Luersen et al., J2003]) est une amélioration du populaire algorithme de Nelder-Mead. Grace à une stratégie de redémarrage probabilisé, GBNM est global. De plus, il prend en compte les bornes sur les variables et les contraintes d'optimisation non linéaires.

- Un algorithme d'optimisation statistique à double distribution (DDOA) a été introduit dans [Grosset et al., J2006]. Bien qu'il n'ait été jusqu'à présent appliqué qu'en conception de composites, l'idée est générale : deux distributions de probabilités simples sont apprises à des fins d'optimisation, l'une dans l'espace des x , l'autre dans un espace de variables auxiliaires. Bien qu'aucune de ces distributions ne décrive les couplages entre variables, l'utilisation conjointe des deux distributions génèrent des lois couplées dans l'espace des x . De plus, les variables auxiliaires permettent d'introduire une information de haut niveau dans l'optimisation, les rigidités globales de la structure alors que les variables x sont des détails géométriques.
- Un critère d'optimisation globale basé sur le krigeage, l'amélioration espérée multi-points, a été proposé dans [Ginsbourger et al., C2007]. Il s'inspire du critère d'amélioration espérée de [93], mais, contrairement à celui-ci, il fournit plusieurs points à chaque itération. L'amélioration espérée multi-points permet donc des mises en œuvre parallèles et n'est pas un critère d'amélioration immédiate.

Nos travaux sur la spécialisation des optimiseurs évolutionnaires pour l'optimisation de structures ont été parmi les premiers dans le domaine.

- [Le Riche and Haftka, J1993] et [Le Riche and Haftka, J1995] ont ouvert la voie à de nombreux projets de conception de structures en composites par algorithmes évolutionnaires (e.g. [112, 91, 141, 151, 179, 145], entre autres).
- En optimisation de forme, nous avons développé une méthode combinant une heuristique de type fully-stressed-design et un algorithme évolutionnaire ([Le Riche and Cailletaud, J1998]).

Applications

Nous aimerions également mentionner deux familles d'applications prises parmi nos études: l'apprentissage de réponses d'automobiles et l'identification de paramètres de lois de comportement.

- Identification de réponses d'automobiles. [Le Riche et al., J2001], [Rakotomamonjy et al., J2002] et [Rakotomamonjy et al., J2007] avaient pour but d'estimer les efforts transmis au châssis, la masse instantanée, et le couple moteur, respectivement, à partir de mesures en clientèle d'accélération et des commandes du pilote (vitesse engagée et course de la pédale d'accélérateur en particulier). Comme l'état de la route, le vent et le chargement du coffre étaient inconnus, une modélisation physique n'était pas possible. Nous avons mis en œuvre diverses techniques d'apprentissage statistique (machines à supports vectoriels, réseaux de neurones, régression) pour réaliser les estimations. Dans tous nos travaux, les variables utilisées dans l'apprentissage et la forme fonctionnelle des régresseurs étaient trouvées au moyen de modèles physiques simples d'automobiles. Une illustration est donnée en FIG. 1.
- Identification de lois de comportement. Dans [Le Riche and Guyon, T1999], nous avons revu en profondeur les moindres carrés et l'algorithme de Levenberg-Marquardt. Outre une synthèse des principaux résultats sur l'identifiabilité, deux contributions éventuellement originales du rapport sont *i*) une preuve de convergence de l'algorithme vers la solution de moindre norme pour les problèmes linéaires et *ii*) une analyse de sensibilité réalisée à partir de la

trajectoire de l'algorithme. L'algorithme de Levenberg-Marquardt a été re-programmé avec prise en compte des bornes sur les variables et diverses techniques de mise à l'échelle. Ces travaux ont ensuite servi de base à [Molimard and Le Riche, J2003] et [Molimard et al., J2005].

Nous avons aussi contribué à l'identification de propriétés élastiques de solides à partir de mesures de champs ([Molimard et al., J2005], [Silva et al., J2007-1], [Silva et al., J2007-2]). FIG. 2 est une illustration extraite de ces travaux. L'influence de la formulation du résidu sur les propriétés identifiées a été étudiée. Nous avons montré dans [Molimard et al., J2005] que, du fait de la grande quantité de mesures, il était possible d'inclure des incertitudes expérimentales, e.g., un mauvais alignement d'éprouvette, parmi les variables identifiées.

Récemment, nous nous sommes efforcés de fournir une précision avec la valeur des paramètres identifiés. Dans [Silva et al., J2007-2], des intervalles de confiances sont estimés par ré-échantillonnage (bootstrap) des mesures, répétition des identifications, et statistiques sur l'ensemble des propriétés obtenues. Dans [Gogu et al., J2008] enfin, la distribution des paramètres estimés est obtenue directement par une technique Bayésienne.

Perspectives

Optimisation et incertitudes. Il est important pour les chercheurs en optimisation de réaliser que les simulateurs y qu'ils utilisent ne sont que des représentations imparfaites de la réalité. Il existe donc des incertitudes, qui peuvent être de nature déterministe, comme des paramètres expérimentaux inconnus, ou de nature aléatoire, par exemple un effort aérodynamique. Ces incertitudes peuvent le plus souvent être traduites dans les modèles sous la forme de paramètres incertains dont il faut tenir compte lors de l'optimisation [104, 180].

La conception en présence d'incertitudes, en particulier l'optimisation et l'estimation simultanées de quantiles, est donc une de nos perspectives. Elle est présentée dans la partie 3.1.

Une seconde perspective à nos travaux est l'identification, à partir de mesures de champs, de propriétés matériaux. La surabondance des données obtenues (1000 à 10000 mesures indépendantes) doit permettre à l'avenir *i*) de mieux identifier des changements locaux du matériau, c'est à dire de l'endommagement, et *ii*) de caractériser la distribution des paramètres matériaux identifiés au moyen d'une (ou de peu d') expérience(s), ce qui est une information plus riche que les intervalles de confiance. Ces perspectives sont détaillées en section 3.2.

Optimisation à coût fini. Il est clair que le coût numérique des simulations est, et restera, un facteur limitant à l'optimisation. De nombreux travaux s'attaquent à ce verrou en substituant aux simulateurs fins des modèles moins coûteux, parce qu'obtenus à partir de physiques dégradées ou bien appris sur des données [150, 177, 74]. Pour répondre au problème du temps de calcul, nous proposons une direction différente mais complémentaire. Il s'agit de développer des méthodes d'optimisation qui, explicitement, à chaque itération, prennent en compte le nombre de simulations restantes pour décider quelle région de l'espace du possible explorer. La prise en compte du temps de calcul encore disponible lors de l'optimisation permettra également de contrôler la finesse des simulations à venir. Ces idées sont décrites en section 3.3.

Optimisation distribuée. Ces vingt dernières années, la capacité mondiale en calcul haute performance (HPC) a doublé chaque année : elle était d'un teraflop

en 1994, le petaflop a été atteint en 2005 et, fin 2007, la HPC était estimée à 7 petaflops, dont 0.5 appartenait à l'ordinateur le plus rapide au monde. Cette accroissement des capacités de calcul n'est pas seulement dû aux super-calculateurs intégrés, mais aussi aux grilles de calcul, qui permettent la création d'équipements de grande capacité à partir d'ordinateurs faiblement connectés (e.g., la grille de production EGEE, [53]).

Le calcul distribué est une solution technologique pour réduire le temps de calcul de l'optimisation. Les disciplines les plus consommatrices en calculs, telles que la météorologie, la mécanique des fluides, l'astrophysique ou la génétique, utilisent déjà des ressources informatiques distribuées. L'optimisation en sciences pour l'ingénieur a également amorcé le virage vers la distribution des calculs [190, 199, 105, 143, 193], pas seulement d'ailleurs pour réduire le temps de calcul mais également parce que les acteurs de l'optimisation multi-disciplinaire sont éloignés géographiquement [67]. Néanmoins, l'essentiel des méthodes d'optimisation actuelles ont été conçues dans un cadre déterministe. Il est important aujourd'hui de penser l'optimisation de manière asynchrone. C'est une des conditions² au développement de méthodes qui passent bien à l'échelle d'un grand nombre de nœuds de calcul. Nos perspectives scientifiques à savoir, l'optimisation en présence d'incertitudes, l'identification à partir de mesures de champs et l'optimisation à coût fini, sont toutes susceptibles de bénéficier des technologies de distribution des calculs.

²Les autres obstacles au développement de l'optimisation distribuée sont la complexité et l'instabilité des middlewares impliqués et, en optimisation multi-disciplinaire, le volume de données à échanger entre disciplines.

Foreword

This thesis has been written in partial fulfilment of the French “Habilitation à Diriger les Recherches” degree (HDR, which means authorization to advise doctorate students). The first part of the dissertation consists of a summary of the research work accomplished between 1994 and 2008 by the author and his collaborators. The summary is partial because it will only concern the global optimization of mechanical systems: to preserve a unity of topic and avoid a long list of weakly related projects, works devoted to system identification and complex optimization problem formulations will not be covered in this text. The reader should also be informed that this dissertation is not a self-contained scientific text: the main results dealing with the global optimization of mechanical systems are presented and situated with respect to related works with references to the associated articles and reports. Chapter 4.2 is an extensive list of published articles, reports, patents, and seminars.

The text starts with general considerations on optimization in mechanical engineering. This introduction gives a high-level view of the engineering optimization activity. It explains the logic that relates the different technical results which are presented afterwards in two chapters: general global optimization algorithms (chapter 1) and specialized evolutionary algorithms.

My research perspectives are presented in chapter 3.

The last part of the dissertation is devoted to the other aspects of my research activities, in particular teaching, advising students, managing projects in collaboration with industry, research agencies and other universities, and reviewing. Selected publications are added as Appendices for easier reference.

Two standards are used in the text to cite publications: extended citations, e.g., [Le Riche et Guyon, C2001], refer to works that I have co-authored and are listed in section 4.2. They are composed of the authors names, followed by a letter and the year of publication. The letters P, J, C, S, T and V designate patents, journal article, conference articles, seminars, technical reports (including contract reports) and popularizing articles, respectively. Bracketed numbers refer to publications in which I was not involved. They are listed in section 4.1.

Optimizing mechanical systems

0.1 The different sides of an optimization problem in engineering

Optimization is the art of minimizing or maximizing functions or functionals. In any modelling or design effort optimization problems are embedded. Identifying a model involves minimizing a distance between experiments and model predictions. Modelling can often be formulated as energy minimization. For example, the equilibrium of a conservative system can be obtained by minimizing its total potential energy. And, of course, optimal design is also concerned with the maximization of performance criteria.

Optimization is therefore a versatile practice. Readers who are not familiar with optimization in engineering may find it difficult to understand what “optimizing a mechanical system” encompasses. This is particularly true in France where none of the academic committees (the “sections” of the Conseil National des Universités) can give a ruling on all of the engineering optimization activities. Related committees are applied mathematics, computer science, and the applications fields.

The next two sections aim at clarifying the problematics of mechanical system optimization so that all readers may understand what unites the scientific results that are presented later in the text: the problematics is decomposed into elementary tasks in section 0.1.1, then the links between these tasks are discussed in section 0.1.2. These sections will also introduce basic notations.

0.1.1 Optimization basic components

The optimization practice can be divided into the following elementary tasks:

Modelling is the task of defining the model y of the considered system. y is a vector of responses. Physical based models start with the writing of the basic equations that the system has to satisfy. It is followed by the implementation of a numerical³ solution. In the typical example of the design of a structure, the basic equations are those of solid mechanics (equilibrium, behavior laws and boundary conditions) and most numerical solutions involve finite elements softwares. y could consist of the mass of the system and finite elements nodes displacements.

The other type of models are statistical, i.e., learned from a given set of input x - output y . These a priori data come from measurements or other simulations (often from a calculation-intensive physics-based model).

³Analytical models are of great interest for developing search algorithms, but current real problems rarely have analytical expressions.

Note that there can be many models of the same system, which is the topic of multi-fidelity design. (See section 3.3.2.)

Formulation is the definition of the performance criteria f , of the solution's feasibility through the inequality constraints g , and of the optimization variables x ,

$$\begin{cases} \min_{x \in \mathcal{S}} f(x) \\ \text{such that } g(x) \leq 0 \end{cases} \quad (2)$$

where \mathcal{S} is a search space. f and g are in fact implicit functions of the optimization variables x and other (not optimized) problem parameters, e , through the model y , i.e., $f(x) \equiv f(x, e, y(x, e))$ and $g(x) \equiv g(x, e, y(x, e))$. To be comprehensive, one should add equality constraints, $h(x) = 0$. However, actual equality constraints are seldom encountered in engineering optimization. They usually express physics fundamental relations that must be satisfied, for example an equilibrium of forces, and are therefore taken care of in y . In many other cases, equality constraints are handled by removing one variable, say x_1 , by solving for it $h(x_1, x_2, \dots, x_n) = 0$. In most remaining cases, the equality constraint can be relaxed into two inequalities, $-\varepsilon \leq h(x) \leq \varepsilon$, ε a small positive constant. For these reasons, this manuscript only discusses inequality constraints. The topic of equality constraints regains importance in multi-disciplinary optimization (MDO, see [1, 117]) and simultaneous analysis and constraints (SAND, see [77]) because parts of the physical simulations and the optimizations are considered together.

Optimization algorithm After the formulation, some mathematical features of the optimization problem appear: is the search space, \mathcal{S} , continuous, discrete or mixed (between continuous and discrete variables)? How many variables and constraints are there? Do we know if the problem is linear, quadratic, or convex? Do we expect a unique solution, many local solutions, or a continuum of global solutions as in ill-posed problems? Are the objective functions and constraints noisy? The answers to these questions guide the choice of an off-the-shelf optimization strategy or the creation of a new algorithm.

Implementation In terms of programming, the implementation involves, firstly, the parameterization of the model y so that x can be changed easily, as will be needed for calculating the objective functions and the constraints. In the simplest parameterizations, the model y is completely recalculated for each new x . For calculation intensive models, it may save a lot of computing time to program a parameterization that only updates what is required when x changes. For example, if x is a vector of material parameters, it is a major waste of time to reload the geometry of a structure and remesh it. Next, the calculation of f and g from y is implemented. Finally, interfaces between the optimizer and the simulator can be created. The implementation typically involves a high labor input.

0.1.2 Engineering optimization: connecting the basic components

The above elementary optimization steps are traditionally treated separately in well-established fields. Modelling a system based on its physics is the topic of one or many particular physical domains such as fluid mechanics, solid mechanics, magnetism, thermics Modelling from data is the statistical data mining problem.

Devising optimization algorithms for treating well formulated optimization problems is an applied mathematics objective. Implementing models and optimizers and interfacing them together is a computer science activity.

However, the relations between these aspects of the optimization are numerous. In our experience, the links are most often created by the following elements.

Cost of a single analysis The computing time of $y(x)$ is the most frequently encountered obstacle to solving an optimization problem. It will remain this way despite progress made in computers because the models' complexity increases faster than the calculation capacities.

The number of affordable analyses guides the choice of the optimization algorithm. For example, below about 1000 analyses, no statistical optimization can be performed.

The cost of one analysis may have an influence on the model: a common technique to alleviate the analysis cost is to add a low cost metamodel to y . This metamodel will partly guide the optimizer. Such practice is called multi-fidelity optimization. A high analysis cost may also affect the problem formulation: it can be preferable to forego the optimality of the solution and decompose the problem into smaller, affordable, subproblems. [Luersen et al., J2006] is an example of problem decomposition: the design of a monofin is first treated in two dimensions to make it computationally affordable and the result is then translated into three dimensions.

Sensitivity information Local sensitivity analysis, that is, the calculation of $\partial f_i/\partial x_j$ and $\partial g_i/\partial x_j$, has been one of the founding tasks of engineering optimization. In structural optimization for example, great efforts were made from the 1980's to the turn of the century to calculate shape sensitivity information ([187, 72, 29, 5]). Sensitivity analysis connects all of the elementary steps of the previous section: the formal calculation of $\partial y_i/\partial x_j$ takes place at the model level; it is then propagated at the formulation level ($\partial f_i/\partial x_j = \sum_{k=1}^{ny} \frac{\partial f_i}{\partial y_k} \frac{\partial y_k}{\partial x_j}$) and subsequently needs to be implemented. The availability of sensitivity calculations guides the choice of the optimizer. Exact sensitivity information is so important that it will be used in the optimization algorithm if it is available. When sensitivities have not been calculated, they are often approximated by finite differences. Such approximation is costly when the number of variables, n , grows, and it is numerically noisy. The associated optimization strategy has to be economical and accept noisy gradients.

A priori information Moreover, one almost always has a priori information about the problem solution. Such a priori information can be accounted for at any of the elementary optimization levels.

The most frequently encountered situation is that of a **previously known good solution**. This is the case with redesign projects. One can simply choose this good solution as a starting point of the search, which has a minimal impact on the implementation. Knowledge of a good solution can also affect the problem formulation. In [Le Riche et al., J2003-2] for example, knowledge of the structural optimum design permits the setting of performance objectives (in the form of constraints) of the coupled process-structure optimization. Ill-posed identification problems, where there is an infinite number of solutions, can be regularized by minimizing the distance to a known good solution (e.g., [Le Riche and Guyon, T1999]).

Another type of a priori information is a **heuristic** which is known to improve solution points. For example, in structural optimization, the fully stressed design rule ([75] and see section 2.3) has been known since at least the 1960's ([163]). It was coupled with a genetic algorithm in [Le Riche and Cailletaud, J1998], which

illustrates how a priori information can modify the optimizer. In fact, many components of the algorithms for optimizing composite laminates that are presented in section 2.2 are based on making efficient optimizers from the a priori understanding of composite laminates.

Another feature is that f (or g) is **not calculable everywhere**. For example, if f is related to the orbit of a satellite that has been launched by a rocket of trajectory y , and if the rocket never reaches the targeted orbit (say it crashes shortly after take-off) for the given x , $f(x)$ cannot be calculated. Another simple example comes from material science when identifying material parameters from a traction-creep experiment: if the material parameters, x , are such that the ultimate stress that can be obtained by pulling on the specimen is smaller than the creep load, the creep response cannot be calculated. More generally, most simulators involving non-linear equations are prone to fail for certain x choices. When this is expected, the formulation, the optimizer and the implementation may be complemented accordingly: the formulation can penalize points in or near non-calculable areas; the optimizer should not fail when the objective function does not return a value; and the implementation should make the simulation hang up after waiting a maximum time for a response.

Experience feedback Since good optimizers may use model, formulation or implementation flaws to improve the performance of their solutions, early optimization results provide important information to improve all the previous steps.

Research in engineering optimization has its essence in considering all of the above basic optimization steps together.

0.2 Bottlenecks when optimizing mechanical systems

On the one hand, there are mathematical features that complicate an optimization problem, such as the design space dimension, the problem non-convexity, the problem irregularity⁴ and the occurrence of local optima⁵ to name a few⁶. These features are so fundamental that they receive constant research attention. Every optimization strategy starts by addressing them. Having optimization algorithms that can escape local optima and eventually locate global optima moved from being a theoretical research subject in the 1980's to being a demand from optimization users. It represents a major change of perspective, equivalent to going from linear to non-linear representations in modelling. The first two chapters of this dissertation will be devoted to the design of global optimization algorithms.

Mixed search spaces, i.e., problems where some of the variables are continuous while others are discrete, are a common bottleneck. In mechanical engineering, mixed search spaces are encountered when some of the components of the optimized system are chosen from a catalogue (e.g., truss members profiles, ply thicknesses in composite laminates, booster configurations and stage propulsion technologies for satellite launchers). The difficulty with mixed variables is that there is no stan-

⁴In Lipschitz continuous functions, the regularity is classically quantified by the Lipschitz constant. The fitness-distance correlation presented in section 1.1 is another regularity measure.

⁵The importance of local optima, or multimodality, can be measured for a given optimizer by the probability that the optimizer, when randomly started, asymptotically converges to a local (as opposed to global) optimum.

⁶One can always find counterexamples to such statements. In classification and regression, for example, it may be advantageous – at least in terms of direct value of classification and regression – to work in high dimensional spaces.

standard algorithm to handle these cases. It is rarely affordable to solve the continuous subproblem for each combination of the discrete variables. (If it is affordable, the problem becomes easy.) Branch-and-bound approaches ([119, 122]) permit the reduction of some of the combinatorial cost when accurate bounds on the optimal objective function can be obtained for a restriction of the initial problem. Although the paradigm is general, the branching and bounding operations are case dependent. These algorithms tend to remain costly because they are enumerative. Probabilistic search algorithms, in particular evolution strategies ([169, 11, 21, 54]), have standard versions that can be applied to mixed problems. This has contributed to the rapidly growing popularity of these methods in the last twenty years. More will be said about these algorithms in sections 1.1 and 2.2. Note that they remain costly search methods and do not offer a diagnostic about the optimality of their solution. Another important complication in optimization is the presence of uncertainties in the model that make the optimization criteria f and g noisy. Deterministic optimization approaches will typically fail to converge or will provide useless results. Measurement noise, uncertainties in the boundary conditions and material properties are common sources of randomness in mechanics. Optimization approaches that address this issue compose the domains of reliability-based design optimization (RBDO), robust optimization and stochastic optimization. Section 3.1 briefly returns to this topic which constitutes some of the perspectives of our work.

On the other hand, one may encounter practical difficulties when optimizing a system.

As already stated in the previous section, despite the progress made in computers, the simulation times (i.e., the time of one calculation of $y(x)$) have increased in the last 20 years because the models complexity typically grows faster than the computation speed. Of course, the optimized models have become more sophisticated in the meantime. However, the lesson is that optimization researchers should not expect an increasing available number of simulations in the future. To address the simulation cost problem, an important contemporary research direction that has given rise to the field of multifidelity optimization, is to substitute one or many metamodels for y (or f and/or g). Another current response to simulation cost is the distribution of the calculations among many computers. Because the limited simulation capacity is a determining factor for optimization, the GBNM algorithm presented in section 1.3 and the genetic algorithms of section 2.2 are designed to perform at a given number of analyses.

It has also become seldom for simulators (we mainly mean commercial non-linear finite element simulators) to provide sensitivity information. The reason is probably the complexity of implementing sensitivity calculation in an existing large code and the computation overhead that would result for non-optimizing users. Today, automatic differentiation is capable of tackling real size direct calculations (e.g., [130]), but its behavior on iterative numerical procedures is still not understood well [183]. It therefore seems to have become accepted by the engineering optimization community that simulators be handled as black boxes, with the sensitivity analysis being performed outside the code through metamodels and design of experiments.

As a last practical difficulty, we remind the reader of the non-calculable optimization criteria previously discussed in section 0.1.2.

It is important to address these optimization bottlenecks and develop more efficient, more robust, more versatile and more global search methods. Indeed, optimization is a key point in many analysis and design activities. Inferring new models from experimental data is an ill-posed optimization (identification) problem. Non-linear finite element simulators may experience convergence difficulties because of the underlying large scale optimization, that is, the residuals minimization in

the space of degrees of freedom. Convergence is even more difficult to achieve when dealing with non-differentiable (e.g., plasticity) or multi-modal phenomena (e.g., buckling). And, of course, the optimum design domain directly benefits from improvements in optimization methods.

0.3 General versus specific solutions : the No Free Lunch theorem

In 1995, a theoretical result known as the “No Free Lunch” theorem (NFL, [196]) shed a theoretical light on a phenomenon often noticed by optimization methods developers: when improving the performance of a search algorithm for a certain function, there is a simultaneous performance deterioration for another function. The first NFL theorem stated that “any two algorithms are equivalent when their performance is averaged across all possible problems”. The NFL theorem was established in discrete finite dimensional spaces, which is not a major practical restriction since real numbers are encoded in 32 or 64 finite arithmetic in computers. An illustration of the NFL theorem could be that any given search method which is more efficient than a random search for certain functions is less efficient for other functions.

A naive interpretation of the NFL theorem would be that there is no point improving search algorithms for some functions since, simultaneously, one makes them worse for other functions. Such reasoning overlooks that the NFL is based on an average of all problems, whereas one is always interested in a much more restricted problem class.

Other NFL theorems hold over more specific problems classes. By problem classes, we mean functions probability distributions. The above first NFL theorem discusses the uniform functions distribution, i.e., every function is equally likely. Other classes are theoretically possible, but it is necessary and sufficient for the NFL to hold that the considered functions have an invariant distribution with respect to any coordinate (x) permutation ([88]). I am not aware of any practical problem class that matches this condition. For example, if one takes into account all possible performance criteria of composite laminates, permuting the outermost and innermost plies in the function coding will profoundly affect each criterion and, as a consequence, will change the criteria distribution. Optimization algorithms are always meant to solve a small class of problems: the most general algorithms studied here hypothesize functions with essential optima (cf. section 1.1) in order to avoid unsolvable needle-in-a-haysack problems.

The NFL theorem is a result against general optimization algorithms and in favor of specialized algorithms. By specialized we mean that “(the problem) structure must be known and reflected directly in the choice of the algorithm” ([197]). In theory, the specialization should be carried out beforehand, in the design of the algorithm. Checking a posteriori may be misleading: it is not because an algorithm performs better on a certain class of problems than another that it is specialized. Both can have a poor performance. To be pertinent, research in optimization should link the optimization algorithm to its targeted problem.

0.4 Scope of this dissertation

0.4.1 Our research process

In the previous pages, we have introduced elements that can now be assembled to explain the logic that unites our work.

We have decomposed optimization problems into elementary tasks (section 0.1.1) and shown how these tasks are connected by a particular problem (section 0.1.2). The optimization problems that are of interest are difficult and should be tackled from all sides. That is the reason why we have considered the following research subjects:

- *Modelling* was mainly addressed through system identification [Rakotomamonjy et al., J2007; Rakotomamonjy et al., J2002; Le Riche et al., J2001];
- *Optimization formulations*; in [Gogu et al., J2008; Molimard et al., J2005; Molimard and Le Riche, J2003], we looked for adequate formulations of identification problems; [Le Riche et al., J2003-2] compared different formulations for simultaneously optimizing a structure and its manufacturing process; in section 1.3.2, the design of a swimming monofin is decomposed into subproblems in order to become computationally tractable; [Bay et al., P2005] is a work on the formulation of an optimal control problem.
- *Optimization algorithms* will be discussed at length in chapters 1 and 2.
- *Implementation issues* are a practical aspect of every optimization project. [Le Riche et al., J2002] was a reflection on how to achieve flexibility in an optimization software using programming objects.

Section 0.2 gave a short diagnostic on optimization bottlenecks. Many of these will be addressed in this dissertation: the presence of local optima, the computational cost of a single analysis, the absence of sensitivity information and the programming of simulator-optimizer interfaces.

Finally, we have argued in section 0.3 that optimization problems require specialized algorithms. Accordingly, our work is focused on mechanical applications, among which the design of composite structures and the identification of constitutive law parameters have particularly held our attention. Whenever possible, our thought process aims at exploiting synergies between the mathematical and the mechanical view points.

0.4.2 Structure of the dissertation

The dissertation starts with a chapter on general global optimization algorithms. This chapter 1 gives an introduction to evolutionary algorithms that will be useful later on when speaking of constraints handling and specialized evolutionary based approaches. It also describes two non-evolutionary global optimization algorithms, the Globalized and Bounded Nelder-Mead algorithm (GBNM, section 1.3) and the Parallel Efficient Global Optimization (1.4). The chapter concludes with a section on constraints handling strategies that are compatible with these global optimization algorithms.

The specialization of evolutionary algorithms in specific problems is the subject of chapter 2. The evolutionary optimization paradigm is indeed particularly well suited to incorporating problem-specific knowledge. The considered knowledge domains are composite structures (section 2.2) and shape optimization (section 2.3).

Chapter 3 gathers our scientific perspectives. The dissertation finishes with a summary of other professional activities, that is, teaching, project management and reviewing. Copies of our main publications are included in the appendices.

0.4.3 Scientific contributions

Our work constitutes an example of research in engineering optimization. We have most often proceeded with a problem-solving type of approach, i.e., starting from the problem and sometimes reaching methodological developments. Such a progression has high chances of yielding a practical result and the application is an important source of theoretical ideas. Nevertheless, we have always felt the need to balance this approach with efforts starting from theoretical, mathematical bases. In hindsight, we feel that we have not taken enough time to proceed in this direction, from the theory onwards, and plan to achieve a better balance between the two approaches in the future.

The work presented here is the product of collaborative work with all of my co-authors and has been influenced by many other researchers from all over the world. I am grateful to them for their help, ideas and enthusiasm.

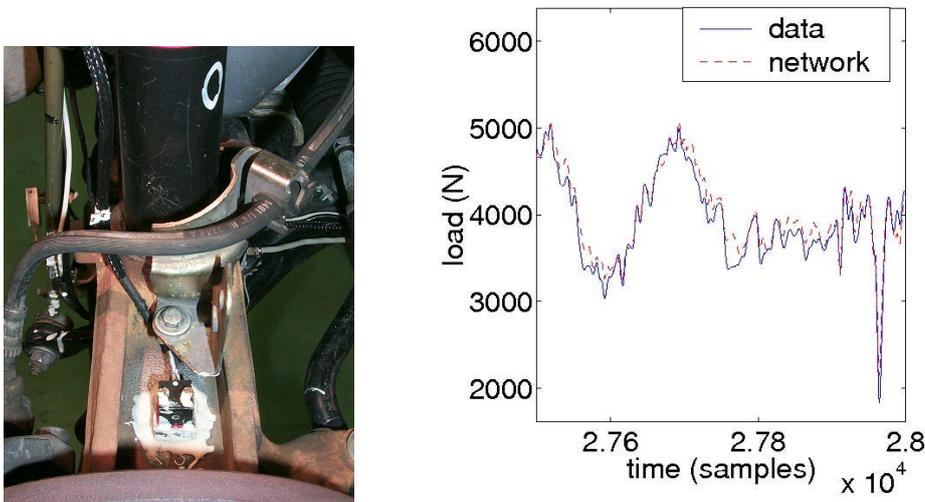
An extensive list of the publications I have co-authored can be found in chapter 4.2. A selection of those is copied in the appendices of this dissertation. Out of these publications, we now outline the contributions which we find most important. They are classified into methodological and applicative contributions.

Methodological contributions

My co-authors and I have contributed to the development of general global optimization methods. In particular, we would like to mention the following works:

- A new strategy for handling constraints based on duality has been proposed in [Le Riche and Guyon, J2002]. It achieves a minimal amount of penalty, at the expense of losing continuity. It is particularly adapted to evolutionary algorithms but can also be used with other zeroth-order optimizers.
- Another constraint handling strategy based on the coevolution of feasible and infeasible solutions has been proposed in [Le Riche et al., C1995-1]. The method, the SeGregulated Genetic Algorithm (SGGA), is specific to evolutionary algorithms. To our knowledge, it was the first constraint handling strategy based on coevolutionary concepts applied to numerical optimization ([146, 147] have developed comparable ideas earlier but in operations research).
- The Globalized and Bounded Nelder-Mead algorithm is an improvement of the well-known Nelder-Mead algorithm. It is global (thanks to a fixed cost restart mechanism) and handles variables bounds and non-linear constraints (see [Luersen et al., J2003]).
- A double-distribution statistical algorithm for optimization has been devised in [Grosset et al., J2006]. Although it has only been applied to composite laminates optimization so far, the concept offers wide-ranging applications: two simple probability density functions (pdfs) are learned for optimization purposes, one in the x -space, the other in an auxiliary space. Although none of these pdfs accounts for variables couplings, the joint use of the two can create complex pdfs with variables interactions in the x -space, which would be have been difficult to learn directly. The auxiliary variables are also a way to incorporate high level information⁷ into the search.
- In [Ginsbourger et al., C2007], a global optimization criterion based on kriging has been proposed. Like EGO ([93]), it is an expected improvement

⁷E.g., global stiffnesses for structures when the primal variables are detailed structural dimensions.



(a) Acceleration measurement system.

(b) Comparison between measured and predicted forces on a real vehicle. The prediction is based on an autoregressive network and the data was not used in learning.

Figure 1: Identification of vehicle mass, loads in the suspension chamber and torque from online acceleration and pilot controls measures. Cf. [Le Riche et al., J2001; Rakotomamonjy et al., J2002; Rakotomamonjy et al., J2007].

criterion. The originalities are, firstly, that it is not an immediate payoff optimization criterion and, secondly, that many points are provided at each iteration, which makes it suitable for parallel optimization. Approximations to this multi-point expected improvement that are computationally tractable have been devised.

We have also made early contributions to specializing evolutionary optimization algorithms for mechanical problems.

- [Le Riche and Haftka, J1993] and [Le Riche and Haftka, J1995] were among the very first of a large series of projects on the design of composite laminates by specialized evolutionary algorithms (e.g. [112, 91, 141, 151, 179, 145], among others).
- Similarly, in shape optimization, we have developed an approach combining a fully-stressed based heuristic and an evolutionary algorithm ([Le Riche and Cailletaud, J1998]).

Applicative contributions

We also would like to mention two application classes from the body of work, the identification of automotive systems and the identification of behavior law parameters.

- Identification of automotive systems. [Le Riche et al., J2001], [Rakotomamonjy et al., J2002] and [Rakotomamonjy et al., J2007] aimed at estimating the load in the suspension chamber, the instantaneous

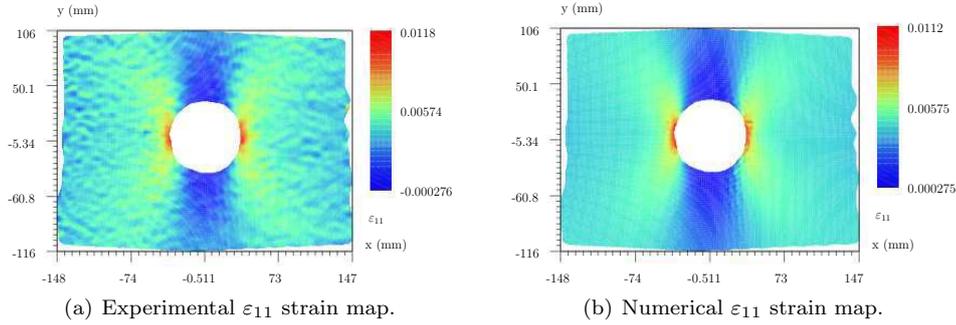


Figure 2: Identification of the elastic properties and the orientation of a composite plate with a hole. Comparison of (a) the experimental with (b) the numerical ε_{11} strain map after identification. $T = 1, 250$ KN. (Cf. [Silva et al., C2007-2].)

mass and the engine torque from accelerations and pilot controls measurements (engaged gear and accelerator position mainly). Because the measurements were made during customer surveys, the state of the road, the wind and the cargo are unknown, a direct physical model cannot be calculated. We relied on various statistical learning techniques (support vector machines, neural networks, least squares regression) to make the estimations. In every case, the regression variables and the functional form of the regressor were inferred from simple physical models of the system. See FIG. 1.

- Identification of constitutive law parameters. In [Le Riche and Guyon, T1999] we revisited non-linear least squares and the Levenberg-Marquardt algorithm in depth. Besides gathering important classical results on identifiability and uniqueness, two original contributions of the report may be the proof of the algorithm's convergence to the minimum norm solution for linear problems and sensitivity analysis based on the algorithm's trajectory. An implementation of the algorithm has been made with variables bounds handling and careful scaling. This work has been of great use in other parameters identification studies ([Molimard and Le Riche, J2003], [Molimard et al., J2005]). Non-linear least squares based identification has been carried out with full-field measurements in solid mechanics ([Molimard et al., J2005], [Silva et al., J2007-1], [Silva et al., J2007-2]). FIG. 2 provides an illustration of the work on identification from full-field measurements. The influence of the residual formulations on the identification convergence has been studied. The possibility of including uncertainties of the experimental set-up in the identification procedure has been proposed and investigated ([Molimard et al., J2005]).

We have recently started to identify the parameters and their accuracies. In ([Silva et al., J2007-2]), confidence intervals stem from repeated identifications and subsequent statistics. Bayesian identification estimates the parameters distributions. Bayesian and least squares identifications have been compared in ([Gogu et al., J2008]).

Chapter 1

General global optimization algorithms

1.1 Evolutionary optimization

1.1.1 Context

Evolutionary algorithms (EAs), of which genetic algorithms are popular examples, are forty years old: in the last fifteen years, they have been applied intensely to difficult engineering optimization problems. Today, they no longer represent an engineering fashion fuelled by a biological metaphor. Accumulated experience enables us to draw lessons from the advantages and limitations of EAs, and how they compare to other global optimization methods.

By default, when dealing with EAs we will address the following (mono-objective) global optimization problem

$$\begin{cases} x^* \in \mathcal{S} \\ x^* = \mathit{Argmin}_{x \in \mathcal{S}} f(x) \end{cases} \quad (1.1)$$

where f goes from the x search space \mathcal{S} to \mathbb{R} . Alternatively, the problem is sometimes transformed into a maximization of a fitness function F . The discussion of constrained problems such as Problem (2) is postponed until section 1.2.

The only hypothesis made on \mathcal{S} is that it is a topological space, i.e., a space in which neighborhoods exists. This assumption is necessary to define local solutions, $\bar{x} \in \mathcal{S}$, such that

$$\begin{cases} \exists V(\bar{x}), \text{ neighborhood of } \bar{x}, \\ \bar{x} = \mathit{Argmin}_{x \in V(\bar{x}) \cap \mathcal{S}} f(x) \end{cases} \quad (1.2)$$

On the contrary, solutions to (1.1) are global.

Two optimization problem classes need to be distinguished. The optimization is parametric whenever \mathcal{S} satisfies one of the following inclusions,

$$\mathcal{S} \subset \mathbb{R}^n \quad \text{or} \quad \mathcal{S} \subset \mathcal{D}^m \quad \text{or} \quad \mathcal{S} \subset \mathbb{R}^n \times \mathcal{D}^m, \quad n, m \in \mathbb{N} \quad (1.3)$$

where \mathcal{D} is a finite discrete set. The optimization variables are the components of $x \in \mathcal{S}$. They are taken in \mathcal{S} continuous (a compact of \mathbb{R}^n), discrete (\mathcal{D}^n) or mixed (continuous and discrete). The remaining cases, for which there are no integers (n, m) such that (1.3) stands, are called **non-parametric**. To avoid numerical problems, we further assume that the global optima x^* of (1.1) are such that $f(x^*) > -\infty$. In continuous parametric cases, it is also assumed that the global optima

are not isolated¹, in order not to focus on artificially deceptive problems and to allow global convergence proofs. No other hypothesis on \mathcal{S} and f is required to treat (1.1). Contrary to mathematical programming, no regularity (e.g., Lipschitz regularity), continuity, differentiability or convexity is needed. So few assumptions are not specific to EAs but it is a property of stochastic optimizers. For example, simulated annealing ([110]) is also very flexible in terms of mathematical framework. Consequently, stochastic optimizers, EAs in particular, can be applied to a wide range of situations. The price of such ubiquity is the large number of analyses EAs typically take to locate good solution points. As the No Free Lunch theorem hinted, this number of analyses depends on the problem. Moreover, since EAs are a large family of algorithms, the performance also depends on the particular EA implementation and its suitability to the problem at hand (cf. paragraph 1.1.4).

1.1.2 Overall structure of an evolutionary algorithm

The Darwinian metaphor

Evolutionary algorithms are stochastic optimization methods that produce a set of points in \mathcal{S} at each iteration using stochastic operations (cf. example in FIG. 1.1).

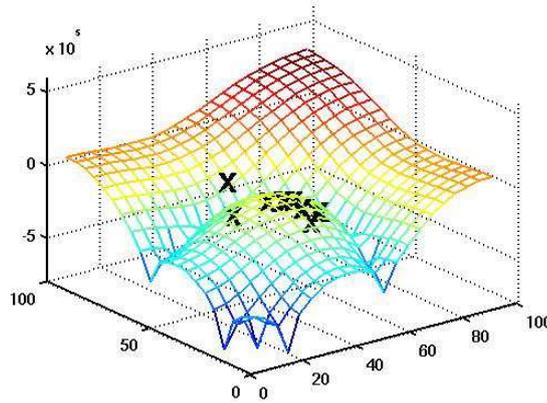


Figure 1.1: Evolutionary optimizers handle a set of points (a “population”) at each iteration. In this example, the objective function f has local optima. f expresses the in-plane stiffness A_{11} of a composite laminate penalized so that Poisson’s ratio is bounded $0.48 \leq \nu \leq 0.52$. The variables x_1 and x_2 are the material fiber orientations, cf. [Grosset et al., J2006]

It is a common practice to introduce EAs through a Darwinian metaphor: a point x in \mathcal{S} is dubbed an *individual* of a given species. An overly simplified version of Darwin’s theory is that species evolve through two mechanisms, *blind variations* that occur during reproduction and *natural selection* that favors those individuals who are most adapted to their environment. If the individuals are associated with points in \mathcal{S} and the adaptation level is quantified by the objective function f , such an evolution process can be seen as solving the optimization problem (1.1). The main metaphorical terminology is summarized in TAB. 1.1. More generally,

¹In mathematical terms, only essential global optima are tracked, i.e.,

$$f^* = \min\{y \mid \forall \varepsilon > 0, v(x \in \mathcal{S} \mid f(x) < y + \varepsilon) > 0\}$$

where v is a set measure such as $v(E) = \text{Volume}(E)/\text{Volume}(\mathcal{S})$.

individual, phenotype	x
chromosome, genotype	coding of x
gene	one component of x 's coding
allele	a gene's – or component's – value
fitness, performance, adaptation	$F(x)$
population	$\mathcal{P} = \{x^1, \dots, x^\mu\} \in \mathcal{S}^\mu$
generation	change of population from \mathcal{P}^t at time t to \mathcal{P}^{t+1}

Table 1.1: The Darwinian metaphor.

frequent allusions to biological concepts can be found in EAs literature (e.g., population niches [127], chromosome's diploidy [65], ...). We will sometimes use this metaphor because it helps to explain and even devise new EAs. However, we will keep in mind that a metaphor does not justify algorithmic choices.

Evolutionary algorithm skeleton

FIG. 1.2 shows the flowchart of a generic evolutionary algorithm, which encompasses genetic algorithms (GAs, cf. section 1.1.4), evolution strategies (ES, cf. section 1.1.6) and estimation of density algorithms (EDAs, cf. section 1.1.6). To keep the flowchart simple, FIG. 1.2 accounts neither for EAs with varying population sizes, nor for EAs with a memory (or archive, [121]), nor for EAs that have subpopulations (as in parallel implementations, [129], or coevolutionary versions, [123, 146]).

The optimizer state at time t is completely determined by the μ individuals of the population \mathcal{P}^t and their performances. The initial population, \mathcal{P}^0 , can be randomly chosen in \mathcal{S} . Alternatively, \mathcal{P}^0 can be initialized in a way that controls its composition, e.g., the number of 1's in binary chains ([99]). Some of the individuals of \mathcal{P}^0 can also be chosen by the user, which is a simple way to inject a priori knowledge into the search. Iterations are made of selection and/or replacement, and stochastic variations. Examples of selection, replacement, crossover and mutation will be given in the paragraphs 1.1.2.

The principle of evolutionary searches is the following: selection and replacement take samples from the population \mathcal{P}^t in a biased way that favors the individual's performance (so-called "natural" selection). Selected individuals, or parents, undergo blind, i.e., independent of the performance, variation operators, which yields λ new points or children. In classical EAs (see section 1.1.2), the variation operators are known as crossover and mutation. In EDAs, the variation operators are the updating of a probability density function p^t followed by its sampling (cf. section 1.1.6). The function of crossover is to transmit the parent's beneficial features to the children (*your brain and my beauty*); preserving the population diversity and exploring the search space are objectives of the mutation (more will be said on this topic in 1.1.3). As illustrated in FIG. 1.3, the variation operators, both crossover and mutation, as well as the initialization, work in the space of genotypes (the space of the codings of x). On the contrary, selection and replacement depend only on the phenotypes (the optimization criteria) and are independent of the coding.

Variation operators create stochastic transitions in \mathcal{S}^μ . These transitions are biased towards highly performing individuals by the phenotypic operators. Three stopping criteria can be used: a maximum number of calls to f (and/or g for constrained problems, or any measure of a maximum simulation time), a maximum number of iterations without improvement of the best known point, and a minimum population diversity².

²Population diversity can be measured in terms of the f 's, e.g., σ_f/f^{ref} , where σ_f is the

To finish this introduction, the following paragraphs provide examples of usual selection and variation operators, without discussing the details of various possible implementations of EAs. Even though there are standard EA implementations (for example, CMA-ES for continuous parametric optimization, cf. section 1.1.6), as soon as the problem is non-parametric, there is no other choice but to create a specialized EA from variation operators defined for the specific search space considered, and mixed in the spirit of the flowchart of FIG. 1.2. The examples given hereafter should be seen as components to be assembled. We will provide some rules whereby this assembly will result in an efficient optimizer.

A reader willing to complete this introduction may refer to the following books: [66] is an historical book, even if the hopes raised by the schema theory are now thought not to be founded (cf. paragraph 1.1.4). A good introduction to modern EAs that is still relevant today is [135]. A contemporary book on EAs is by Eiben and Smith (2003) [54]. The recent book by K. DeJong [96] gives a unified view of the evolutionary calculation field. French readers may want to consult [Le Riche et al., J2007].

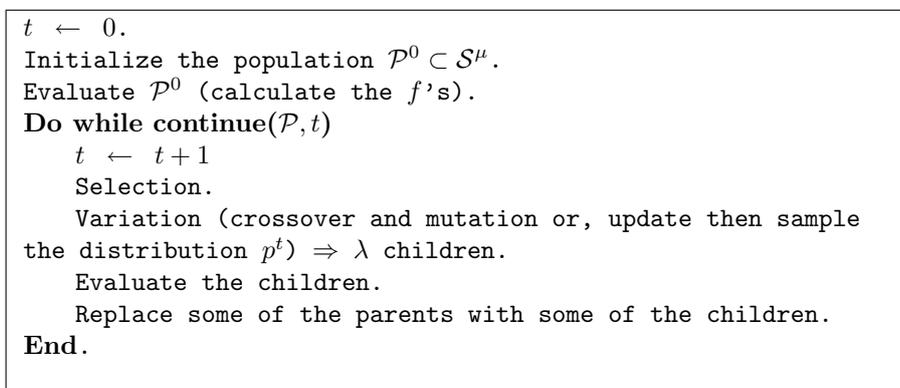


Figure 1.2: Generic flowchart of an evolutionary algorithm with μ “parents” and λ “children”. \mathcal{P}^t is the population at iteration t . For EDAs, p^t is an estimate of the good points density at iteration t . For CMA-ES, p^t is a distribution of good steps (cf. section 1.1.6) ...

Phenotypical operations : selection and replacement

Selection chooses a few individuals from a population with a probability based on their performance (f and g). The underlying idea of a probabilistic selection is that it may be beneficial for the optimization to have a small, yet non-null, probability of keeping low performance individuals. This is because there is typically a bad correlation between performance and distance to the optima in difficult problems (cf. section 1.1.5). It is therefore sometimes advantageous not to select in a deterministic fashion.

A simple yet fine selection implementation is the size τ **tournament**. Its pseudo-code is:

standard variation of the f 's in \mathcal{P}^t and f^{ref} is a normalization quantity. Alternatively, diversity can be calculated in S , but for large dimensional spaces the calculation may become numerically costly.

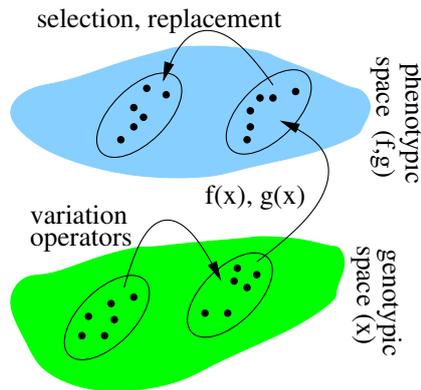


Figure 1.3: Within the evolutionary metaphor, the working space of variation operators and the space of optimization performance criteria (f and g) where selection operates are called “genotypes” and “phenotypes”, respectively (from Lewontin, 94).

```

TournamentSelection( $\mathcal{P}$ )    % Usually,  $\mathcal{P} \subset S^\mu$ ,
                          % parents' population
  Take  $\tau$  individuals uniformly in  $\mathcal{P}$  without replacement
  Return the individual with optimal performance
End

```

The procedure is repeated when many individuals are needed. For example, in genetic algorithms (GAs), one would apply³ `SelectionTournament` λ times on \mathcal{P}^t (sampling with replacement) to obtain λ children. Tournament selection has two advantages. Firstly, it is not sensitive to the values of f and g , but only to the rank. This is important in order to maintain a constant selection pressure⁴ even when the individuals in the population become similar to one another. Secondly, the selection pressure can be tuned by changing τ : if $\tau = \mu$, the population’s best individual is always selected. At the other extreme, if $\tau = 1$, tournament selection degenerates into random selection. $\tau = 2$ is usually the recommended value ([11]). There are many other selection implementations. We can cite selections based on the performance value (e.g., the roulette wheel of GAs, [86]), and selections whose pressure differential follows a Boltzmann law such as simulated annealing ([128]).

Replacement, as the name suggests, replaces some of the parents with some of the children depending on their respective performances. Evolution strategies (ES, cf. section 1.1.6) exclusively rely on replacement rather than on selection in order to bias the search towards optimal regions of S . In ES, replacement is deterministic and can take two forms, written as (μ, λ) -ES and $(\mu + \lambda)$ -ES. In (μ, λ) -ES replacement, the μ best individual among the λ children⁵ make up the next population, \mathcal{P}^{t+1} . \mathcal{P}^t is not kept. It is therefore possible that the best individual at iteration $t + 1$ is not as good as the former best at iteration t . Note that the accidental loss of the best solution may seem difficult to accept, particularly when the simulation is costly. However, this is a trait that favors global search and auto-adaptation

³Historically, GAs did not use the tournament method but rather a selection probability proportional to performance known as “roulette wheel”, [86]. See section 1.1.4.

⁴Selection pressure is defined as the expected number of offsprings in the population \mathcal{P}^{t+1} of \mathcal{P}^t ’s best individual. It is related to the take-over time ([10]), which is the expected number of iterations before the population is entirely composed of copies of the same best individual as a result of the application of selection only.

⁵It is necessary that the number of children λ be larger than the number of parents μ . In practice, a recommended ratio is $\lambda/\mu = 7$ [11].

(cf. section 1.1.6): one needs to accept that performance may decrease from one generation to the next in order to have the chance to locate global optima. Refusing the loss of the population's best individual also stems from the confusion between the search and memory roles of the population. If memory is added to (μ, λ) -ES, the simplest version of which would be to remember the best ever solution found, the incentive to non-elitist⁶ EAs becomes stronger.

The $(\mu + \lambda)$ -ES replacement builds \mathcal{P}^{t+1} by taking the μ best individuals from \mathcal{P}^t and the λ children. If the best individuals of \mathcal{P}^t remain competitive with respect to the children, they are kept (this is the elitism property). $(\mu + \lambda)$ -ES is recommended when the calculation cost of the analyses is high, with the risk of converging to local optima.

Before we turn to the variation operators, note that we will not cover an important phenotypic aspect of EAs, the (selection in) multi-objective optimization. Multi-objective optimization has been one of the major successes of evolutionary algorithms. We omit it here because our research has only marginally concerned multi-objective optimization (with the exception of [Le Riche et al., J2003-1]). For French speakers, multi-objective evolutionary optimization is briefly discussed in [Le Riche et al., J2007]. An extended discussion is provided in the specialized books [40, 31] and further information can be found in the specialized conferences [200, 58, 25].

Genotypic variation operators

We will now illustrate the preceding discussion on three common search spaces: real variables, discrete variables, and permutations. The reader should not forget that EAs can be modified to better fit a particular problem (see chapter 2 on specializing EAs). The variation operators are, initially, a crossover that produces an individual x' , and subsequently a mutation that yields x^e (densities p^t are the subject of section 1.1.6). The crossover operator takes as input ρ previously selected points, either directly by the selection operator, or indirectly through the replacement. In the latter, the ρ points are randomly taken from the current population \mathcal{P}^t . Some EAs do not use crossover, in which case x' is the copy of a selected individual.

The purpose of the crossover is to recombine features of points that have been selected in order to create new points. In particular, it is necessary that the crossover transmits to the children features that all of their ρ parents share. This is the “respect” property of [152]. The mutation operator stochastically perturbs individuals from the current population. Without mutation, under the combined action of selection (and/or replacement) and crossover, the population would converge to a single point that depends on the initial population. Such loss of population diversity is called *premature convergence*. EAs can not be global optimizers without mutation.

Continuous variables

Let us now consider continuous variables, $x \in \mathbb{R}^n$. An example of crossover between ρ (previously selected) points x^1, \dots, x^ρ is⁷

$$x'_i = \sum_{j=1}^{\rho} \beta_j x_i^j \quad , \quad i = 1, n \quad (1.4)$$

⁶The elitism heuristic is to copy the best individual of \mathcal{P}^t into \mathcal{P}^{t+1} .

⁷This crossover is given here as a general form. When $\rho = 2$, it is the “panmictic generalized intermediate recombination” of [11]. When $\beta_j = 1/\rho$, it corresponds to the “global intermediate recombination” of [169].

with

$$\beta_j \geq 0 \quad , \quad \sum_{j=1}^{\rho} \beta_j = 1 \left(\text{for expl., } \beta_j = \frac{\alpha_j}{\sum_{k=1}^{\rho} \alpha_k} \quad , \quad \alpha_k \sim U[0,1] \right) . \quad (1.5)$$

Note that here, x' is in the convex hull of the ρ parents. By reducing the convex hull from one generation to the other, this may jeopardize the chances of finding a global optimum. To avoid such a contracting crossover, it is recommended to take α_k uniformly in $[-.5, 1.5]$ [135].

x' is then mutated to yield a child point, x^e , typically using a multi-gaussian law centered on x' and having covariance C ,

$$x^e = x' + \mathcal{N}(0, C) \quad (1.6)$$

The choice of C induces an average step size and privileged directions. It is discussed in section 1.1.6. If the variables are independently mutated, C is a diagonal matrix with $C_{ii} = \sigma_i^2$ [11].

Discrete variables

Let us turn to discrete variables. Each x_i , $i = 1, n$, belongs to an alphabet $\{1, 2, \dots, A\}$. The crossover between x^1, x^2, \dots, x^ρ can take the following expression (this is the **uniform crossover** or ρ -ary discrete recombination of [169]).

With probability p_c , do,

$$x'_i = x_i^j \quad \text{where } j \text{ is randomly taken from } \{1, \dots, \rho\} \quad (1.7)$$

If the crossover does not take place (with probability $1 - p_c$), one of the ρ selected individuals is copied as x' . Note that if the variables x_i are seen as independent random variables, the uniform crossover is a bootstrap procedure [52] from the parent's population. This interpretation is important because it explains the link between traditional EAs (with crossover and mutation) and estimation of distribution algorithms: performing uniform crossover and sampling from UMDA or PBIL (these estimation of distribution algorithms are discussed in section 1.1.6) is equivalent. It amounts to creating new points from the distribution underlying the parent's population.

Each component of x'_i can then be independently mutated with probability p_m . For example,

$$x_i^e = j \quad , \quad j \neq x'_i \quad , \quad \text{with probability } 1/(A-1) . \quad (1.8)$$

This corresponds to a mutation without correlation between variables. Correlated mutations are discussed in the part on EDAs and in [140] and [Grosset et al., J2006].

Permutations

Finally, we will briefly discuss the optimization of permutations. Permutations are ordered lists of n distinct elements. There are $n!$ possible permutations. The simplest example is the traveling salesman problem that consists of the determination of the shortest tour linking n given cities. Permutations appear in many other ordering problems. They present specific implementation issues for evolutionary optimization: it becomes non-trivial to propose a coding and the associated crossover that yield valid permutations. For example, if the permutations are encoded as ordered lists and one seeks to cross the two following permutations (one is in subscripts, the other in superscripts) with the uniform crossover of discrete variables, one could have

$$\begin{aligned} x^1 &= [A B C D E F G H I] \\ x^2 &= [d f e b a g c h i] \\ x' &= [A f e D a g G H I] \end{aligned}$$

which is not a valid tour (two a's, no b, ...). One of the oldest and simplest operators for permutations coded as ordered lists is the ordered crossover by Davis ([37]): *i*) Take two cutting points randomly in the parents x^1 and x^2 . *ii*) Elements of x^1 located between the cutting points are copied into x' . *iii*) Starting from the second cutting point and going to the start of the list when its end is reached, find the next elements of x^2 that do not appear in x' . Copy them, still starting from the second cutting point.

$$\begin{aligned} i) \quad x^1 &= [A B C | D E F G | H I] \\ i) \quad x^2 &= [d f e | b a g c | h i] \\ ii) \quad x' &= [- - - | D E F G | - -] \\ iii) \quad &\text{non copied elements : h i b a c} \\ iii) \quad x' &= [b a c | D E F G | h i] \end{aligned}$$

This crossover preserves relative and absolute positions for x^1 between the cuts as well as relative positions for x^2 outside the cuts. There are many other crossovers for permutations. For example, the “edge recombination” operator ([195]) seems particularly adapted to the traveling salesman problem because it respects the edges (parts of the tour that determine the tour total length).

The simplest mutation of a permutation is an inversion of the order between two randomly chosen positions (2-opt operator of [125]). For example, if the inversion occurs between positions 3 and 6,

$$x^e = [b a | F E D c | G h i] .$$

Evolutionary optimizer characteristics

Evolutionary optimization algorithms therefore have four characteristics:

1. The EA produces many points at each iteration, as opposed to methods that change one point at a time (e.g., mathematical programming [138], greedy search, simulated annealing). This bestows a higher robustness for multimodal functions on EAs since the search occurs in the volume of \mathcal{S} , and not on trajectories in \mathcal{S} as is the case for point-wise optimizers. However, this robustness has a numerical price, that of evaluating μ points instead of one. Therefore, it is not guaranteed that, at a fixed number of analyses, an EA is more efficient than many local searches with a random restart.

There are, however, functions for which the evolution of a population leads to a faster search than the evolution of a unique point. Let us compare in \mathbb{R}^n a $(\mu+\lambda)$ -ES, $\mu > 1$, and a (1+1)-ES with Gaussian mutation and no crossover in a sphere-tunnel function (see FIG. 1.4): as soon as (1+1)-ES has found the tunnel, it is doomed to follow this tunnel because any point outside has a lower performance. Following a tunnel can be made arbitrarily long by narrowing it. $(\mu+\lambda)$ -ES can hit the tunnel but, since it has a population, some of its individuals typically remain outside it, which makes it easier for them to approach the optimum. Notice that tunnel topologies are not unrealistic. They reflect what happens in constrained optimization with penalty functions. The tunnel interior corresponds to the feasible domain. For a fixed number of f (and g) analyses, the advantage of population-based approaches has also been proven for noisy functions [4] and empirically seen for the auto-adaptation (cf. [21] and section 1.1.6).

2. Variation operators and variables coding must be designed together. The only explicit example of coding choice we have seen so far concerned permutations (cf. section 1.1.2). Permutations were encoded as ordered lists. Before that, we had implicitly encoded real and discrete variables as vectors and lists respectively. Other representations exist, for example those based on binary alphabets (see section 1.1.4). It is common and advisable to compare different representations for a given problem; one quality of EAs is that changing the representation only impacts on the initialization, crossover and mutation operators.
3. EAs move in \mathcal{S} through transitions in probabilities⁸. Therefore, it is not likely that two successive executions of an EA will yield the same result. Moreover, EAs need a minimum number of points calculations before the probabilities engine is started. This induces a numerical cost that may be unaffordable on certain applications. I estimate that there is no possible evolutionary optimization below 1000 analyses. An advantage of probabilistic transitions is that they make convergence to the global optima possible. Proofs of globality are asymptotic (i.e., for $t \rightarrow \infty$) and assume an upper bound on the selection pressure and a lower bound on the mutation strength and population size, cf. [27, 162].
4. EAs are zeroth-order optimizers, i.e., they only need the values of the optimization criteria and do not require gradients or Hessians. Today, zeroth-order optimizers have a large application field as most non-linear simulators do not provide gradients – a fortiori Hessians – because they are too complex to calculate or because they do not exist.

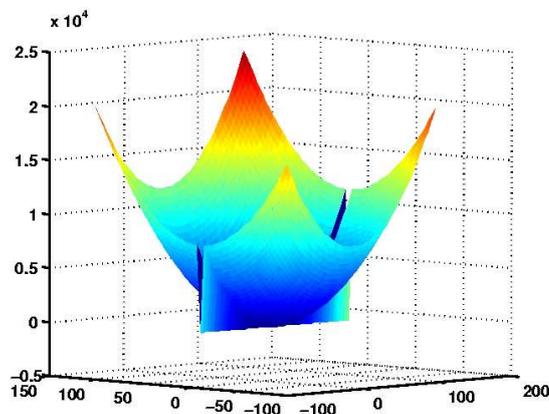


Figure 1.4: 2D view of the sphere-tunnel function, to be minimized. $f(x) = x^T x$ if $x \in \mathcal{T}$, $f(x) = -\exp(-\|x\|/10)$ otherwise, where the tunnel is defined as $\mathcal{T} = \{x \in \mathcal{S} \mid -0.5 \leq x_i \leq 0.5, i \in [2, n]\}$. Any point inside the tunnel has a better objective function than any point outside. A population enables the EA to progress outside the tunnel even if some of its members are inside, which eases progress towards $x^* = 0$.

⁸EAs can be seen as Markov chains in \mathcal{S}^μ where the selection, replacement and variation operators define the probabilistic transitions between the states, i.e., between the populations, [191].

1.1.3 Tuning an EA : the exploration-exploitation compromise

Every global optimizer compromises between the exploration of new regions in \mathcal{S} and the exploitation of already acquired knowledge about \mathcal{S} and f . Past knowledge corresponds to the identification of the best regions among those already explored. Exploration is necessary for global convergence, but it implies numerous f calculations. The archetypal exploratory search is the random sampling of \mathcal{S} with memory of the best point (Monte Carlo method). Exploitation leads to an accelerated convergence at the risk of missing global optima (premature convergence).

EA parameters tune the exploration-exploitation compromise. Increasing the mutation strength (p_m , eigenvalues of C) and the population size (μ) makes the EA more exploratory. Reciprocally, increasing the selection pressure (τ), the number of children (λ) and the crossover probability (p_c) moves the balance towards more exploitation. In EAs, an excess of exploitation induces premature convergence.

There is no general optimal EA tuning since the optimal tuning depends on the problem (f and g), the objective of the optimization (find a global optimum with ... accuracy after ... analyses and with ... probability) and even the distance to the optimum. As a general rule, the more difficult it is to optimize the function, the more exploration should be emphasized and vice versa.

Theoretical studies on the dependence of EA parameters on the problem dimension n have established the following rules:

- Minimum population size on binary alphabets, $\mu \sim \mathcal{O}(n)$ [62, 27].
- Mutation probability on binary alphabets, $p_m \sim \mathcal{O}(1/n)$ [9].
- Multi-Gaussian mutation strength, $\sigma_i^2 \sim \mathcal{O}(1/n)$, where σ_i^2 are the eigenvalues of C [168].

1.1.4 An historical standpoint

“Evolutionary algorithms” is the common name of various algorithms:

- Genetic algorithms (GAs) were invented and analyzed by J. Holland in 1975 [86] as a means of mimicking adaptation. They have been used as optimizers by K. DeJong [43], and popularized by D. Goldberg with his book [66]. One should read the article and recent book by K. DeJong on this topic [42, 96].
- Evolution strategies were devised by two engineering students in Berlin in 1965, I. Rechenberg [157] and H.-P. Schwefel whose reference book was written in 1981 [168]. Th. Bäck was one of the first to link evolution strategies and genetic algorithms in 1995 [11].
- Evolutionary programming dates from the early 1960’s in California with the work of L. J. Fogel [57]. After first being applied to the evolution of finite state machines, this technique was generalized for any representation by D. B. Fogel at the beginning of the 1980’s. D. B. Fogel also coined the term “Evolutionary Computation” in his book [56].
- Genetic programming has appeared more recently as a special type of genetic algorithm that evolves programs represented as trees [32]. It has been popularized by J. Koza (92) [114, 115, 116].
- Statistical optimization, or Estimation of Density Algorithms (EDA, see section 1.1.6), are even more recent and date back to S. Baluja’s and H. Mühlenbein’s research ([13], [140]).

These various EAs have existed for over 35 years now. In the meantime, they have had thousands of applications in every scientific field. They have mainly merged under the influence of Z. Michalewicz and his book [135] where he shows that these algorithms (which he still called “Evolution Programs” in 1992) can be applied to every representation. Today, EAs share over four international annual or bi-annual conferences (GECCO and CEC organized from the USA, PPSN and Evo* organized from Europe), and a French conference which is open to international participants, *Evolution Artificielle*. There are three specialized journals (*Evolutionary Computation*, *IEEE Trans. on Evolutionary Computing* and *Genetic Programming and Evolvable Machines*), but many contributions can also be found in *Complex Systems*, *Journal of Heuristics*, *BioSystems*, *Natural Computing*, *Journal of Global Optimization*, and *Theoretical Computer Science (C)*.

EA history can be split into two time periods. The first lasted until the middle of the 1990’s when researchers aimed at a universal optimizer, efficient in every problem. This was followed by the second contemporary period when the scientific community realized that EAs needed to match the problem at hand.

The Genetic Algorithm myth of a universal optimizer

Research for a universally well-performing optimizer lasted from 1970 to the middle of the 1990’s and hinged on three topics: representation, the respective roles of crossover and mutation, and the algorithm’s optimal tuning.

The algorithm embodying the best the ideas of this period, and the algorithm that received the most attention, was the genetic algorithm (GA, [86, 95, 66]). The classical GA takes the set of binary n -tuples, $\mathcal{S} \equiv \{0, 1\}^n$ as search space. As we shall soon see, this choice is substantiated by the schemata theory. The selection is made in proportion to the fitness F . The selection probability of $x^i \in \mathcal{P}^t$ is

$$p_s(x^i) = \frac{F(x^i)}{\sum_{x^j \in \mathcal{P}^t} F(x^j)}. \quad (1.9)$$

Two parents are taken with replacements from the current population \mathcal{P}^t with probability p_s . Their crossover generates two points that are added to the children’s population. The selection/crossover/mutation cycle is repeated $\mu/2$ times until the population \mathcal{P}^{t+1} is complete. In the elitist versions of GAs, the best individual is copied from the previous to the current population, replacing any of the individuals before evaluation. Most often, a one-point crossover is used in which one breaking position is taken at random in the parents’ chromosome and the chromosome’s parts are exchanged. For example,

$$\begin{array}{c} [0\uparrow 1 \ 1 \ 1] \\ [1\uparrow 1 \ 0 \ 0] \end{array} \Rightarrow \begin{array}{c} [0 \ 1 \ 0 \ 0] \\ [1 \ 1 \ 1 \ 1] \end{array}.$$

The GA’s mutation is that of discrete variables, which has already been described in section 1.1.2.

The efficiency of GAs has been analyzed with the schemata theorem [86]. A **schema** \mathcal{H} is a subspace of $\{0, 1\}^n$. It is usually written with the symbol * that stands for “0 or 1”. For example,

$$\mathcal{H} = [0 \ * \ 1 \ * \ 1] = \{[0 \ 0 \ 1 \ 0 \ 1], [0 \ 1 \ 1 \ 0 \ 1], [0 \ 1 \ 1 \ 1 \ 1], [0 \ 0 \ 1 \ 1 \ 1]\}.$$

The order of the schema, $o(\mathcal{H})$, is the number of fixed bits ($o(\mathcal{H}) = 3$ in the example). The schema length, $\ell(\mathcal{H})$, is the maximum distance between any fixed bits ($\ell(\mathcal{H}) = 4$ in the example). Let us use $N(\mathcal{H}, t)$ for the number of individuals of the population \mathcal{P}^t that belong to the schema \mathcal{H} , and $\bar{F}(\Omega)$ for the average performance

of individuals in Ω . The schemata theorem states that the expected number of individuals in \mathcal{P}^{t+1} that belong to \mathcal{H} grows as a function of the number of individuals of \mathcal{H} in \mathcal{P}^t and the performance advantage of \mathcal{H} . Vice versa, the number of \mathcal{H} representatives in \mathcal{P}^{t+1} decreases with the order and the length of the schema because it makes \mathcal{H} more sensitive to the effects of the mutation and crossover [86]:

$$E[N(\mathcal{H}, t+1)] \geq N(\mathcal{H}, t) \frac{\bar{F}(\mathcal{H} \cap \mathcal{P}^t)}{\bar{F}(\mathcal{P}^t)} \left[1 - p_c \frac{\ell(\mathcal{H})}{n-1} \right] [1 - p_m]^{o(\mathcal{H})}. \quad (1.10)$$

The term $\frac{\bar{F}(\mathcal{H} \cap \mathcal{P}^t)}{\bar{F}(\mathcal{P}^t)}$ describes the amplification effect of the roulette wheel. The term $\left[1 - p_c \frac{\ell(\mathcal{H})}{n-1} \right]$ captures the destructive effects of the one-point crossover, if the breaking point falls between the schema defining bits. For example, $\mathcal{H} = [1 * 0 *]$ is lost through a crossover of $[1 0 \uparrow 0 1]$ and $[0 1 \uparrow 1 1]$. Finally, the term $[1 - p_m]^{o(\mathcal{H})}$ is a model of the destructive effects of the mutation; if mutation occurs at one of the schema defining bits, the schema is lost for this individual.

This theorem has given rise to the idea of **building blocks**. Building blocks are short schemata ($o(\mathcal{H})$ and $\ell(\mathcal{H})$ small) that have above-average performance. For short schemata and a low mutation probability p_m , the last two terms in equation (1.10), i.e. the terms related to schemata extinction by crossover and mutation, can be neglected. If a schema has above average performance on a long-term basis, i.e.,

$$\frac{\bar{F}(\mathcal{H} \cap \mathcal{P}^{t+k})}{\bar{F}(\mathcal{P}^{t+k})} > 1 + \varepsilon \quad , \quad \varepsilon > 0 \quad , \quad k = 0 \dots T \quad (1.11)$$

then

$$E[N(\mathcal{H}, t+T)] > N(\mathcal{H}, t)(1 + \varepsilon)^T. \quad (1.12)$$

At the condition of a low mutation rate, GAs would therefore favor the rapid emergence of building blocks in the population. Crossover would then assemble these building blocks to lead to the optimum x^* . Moreover, each individual is part of a large number of schemata (2^n in a binary alphabet, * or its bit value at each tuple position) whose performance is estimated through the calculation of one $f(g)$. Such underlying parsimony in GAs has been called “implicit parallelism” [86].

The debates of the 1990’s

The above arguments that stem from the schemata theory provided hope until the 1990’s that it would be possible to obtain algorithms which would be efficient on average in all problems. However, contradictions between GA theory and empirical results were rapidly been observed, which spawned debates on representation and the relative importance of crossover and mutation in EAs.

The coding of the x ’s

An often cited argument [86, 66] is that binary codings maximize the schemata proportion sampled by one individual, hence it maximizes its implicit parallelism. It has since then been proved by Antonisse that this statement is not correct⁹ [3].

⁹The number of schemata to which a given individual belongs is 2^n . In a cardinality A alphabet, this number of schemata is $(A+1)^n$; the schemata fraction sampled by one individual would be maximum when $A=2$. The mistake here is that there are in fact many * symbols in an alphabet whose cardinality is $A > 2$ if one wants to count all subspaces. For example, for $A=3$, one has $*_{0,1}$, $*_{0,2}$, $*_{1,2}$, $*_{0,1,2}$. Moreover, the comparison between cardinalities must be performed at a constant search space size, $2^{n_2} = A^{n_A}$. With these two corrections, individuals coded with alphabets of cardinality $A > 2$ sample a larger proportion of the total number of subspaces (schemata) than binary coded individuals do.

It seems that binary coding proponents have been intuitively attracted by the idea that an alphabet of smallest cardinality makes a maximum decomposition¹⁰ of the information in x . Since there is no informative entity smaller than a bit, a binary alphabet reveals all possible building blocks in a problem. The difficulty is to find, among the numerous possible binary codings ($2^n!$), the one that favors the most compact high performance building blocks. To this end, projects that aim at optimizing the coding by the GA itself have been carried out. The meaning of the allele is coded on the gene and manipulated by the GA. Examples of these are inversion [86] and the messy GAs [63, 64].

Generally, “natural” representations, that match the problem at hand, have turned out to be more efficient than binary codings. Building blocks can often be inferred from the problem statement. It is preferable to handle them directly using an ad hoc coding. Describing solutions on a smaller scale (binary) will often induce a loss of problem understanding. In this way, evolution strategies (ES, [168, 11, 82]) directly handle vectors of real numbers and have turned out to be more efficient for continuous variables optimization than EAs working with binary codings. In [102, 103, 78], many representations have been compared for the topological optimization of solid structures. Binary codings are not as efficient as, for example, codings based on Voronoï cells because they do not propagate topological features well during crossover. Moreover, binary codings cannot locally refine the shape description for a fixed n , unlike Voronoï cells.

Crossover versus mutation

Another background discussion between 1980 and 1995 dealt with the relative importance of crossover and mutation in EAs. The schemata theorem (equation (1.10)) describes the building blocks’ growth assuming a marginal use of mutation, whose purpose is only to prevent premature convergence. By recombining building blocks, crossover is considered as the main search operator¹¹. On this basis, genetic algorithms use low mutation probabilities around $p_m = 0.01$ (in $\mathcal{O}(1/n)$) and high crossover probabilities, $0.6 \leq p_c \leq 1$.

On the contrary, evolution strategies and evolutionary programming [57, 55] mainly rely on mutation to find x^* . The first versions of ES and evolutionary programming in the 1960’s and 1970’s did not use crossover at all. Similarly to GAs that look at mutation as a repair operator (re-injecting in the population building blocks that have been lost by sampling errors), ES consider crossover as a repair operator: it extracts similarities in high performing individuals of the population to compensate for some of the mutation induced noise, thus increasing convergence speed [19, 20].

The end of the myth

Around 1995, feedback from numerical experiments and some elements of analysis, the No Free Lunch theorem (cf. section 0.3), showed that a universal optimizer cannot exist.

Experiments feedback

The EA development in the 1990’s fostered a large number of studies and applications whose results often contradicted, at least partially, the schemata theory. For example, in [Le Riche and Haftka, J1993] and [Le Riche and Haftka,

¹⁰Maximum decomposition means here the non-redundant coding that yields the largest number of components, n . If the coding is seen as a map $\psi : \mathcal{C} \rightarrow \mathcal{S}$, where \mathcal{C} is the space of coded individuals, the coding is non-redundant if ψ is bijective.

¹¹It should be noted that this constructive action of the crossover operator is not accounted for in the schemata theorem

J1995], an empirical optimization of a GA parameters for designing composite laminates concluded that small population sizes ($\mu \approx 10$) and large mutation probabilities yielded the best strategy. Progressively, the idea of matching the EA and the problem emerged.

Critics of the schemata

At the same time, it was becoming clear that two hypotheses were missing in the schemata theory. Firstly, the schema performance estimate in the current population, $\bar{F}(\mathcal{H} \cap \mathcal{P}^l)$, is a rough approximation of the schema performance, $\bar{F}(\mathcal{H})$. Its accuracy decreases when the spread of $F(x)$, $x \in \mathcal{H}$, increases. The schemata theorem is therefore valid only for schemata with a low performance variation. This discussion is continued in section 1.1.5. Secondly, in difficult, non-linear problems, good schemata typically do not contain the global optimum x^* . The assumption that recombining short low order schemata leads to the optimum is close to a linearity hypothesis. Difficult problems are often deceptive [194]: for binary alphabets, a function is deceptive at the order k if all the order k schemata that have best average performance do not contain x^* . For example, a 3 bits problem such that $\bar{F}([0 * *]) > \bar{F}([1 * *])$, $\bar{F}([* 0 *]) > \bar{F}([* 1 *])$, $\bar{F}([* * 0]) > \bar{F}([* * 1])$, and $x^* = [1 1 1]$ is deceptive at the first order.

1.1.5 Tools for designing EAs

Once the idea of adapting the algorithm to the problem emerged, it became necessary to measure how an EA component (the coding, the initialization procedure and the variation operators) fits the function f (and g if there are constraints). Of course, the simplest way to measure the appropriateness of the implementation is to perform a series of runs and decide a posteriori. However, a priori analysis is more likely to help an understanding of why an optimization mechanism fits a given function. We present hereafter two types of a priori performance analysis¹².

The EA-performance relation

An evolutionary algorithm is made of an x coding, an initialization strategy for \mathcal{P}^0 and variation operators. It is instructive to confront the distribution of the points generated by each of these components with the distribution of their performances.

A simple tool to study a coding or an initialization implementation is to plot $f(x)$ as a function of the distance $d(x, \hat{x}^*)$ between x and the best sampled point \hat{x}^* using this implementation [98, 99]. Without further a priori knowledge, one should favor a coding that yields a good covering of the (f, d) plane. An example is shown in FIG. 1.5; the optimization aims at finding a tuple s in $\{0, 1\}^{900}$. The fitness function $F(x)$ is the number of digits that are common to s and x . There are 100 0's in s . Two initialization procedures are compared in figure: on the left, each bit is independently set to 0 or 1 with a probability of 1/2. The total number of 1's follows a binomial distribution and for $n = 900$, 99% of the generated tuples have between 411 and 489 1's. On the right, the probability of an apparition of a 1 at each bit is taken from $U[0, 1]$ independently for each individual. The plot shows that, for this problem, the second initialization procedure is more appropriate since a larger (d, f) diversity is observed.

There are measures that summarize (d, f) plots, in particular the fitness-distance

¹²There is a third approach based on estimating the **epistasis**, i.e., the distance between a function and its closest decomposable function, $f_d(x) = \sum_{i=1}^n f_i(x_i)$, [159]. The idea here is that the more the variables are coupled, the more difficult optimization is. In fact, it depends on the ability of the search operators to identify and account for these couplings.

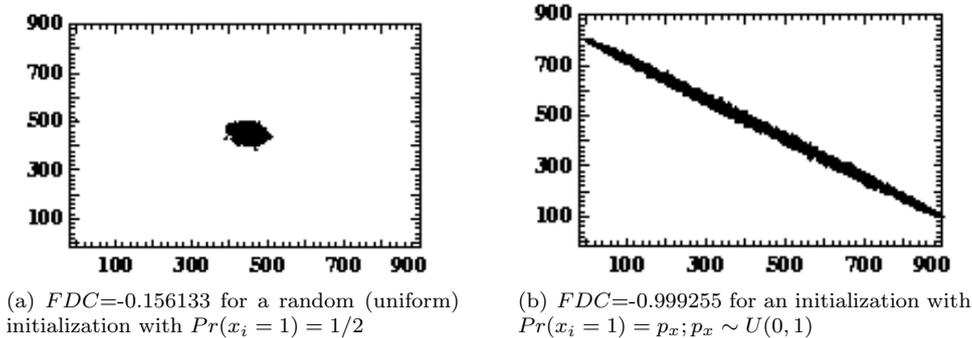


Figure 1.5: Performance vs distance to the best individual of a 6000 points sample for two initialization procedures (from [99]). There is a better covering with the individual based initialization (right), which will induce an easier subsequent convergence to x^* . The empirical FDC close to -1 indicates an easy problem.

correlation or FDC , [94]),

$$FDC = \frac{1/N \sum_{i=1}^N (f(x^i) - \bar{f})(d(x^i, \hat{x}^*) - \bar{d})}{\sigma_d \sigma_f} \quad (1.13)$$

where N is the sample size, \bar{f} , σ_f , \bar{d} and σ_d are the averages and standard deviations of f and d in the sample, respectively. The *OneMax* function, which calculates the number of 1's in x , has an FDC equal to 1. This corresponds to an easy problem, because every fitness improvement is associated with a step towards x^* . The individual-based initialization on the right of FIG. 1.5, with an estimated FDC close to -1, characterizes an easy problem. Note that the FDC does not provide reliable information on a particular optimization difficulty for an EA except for extreme values (1 and -1) that indicate an easy problem.

Similarly, the study of the performance-distance distribution can be applied to the choice of couples (coding, crossover) and (coding, mutation). In these cases, one should look at the relation between the performance of the best parent and the performance of its best offspring. Several averages that sum up this relation are analyzed in [98] but, like the FDC , the complete plot carries more information.

The formae variance

Formae are a generalization of the schemata of section 1.1.4. Formae are equivalence classes generated by the equivalence relation “has the same features as” applied to points of \mathcal{S} [152]. If the considered feature is the bits values at certain locations of a binary n -tuple, then formae and schemata are synonymous. However, formae also apply to variable length codings (used in non-parametric optimization, an example of which can be found in section 2.2). The features are elementary entities carried by the coding and recombined by the crossover. Out of this generalization, the concepts of formae order and performance are similar to those of schemata already seen in section 1.1.4.

If one accepts the conjecture that EAs work by recombining elementary features carried by good individuals (the building blocks), it is logical that the efficiency of EAs depends on the possibility of identifying these features from a limited group of individuals (the population). Let \mathcal{H} be one of the formae that contains the global optimum x^* . It can be expected that the EA optimization is made easier if the average performance of \mathcal{H} is better than that of other same order formae that do

not contain x^* . But for \mathcal{H} to be detected and propagated, it is also necessary that the variance of its performance be small. Therefore, a principle for choosing an EA coding is to minimize the associated formae variance. An example can be found in [153] where the formae variance of different codings for the traveling salesman problem are compared and a correlation between low variance and EA efficiency is observed.

1.1.6 Parametric evolutionary optimizers: EDAs and ES

We now turn to standard instances of parametric EAs that optimize a finite number of variables. An example of a non-parametric evolutionary optimizer in the design of composite laminates will be given later in the text (section 2.2).

At each iteration t , all the evolutionary optimizers define a probability distribution p^t that will be sampled to produce the λ new points. Algorithms based on crossover operators define these distributions implicitly: p^t is not calculated and the offspring are produced with the aid of the variation operators. When expertise about the optimized functions is available, as in composite laminate design ([Le Riche and Haftka, J1993], [Le Riche and Haftka, J1995]), working directly on the variation operators and the coding may help better accounting for background knowledge. Nevertheless, general parametric EAs benefit, through the explicit use of p^t , from a better formalization which, in turn, permits better control.

Three types of parametric EAs will now be summarized: the estimation of distribution algorithms (EDAs), the auto-adaptive ES and the adaptive ES. A shared feature of all these algorithms is that one learns at each iteration a probability distribution parameterized by a vector θ , $p^t(x) \equiv p(x|\theta^t)$ (or $p^t(x) \equiv p(x|\theta^t(x))$ for auto-adaptive ES). The difference between these algorithms resides in the distribution updating. EDAs learn p^t by maximizing the likelihood of good observed points at t . Adaptive ES rely on the dynamics of good observed points over the iterations. Auto-adaptive ES optimize the parameters θ at the same time as the variables x .

Estimation of distribution algorithms

Although EDAs are not regarded today as efficient evolutionary algorithms [106], they are of great theoretical interest [120, 61]. EDAs illustrate the difference between evolutionary optimization and other global optimization approaches such as stochastic optimization and Bayesian optimization. Comparing EDAs and ES is also instructive.

EDAs search the space \mathcal{S} by sampling the distribution p^t and updating it using the best observed points. EDAs were first introduced in discrete spaces, with the distribution p^t replacing the variation operators (“Population Based Incremental Learning” – or PBIL – algorithm in [13], “Univariate Marginal Distribution Algorithm” – or UMDA – in [139]). An EDA flowchart has the following structure:

1. Initialize the distribution $p^0(x)$, $t = 0$.
2. Create λ points by sampling $p^t(x)$.
3. Evaluate F for the λ points.
4. Select the μ best points ($\mu < \lambda$).
5. Update p^{t+1} using the μ best points^a and $p^t(x)$.
6. If continue, $t \leftarrow t + 1$, go to 2.

^aIn some variants, the updating also uses the *worst* points of the current population [14, 172].

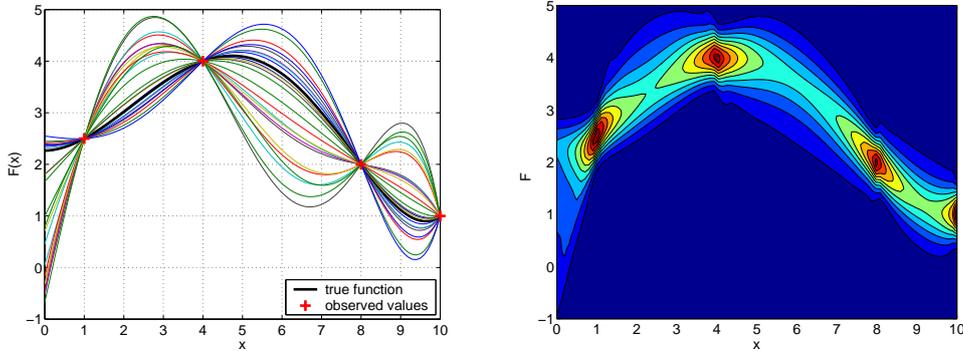
The distribution p^t is a representation of what was learned about F during the evolution [33]; the individuals $(x, F(x))$ are a sample of the function (see FIG. 1.6),

and assumptions made on F spatial correlation provide bounds on how $F(x)$ is likely to change as a function of the distance between x and known points.

If \hat{F}^* is a performance value beyond which a point is considered as good (for example, the best known performance at time t), the distribution p^t can be interpreted as

$$p^t(x) = P(x|F(x) > \hat{F}^*) = \frac{P(F(x) > \hat{F}^*|x)P(x)}{P(F(x) > \hat{F}^*)} \quad (1.14)$$

where $P(x)$ is a prior distribution. For a uniform prior distribution, $p^t(x) \propto P(F(x) > \hat{F}^*|x)$. Equation (1.14) shows the relation between EDAs and stochastic optimizers. Stochastic optimizers process a probabilistic representation of $F(x)$. An example of a stochastic optimizer is given in [Ginsbourger et al., C2007] and summed up in section 1.4. Other examples can be found in [164, 93]. In [93], the point taken at iteration $t + 1$ is chosen so as to maximize the expected improvement $E[\max(0, F(x) - \hat{F}^*)]$.



(a) Functions compatible with the observations.

(b) Contours of $P(F(x)|F(x_i) = f(x_i))$.

Figure 1.6: Representation of the uncertainties on non-sampled regions of \mathcal{S} as a random process $P(F(x)|F(x_i) = f(x_i))$ (from [69]).

EDAs can also be regarded as a Bayesian method. If one assumes that the selection defines a probability of selection on \mathcal{S} , p_s , and in the *idealization of an infinite population*, the flowchart steps 2 and 4 lead to

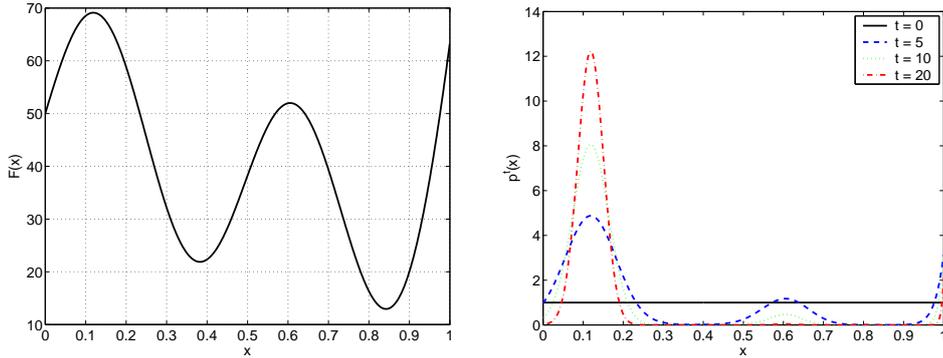
$$p^{t+1}(x) \propto p_s^t(x).p^t(x) . \quad (1.15)$$

FIG. 1.7 illustrates the convergence of such an idealized EDA (p^t is not parameterized and $f(x)$ is supposed to be known everywhere in \mathcal{S}). In practice, the densities p^t are parameterized through θ , $p^t(x) \equiv p(x|\theta^t)$, and the populations have a finite size, which brings about two difficulties.

First of all, in order to locate the optima efficiently, the EDA (its distribution p^t) must be able to describe the couplings between the variables. In FIG. 1.8 for example, the variables x_1 and x_2 are independent, so that $p(x_1, x_2) = p_1(x_1)p_2(x_2)$ cannot have its highest probability region coincide with x^* . In order to improve the description of variables couplings, early EDAs that had independent variables (PBIL [13], UMDA [139]) were supplemented with algorithms with more complex distributions (e.g., FDA [140] or BOA [148] which relies on Bayesian networks).

Secondly, estimating distribution requires a number of data ((x, F) couples) that increases with the $p(\cdot|\theta)$ distribution flexibility¹³. In EDAs, $p(\cdot|\theta)$ is inferred from the μ best points among λ and a prior distribution. The real cost of each iteration

¹³The flexibility, or complexity, of a parameterized functional reflects the independence of the



(a) Example of a 1D function to maximize with three local maxima $x = 0.11$, $x = 0.61$, and $x^* = 1.00$ (global optimum). $F(x) = 50 - 60x^2 + 50x^5 + 20x^{10} + 20\sin(40x/\pi)$

(b) Convergence of the distribution $p(x)$ to the Dirac distribution $\delta(x^*)$. The idealized EDA process is $p^{t+1}(x) \propto p_s^t(x)p^t(x) \propto f(x)p^t(x)$

Figure 1.7: Example of an idealized EDA convergence in 1D (from [69])

is λ analyses. If μ is too small with respect to how $p(\cdot|\theta)$ is flexible, $p(\cdot|\theta^t)$ will strongly vary with the sampled points and the algorithm's behavior will be erratic.

EDAs offer another illustration of the bias-variance compromise known in regression [48, 188]. A balance must be found between p^t 's complexity and the populations size. At one extreme, some EDAs rely on simple distributions: for example, in continuous spaces, PBIL_c [171] uses a multi-Gaussian density without correlations. The Double Density Optimization Algorithm discussed in section 2.2 and introduced in [Grosset et al., J2006] is an attempt to keep the distribution structure simple, while enriching its abilities to capture variable interactions by working in two joined spaces. At the other extreme, EDAs that opt for rich dependencies between the variables must, in addition to maximizing the likelihood of the good points, resort to regularization techniques. The two main regularization techniques are the minimization of a regularized risk and cross-validation (cf. [48]). The IDEA algorithm [23] represents joint continuous variable densities by products of conditional densities. When estimating p^t , IDEA maximizes the likelihood of good points penalized (regularized) by the number of edges in the conditional probabilities graph.

To stabilize EDAs, it is common to express the new θ as a linear combination of the last estimate and its previous value,

$$p^t(x) = p(x|(1 - \alpha)\theta^t + \alpha\theta^{t-1}) \quad , \quad \alpha \in [0, 1] \quad . \quad (1.16)$$

As already stated, EDAs are not regarded today as efficient optimizers. They are based on learning, at each iteration, the probability of a point being good, and this in the entire search space \mathcal{S} . The underlying assumption is that a large proportion of \mathcal{S} has been sampled, which is a costly, or even unrealistic, hypothesis. Furthermore, the learned density is a static picture of what is already known from \mathcal{S} at time t . It does not encourage the exploration of new regions because these lie in the distribution tails and have low probabilities of being visited. Efficient optimizers, such as adaptive ES, do not attempt to infer a distribution on F at every point of \mathcal{S} . Instead, they exploit the population dynamics over many iterations and learn favorable moves. They can hasten their steps in the directions associated with the greatest progress.

values it can take knowing some of its values; for example, two points uniquely define a linear function in 1D. Formally, the complexity of a class of functions can be measured by its VC (Vapnik-Chervonenkis) dimension. The interested reader is referred to [188, 170].

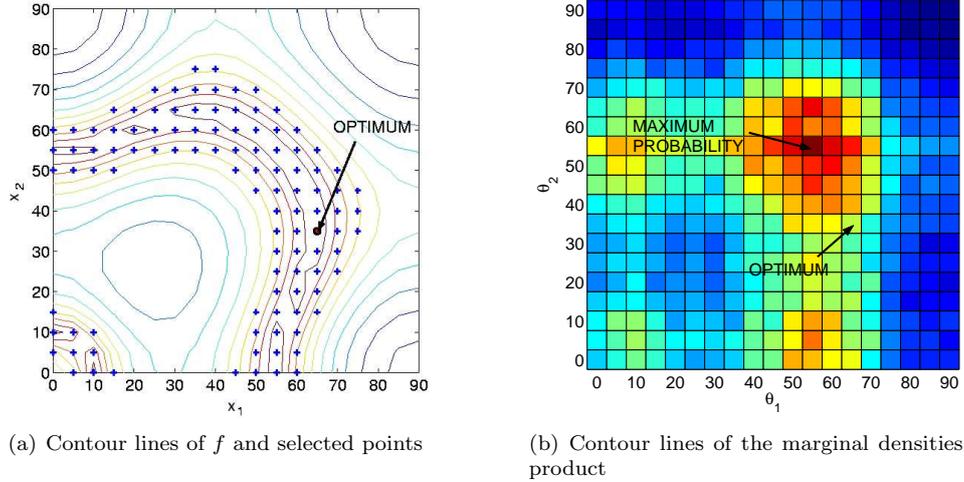


Figure 1.8: Selected points (a) and contour lines of the marginal densities (b) for $(x_1, x_2) \in \{0, 5, 10, \dots, 85, 90\}^2$ on a composite laminate designed for a vibration problem (from [70]). When the selected points distribution is represented by independent variables (the product of the marginal densities, $p(x_1, x_2) = p_1(x_1)p_2(x_2)$), high probabilities and high performance regions may not coincide.

We believe that EDAs should be compared to EAs with crossover but without mutation. The equivalence between the uniform crossover and the PBIL and UMDA distributions makes this point clear (cf. section 1.1.2). $p^t(x)$ and the crossover are two ways of sampling new points from the empirical distribution underlying the parents. Furthermore, we have found experimentally that mutation improved the efficiency of EDAs (cf. section 2.2.2).

Evolution strategies

Evolution strategies (ES) are EAs for continuous parametric optimization ($S = \mathbb{R}^n$, or a subset of \mathbb{R}^n).

The history of evolution strategies illustrates well the different principles for tuning the parameters of evolutionary parameters. More specifically, the tuning of the Gaussian mutation, which is central to ES, has gone through all the stages, starting from fixed to adaptive, then auto-adaptive, before returning to adaptive.

Gaussian mutation

The main operator in ES is the Gaussian mutation, which adds a centered Gaussian noise to each individual. A way to substantiate the Gaussian framework for optimization is to note that among all distributions with a given covariance, the Gaussian law has the highest entropy, and therefore corresponds to injecting the least a priori knowledge. This is appropriate when no a priori information is known. In n dimensions, the multi-Gaussian distribution $\mathcal{N}(m, C)$ has an average m and an $n \times n$ positive definite covariance matrix C . The associated probability density function is

$$\Phi(X) = \frac{\exp(-\frac{1}{2}(X - m)^t C^{-1}(X - m))}{\sqrt{(2\pi)^n |C|}}$$

where $|C|$ is the determinant of C .

The Gaussian mutation of a vector $x \in \mathbb{R}^n$ can be rewritten by separating a scale factor σ (also called step) from the principal directions of the mutation

$$x \rightarrow x + \sigma \mathcal{N}(0, C)$$

The simplest example is when C is the identity matrix: the variables are mutated independently from each other by a Gaussian noise of variance σ^2 .

Tuning ES parameters amounts to looking for the best values of the step and the covariance matrix during the evolution. The first experiments dealt with the isotropic case in which the only degree of freedom is the step σ .

Step adaptation

The Gaussian mutation step sets the scale at which the optimizer “sees” the objective function. As a simple illustration, let us consider a (1+1)-ES (see section 1.1.2) on $f = x^2$ in one dimension. Next, the average distance between two successive individuals is proportional to σ , with two consequences: firstly, if the starting point is d_0 , an average of d_0/σ iterations will be necessary to reach a neighborhood of 0. Secondly, the maximum convergence accuracy that one should expect is also proportional to σ . Such arguments lead to the idea of adapting the step, for example in proportion to the distance to the optimum. The *progress rate* theory by Schwefel [168] and later by Beyer [20] detailed these ideas. Recently, Auger [7] provided formal proof of the convergence of this type of adaptive algorithm.

Of course, in practice, one does not know the distance to the optimum, so adaptation must rely on other information. The *success rate*, i.e., the rate at which mutation has improved an individual, indirectly gives information about the current step. This observation by Rechenberg [158] has produced the “1/5 rule”, the first known example of step adaptation in EAs. The 1/5 rule increases the step size when the success rate is larger than 0.2 (1/5) and vice versa. Indeed, when the step tends toward 0, the success rate tends toward 0.5 (think of a zoom making contour lines linear). Too small a step (in terms of the speed at which one approaches the optimum) yields too high a success rate. The 1/5 rule has been proved for some simple functions (sphere, corridor), but it is put at fault by any function that does not allow a 1/5 success rate when the step becomes small (such as contour lines making a sharp V). Although the 1/5 rule is no longer used today, it represented an important phase in the development of ES.

Auto-adaptation and ES

The next important phase for ES was the invention of the *auto-adaptive* mutation: the mutation parameters (the step σ and even the covariance matrix C) are attached to each individual and are themselves subject to ... mutations. An individual mutation starts with a mutation of its mutation parameters, followed by a mutation of x with the new parameters. The underlying idea is that selection, although only taking the individual’s performance $f(x)$ into account, can also select the individuals that carry the best mutation parameters because they are indirectly expressed in the performance of the offsprings. Let us take two individuals with same performance, of which one has better mutation parameters than the other. The individual with the better parameters will produce, on average, better offsprings, so that its parameters will propagate in the population over several generations, to the detriment of the other individual’s settings. It has hence been said that auto-adaptation tuned mutation parameters “for free”. Auto-adaptation has long been considered as the state-of-the-art in ES [11].

We can distinguish three versions, according to which parameters are used: the *isotropic* mutation is exclusively based on one step, σ , per individual, the covariance matrix being equal to the identity; the *non isotropic* mutation assigns a vector of

steps $(\sigma_1, \dots, \sigma_n)$ to each individual and the covariance matrix is diagonal with σ_i^2 at the i -th position. The variables are, again, mutated independently from each other but with different variances; finally, the *correlated* mutation attaches a full covariance matrix to each individual. More details can be found in [168, 12, 54].

Let us return to the discussion on what “good” mutation parameters are. This has been addressed for a single step parameter in one dimension. Similar arguments obtained by replacing the sphere function $(\sum x_i^2 \equiv x^t x)$ by an ellipsoid $(\frac{1}{2} x^t H x, H \text{ positive definite})$ show that the covariance matrix should be proportional to the inverse of the Hessian, H^{-1} . However, it has been proved that the auto-adaptation, while it finally learns the correct step [11, 41], does not learn the inverse of the Hessian [6].

CMA-ES: back to a deterministic adaptation

Even when auto-adaptation works, it tends to be slow. It takes several generations to learn a correct step size. This observation led Hansen and Ostermeier to propose a deterministic adaptation procedure first for the step size [83], and then for the complete Gaussian mutation covariance matrix [81]. The intuition is that if two successive mutations have been successful in a given direction, one should probably hasten the step in this direction. The complete algorithm, called CMA-ES (*Covariance Matrix Adaptation*) has been proposed and well analyzed – with carefully set default parameters – in [82]. Contrary to EDAs that learn the distribution of good individuals, CMA-ES learns the distribution (assuming it is Gaussian) of successful steps.

Today, CMA-ES is the state-of-the-art evolutionary optimizer for continuous parametric problems. It won the 2005 contest on evolutionary optimization at the CEC congress [181], and later was even found to be competitive in terms of convergence speed with the BFGS mathematical programming algorithm on unimodal twice differentiable function in “real life” conditions, i.e., with gradients estimated by finite differences [165].

Conclusions on ES

This summary of the history of evolution strategies illustrates the experience in tuning EAs parameters acquired during the last twenty years. It seems clear today that there are no static optimal EA parameters, as was seen for the mutation operator. The 1/5 rule often fails because it is based on indirect information, the success rate. Auto-adaptive methods are more robust, but they remain slower than modern adaptive methods, i.e., CMA-ES, that directly process the successful steps.

1.2 Handling non-linear inequality constraints

1.2.1 Context

Most engineering optimization problems have inequality constraints, $g(x) \leq 0$, which must be satisfied at the optimum. These constraints describe bounds on the variables, limits on a structure displacements, load carrying limitations due to strength or buckling failure, minimum performance in a multi-criteria context, compatibilities between neighboring components, Variables bounds and, more generally, linear constraints are efficiently handled by projection techniques. We do not discuss them here and focus on non-linearly constrained problems. The primal constrained optimization problem (P) is,

$$(P) \quad \begin{cases} \min_{x \in S} f(x), \\ \text{such that } g(x) \leq 0. \end{cases} \quad (1.17)$$

Assumptions on the objective function f , the constraints g , the search space S are the same as for unconstrained problems (equation (1.1)). We further assume that there is at least one feasible point in S , i.e., a point x such that $g(x) \leq 0$.

Four types of methods for handling constraints exist: the transformation of (P) into an unconstrained problem using penalty functions (differentiable penalty functions [76] section 5.7, non differentiable penalties [87], and for EAs [107, 149, 160, 178, 16, 73, 17]), the projection of infeasible solutions onto the feasible domain (e.g., projected gradient), constraints representation building in the course of the search (e.g., sequential quadratic programming) and, for EAs, the coevolution of populations which together solve the constrained optimization problem (e.g., [147]). These approaches are related and have been coupled, e.g., coevolution and penalty methods in [Le Riche et al., C1995-1] and [182]. A review on constraints handling in evolutionary optimization can be found in [136].

Penalty functions are a general approach to handling constraints. The choice of the penalty function is critical to the success and the efficiency of optimizers, in particular EAs. Let us first consider a *static penalty* function,

$$\bar{L}(x, p) = f(x) + p \max(0, g(x)) \quad (1.18)$$

where p is a constant vector of penalty parameters. If p is too small, the solution to (1.18) is infeasible, $g(\arg \min f_p(x)) > 0$. Vice versa, a too large p makes it impossible for the search to go through the infeasible domain and this absence of “short cuts” may be fatal to the global exploration of the space. A tempting alternative is to start the search with small penalties, so that a global exploration of the domain can occur, and then increase the penalty to force convergence to the feasible domain. However, to be efficient, such a *dynamic penalty* is difficult to tune because it depends both on the problem and on the search state. *Adaptive penalties* adjust the amount of penalty depending on the search history. The adaptive penalty functions which are the best understood and the most important in practice (for the sensitivity of the optimum to the constraints) are based on duality and the related (classical, augmented and extended) Lagrangians concepts.

All the methods presented next use penalty functions. Part 1.2.2 is specific to EAs as it relies on the coevolution of subpopulations who are differently penalized. It is followed, in part 1.2.3, by a study on adaptive penalties and duality which, although often cast in the context of EAs, is applicable to any zeroth-order optimizer such as GBNM (cf. section 1.3).

1.2.2 The segregated genetic algorithm

The SeGREGated Genetic Algorithm (SGGA) is a coevolutionary algorithm for optimizing constrained problems. SGGA was introduced in [Le Riche et al., C1995-1]. In the SGGA, constraints are handled by penalty functions. The unconstrained penalty function, which is minimized instead of solving (P) , is generically written $F_p(x)$, where p is a vector of penalty parameters. The static penalty function of equation (1.18) is an example of $F_p(x)$. As it has been argued in the previous paragraph, it is preferable not to penalize too much infeasible points in order not to forbid short cuts through the infeasible domain, yet it is necessary to sufficiently penalize in order to ultimately converge to a feasible point. This rule has long been known in the evolutionary calculation community as the *minimal penalty rule* [38, 160, 178]. Section 1.2.3 will investigate the mathematical calculation of minimal penalties using duality theory. On the contrary, the SGGA works with fixed, a priori given penalty parameters, but it aims at desensitizing the search to the choice of these parameters.

The basic idea of the SGGA is to use two penalty parameter values (say p_1 and p_2) instead of one. The two values are associated with two groups of solutions that

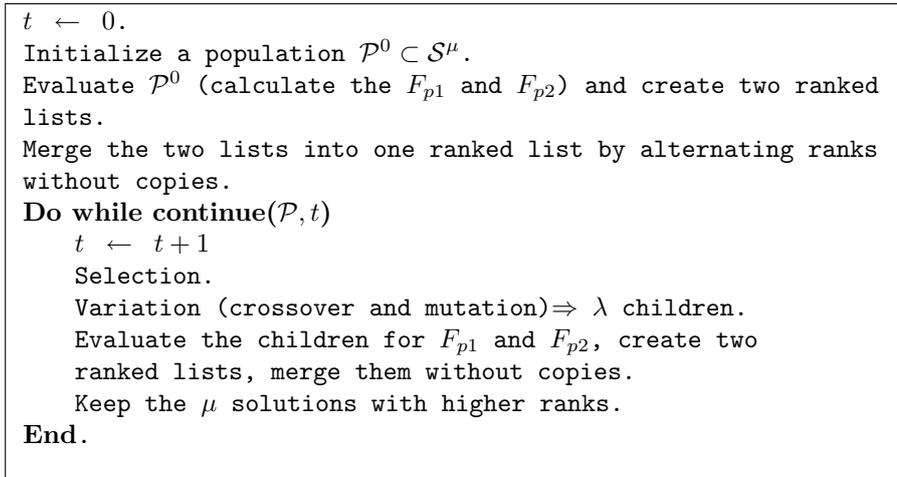


Figure 1.9: Flowchart of a segregated genetic algorithm (from [Le Riche et al., C1995-1]).

have different levels of constraint satisfaction. Each of the groups corresponds to the best performing individuals with respect to one penalty parameter. The two groups interbreed (through crossover), but they are segregated in terms of performance evaluation. Two advantages are expected. Firstly, because the penalty parameters are different, the two groups will have distinct trajectories in the search space. Because the two groups interbreed, they can help each other out of local optima. The SGGA is thus expected to be more robust than an equivalent EA. Secondly, in constrained optimization problems, the optimum is often¹⁴ located at the boundary between feasible and infeasible domains. If one selects one penalty parameter large (say p_1) and the other one small (say p_2), one can achieve simultaneous convergence from both the feasible and the infeasible sides. The global optimum will then be encircled by the two groups of solutions, and since the points are mixed by crossover, the optimum should be located faster.

For example, in structural optimization, one usually seeks to minimize the weight of a structure. The “ p_1 group” typically contains heavy designs that do not fail, while the “ p_2 group” contains light designs that fail. The optimum design, which is a compromise between weight and constraint satisfaction is located somewhere between the p_1 and p_2 groups.

FIG. 1.9 shows the flowchart of the SGGA. It is a $(\mu + \lambda)$ -ES with the exception of the evaluation procedure which ranks the points in two lists according to their F_{p1} and F_{p2} values, and then merges them in a ranked list by taking the best of list 1, then the best of list 2 which was not already selected, then the second best of list 1

The SGGA was tested for the optimum design of composite laminates (cf. formulation in equation (2.1)). The objective was to minimize laminate thickness such that the laminate would not fail, neither from buckling, nor from insufficient strength. The penalty function was geometric, p acting as a power of the constraint (cf. [Le Riche et al., C1995-1]). The reliability¹⁵ of the SGGA was estimated from

¹⁴The solution of a constrained optimization problem does not have to be at the constraint boundary. The Kuhn and Tucker multipliers can be null at the optimum, cf. FIG. 1.10 for example. In practice however, the optimum will most often be in the vicinity of a constraint boundary.

¹⁵The reliability of the SGGA is defined as the probability of finding an optimal weight design with a critical load within 10% of the global optimum.

3000 independent searches of 6000 analyses each. The laminate was subjected to three load cases and failure was calculated for the most critical load case. The experiments confirmed that the SGGA is more reliable and less sensitive to the choice of p than an equivalent classical EA. Furthermore, it was observed that SGGA converged to the optimum with an average population weight inferior to the average population weight of a classical EA.

A practical difficulty left unresolved by the SGGA is that it does not specify how to choose p_1 and p_2 . An improvement of the SGGA was later proposed in [17], where the idea of forcing a balance between feasible and infeasible points in the population remained, but only one penalty parameter was used. And, precisely, it was adjusted so as to preserve a minimum proportion of feasible and infeasible individuals in the population: Let $\Theta_{feasible}^t$ denote the proportion of individuals satisfying the constraints at generation t , and Θ_{inf} and Θ_{sup} two user set bounds on $\Theta_{feasible}^t$. Low values of the penalty parameters favor infeasible individuals, and vice versa, which suggests the updating rule,

$$p_{t+1} = \begin{cases} \beta \cdot p_t & \text{if } \Theta_{feasible}^t < \Theta_{inf} \\ (1/\beta) \cdot p_t & \text{if } \Theta_{feasible}^t > \Theta_{sup} \\ p_t & \text{otherwise} \end{cases}$$

where $\beta > 1$. Empirical default values are $\beta = 1.1$, $\Theta_{inf} = 0.4$, et $\Theta_{sup} = 0.8$ ([17] and [Le Riche et al., J2007]). Note that variations of p are not monotonous, so that there is no guarantee that the best population individual is feasible. It can even happen that no feasible point is in the current population. However, in this case, the steady increase in p will favor the emergence of feasible points.

1.2.3 Non-differentiable penalty functions and duality

Alternatively, duality theory can be called upon to build adaptive penalty functions.

Elements of duality

We now review some duality principles which will be useful in the rest of the dissertation. For the sake of simplicity, the number of constraints is limited to one. It should be noted that problems having $m > 1$ constraints can always be set in terms of a single constraint by taking the most critical constraint,

$$\begin{cases} \min_{x \in S} f(x), \\ \text{such that } g(x) = \max_{i=1,m} \{g_i(x)\} \leq 0. \end{cases} \quad (1.19)$$

The set of solutions of (P) is denoted X^* , x^* is any element of X^* . The Lagrangian formulation (P_λ) of the primal problem is,

$$(P_\lambda) \quad \min_{x \in S} L(x, \lambda), \quad (1.20)$$

where,

$$L(x, \lambda) = f(x) + \lambda g(x). \quad (1.21)$$

λ is a Lagrange multiplier. The set of solutions of (P_λ) is X_λ . We further assume that for each $\lambda \geq 0$, there exists at least a bounded solution $x_\lambda \in X_\lambda$. This assumption is fulfilled, for example, if f and g are continuous (Weierstrass theorem, [138]).

The dual function is,

$$\phi(\lambda) = \min_{x \in S} L(x, \lambda), \quad (1.22)$$

and the dual problem is stated as,

$$(D) \quad \max_{\lambda \geq 0} \phi(\lambda). \quad (1.23)$$

The dual search occurs in the Lagrange multipliers space. The solution of (D) are the Lagrange multipliers at the optimum, λ^* , and associated values of x are X_{λ^*} . An example of dual function is given in Fig. 1.10. When multiple constraints are handled through the maximization scheme of equation (1.19), λ^* is the optimal Lagrange multiplier of the most critical constraint.

The motivations for solving the dual optimization problem (D) are *i*) to directly solve the primal problem (P) when $X^* = X_{\lambda^*}$, *ii*) to calculate λ^* , which permits formulating exact penalty functions (cf. the “minimal step penalty” later). At first glance, the dual problem seems much more complex than the primal problem since calculating the dual function involves solving an optimization problem, $\min_{x \in S} L(x, \lambda)$. However, favorable properties of the dual function added to the possibility of approximately solving (D) make duality a powerful approach for rational constraints handling.

Property 1 (Concavity of ϕ) *The dual function $\phi(\lambda)$ is concave in λ .*

Property 2 (sub-gradient) *For all $\lambda \geq 0$, let us denote $X_\lambda = \{x \in S / L(x, \lambda) = \phi(\lambda)\}$. Then, for all $x_\lambda \in X_\lambda$, $g(x_\lambda)$ is a sub-gradient of ϕ at λ .*

The two above properties, proofs of which can be found in [138], are valid under very general conditions (f and g bounded). They considerably simplify the resolution of (D) since ϕ is a concave function with a known sub-gradient. Lagrangian based penalty functions, such as the ones introduced in [Luersen and Le Riche, J2004] (summarized hereafter as “adaptive linear penalty”), [16, 73] and [107], have penalty adaptation schemes where, schematically, the Lagrange multiplier is increased if the current best solution in terms of the penalized objective function is infeasible ($g(x_b) > 0$) and vice versa. Since $g(x_b)$ is an approximation of the sub-gradient of $\phi(\lambda)$, those penalty adaptation schemes are variations of a gradient based dual search. An alternative strategy is taken to solve the dual problem in [Le Riche and Guyon, J2002] (see later “minimal step penalty and evolutionary optimization”).

Approximate dual problem In terms of λ , (D) is easy to solve because it is a concave problem with a known subgradient. (D) has no local maximum and many algorithms exist to solve it (nondifferentiable optimization or linear programming, such as presented hereafter). The main difficulty remains the resolution of $\min_{x \in S} \{f(x) + \lambda g(x)\}$ at a given λ in the primal space. For this reason, the dual optimization problem is now approximated by restricting S to a discrete set of points T . It yields an approximate dual function,

$$\phi_T(\lambda) = \min_{x \in T} \{f(x) + \lambda g(x)\}, \quad (1.24)$$

and an approximate dual problem,

$$(AD) \quad \max_{\lambda \geq 0} \phi_T(\lambda), \quad (1.25)$$

where $T \subset S$ is a set of points of the primal space. By construction ϕ_T is concave and piecewise linear. (AD) can be formulated as a linear programming problem:

$$(AD) \quad \begin{cases} \max_{w, \lambda \geq 0} w, \\ \text{such that } f(x_i) + \lambda g(x_i) \geq w, \forall x_i \in T, \\ \lambda \leq \lambda_{max}, \end{cases} \quad (1.26)$$

where λ_{max} is an arbitrarily large upper bound on Lagrange multiplier meant to ensure the existence of a solution to (AD) even when all points in T are infeasible. The linear programming problem (AD) is efficiently solved by a simplex algorithm (cf. [Le Riche and Guyon, T2001]). Let its solution be λ_T .

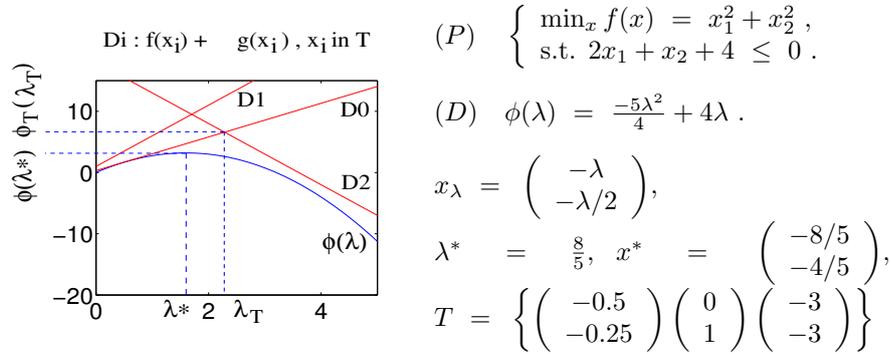


Figure 1.10: Example of dual and approximate dual functions, problem with a saddle point.

Adaptive linear penalty as generalized Lagrangian

An adaptive linear penalty function is the simplest way to handle general inequality constraints. (P) is replaced by

$$(PP) \begin{cases} \min_{x \in S} \bar{L}(x, \lambda), & \text{where} \\ \bar{L}(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i \max(0, g_i(x)). \end{cases} \quad (1.27)$$

This last problem is unconstrained, but appropriate values of the penalty parameters λ_i need to be estimated. More usual penalization approaches are based on quadratic penalty, ordinary Lagrangian or augmented Lagrangian functions. The current adaptive linear penalty has the following advantages, proofs of which are given in Appendix B of [Luersen et al., J2003] using generalized Lagrangian theory ([161, 138]):

- With respect to a quadratic penalty, convergence to the feasible optimum can be achieved for finite values of the parameters λ_i .
- With respect to an ordinary Lagrangian, generalized duality theory can be applied to calculate the λ_i 's for a larger class of functions f and g_i . In other terms, if $\lambda \geq \lambda^*$, where λ^* are the ordinary Lagrange multipliers, then minimizing $\bar{L}(x, \lambda)$ will lead to x^* for more functions than minimizing the ordinary Lagrangian $L(x, \lambda^*)$ would.
- Finally, augmented Lagrangians have more penalty parameters to set than \bar{L} . Unlike \bar{L} , if f and the g_i 's are differentiable, augmented Lagrangians are differentiable at places where $g(x) = 0$. This however is not a decisive drawback of \bar{L} when a non differentiable framework is assumed (e.g., with EAs, GBNM, other pattern search methods, ...).

The penalty parameters are updated after each generation of a new point, x^{new} . The updating scheme is intuitive since it consists of increasing penalty parameters

of violated constraints,

$$\begin{aligned}
& \text{if } (\bar{L}(x^{\text{new}}, \lambda^k) \leq \bar{L}(x^{\text{best}}, \lambda^k)) , \\
& \quad \lambda_i^{k+1} = \lambda_i^k + s \max(0, g_i(x^{\text{new}})) \quad , \quad i = 1, m , \\
& \quad x^{\text{best}} = \arg \min_{x \in \{x^{\text{new}}, x^{\text{best}}, \text{archive of other points}\}} \bar{L}(x, \lambda^{k+1}) , \\
& \text{end} ,
\end{aligned} \tag{1.28}$$

where s is a positive step size. This updating strategy is interpreted as a fixed step approximate gradient search on the dual function (see Appendix B of [Luersen et al., J2003]).

On simple functions like the bounded Rosenbrock problem,

$$\begin{cases} \min_{x_1, x_2 \in [0, 20]} 100(x_2 - x_1^2)^2 + (1 - x_1)^2 , \\ \text{such that } 4 - x_1^2 \leq 0 . \end{cases} \tag{1.29}$$

the above scheme, included in the GBNM algorithm (see section 1.3 and [Luersen et al., J2003]) converges to $\lambda = 0.5 \pm 1.10^{-6}$ after about 800 analyses (based on 100 independent runs, $\lambda^0 = 0$ and $s = 0.001$). However, other experiments with more complex functions (higher dimensions, with local optima) have shown that this adaptive penalty scheme, although of an appealing simplicity, has a convergence which depends a lot on the starting point and step size. The following minimal step penalty, associated with a proper solution of the approximate dual problem, is more stable and therefore recommended.

Minimal step penalty and evolutionary optimization

Augmented Lagrangian functions and Lagrange multipliers updating have been applied to derive adaptive penalty functions in evolutionary algorithms. Bean and Hadj-Alouane ([16]) have proposed a penalty adaptation scheme which resembles Lagrange multipliers updating strategies. Kim and Myung ([107]) and Tahk and Sun ([182]) have used augmented Lagrangian penalty functions in evolutionary optimization, calculating Lagrange multipliers as a by-product of the search. In [182], a coevolutionary algorithm simultaneously evolves a population of unknowns variables and a population of Lagrange multipliers.

The current work is also concerned with solving the dual optimization problem as a way to adapt a penalty function. Fundamentally, it differs from previous works in two aspects. Firstly, an evolutionary algorithm is devised which explicitly solves the dual optimization problem. Secondly, the penalty function is not an augmented Lagrangian. Indeed, augmented Lagrangians were originally derived for mathematical programming. They are continuously differentiable functions. They depend on the choice of a penalty parameter (the ‘‘augmented’’ term), which, if taken too small, leaves local optima. Continuous differentiability is not needed in evolutionary optimization. The freedom gained in the formulation of the penalty permits removing the parameter and obtaining global optimality and minimal penalty properties.

A dual evolutionary algorithm The dual evolutionary optimizer iterates between the primal problem (P_λ) and the approximate dual problem (AD). Based on a particular choice of λ , P_λ resolution by evolutionary optimization produces points to include in T . Based on T , (AD) resolution by the simplex algorithm yields a new Lagrange multiplier λ_T . Most dual optimization methods iterate between primal and dual spaces. Our algorithm bears particular resemblance to Dantzig’s algorithm ([36]). The difference lies in the evolutionary primal optimization:

- It can visit different basins of attraction of the Lagrangian, $L(x, \lambda)$, during convergence, i.e., it can yield many judicious points to be included in T at each iteration.
- It can handle non-convex, discontinuous functions.

A flowchart of a dual evolutionary optimizer is given in Fig. 1.11. X_k^f and X_k^i are the sets of feasible and infeasible active points of (AD) at iteration k , respectively. The evolutionary algorithm used is a steady-state algorithm with continuous mutation and crossover, and tournament selection ([11]). Evolutionary searches are stopped as soon as an improvement on $L(x, \lambda_k)$ has been observed. This is an important implementation aspect as it saves much computational effort that would otherwise be spent minimizing the Lagrangian with λ_k far from λ^* . As a side effect, this stopping criterion increases the number of resolutions of (AD) . The cost of solving (AD) is however negligible because no evaluation of f or g is performed and the simplex algorithm is efficient. In all the tests performed (cf. [Le Riche and Guyon, J2002]), the CPU time spent in (AD) is less than a percent of the total CPU time for T_k sets of up to 10000 elements. Further details on the simplex implementation, existence of X_k^f , and convergence rate of the method can be found in [Le Riche and Guyon, T2001]. Important outputs of the algorithm are $\lambda_{k_{final}}$ and $X_{k_{final}}^f$. $\lambda_{k_{final}}$ is an estimate of λ^* . $X_{k_{final}}^f$ is an estimate of the feasible points in X_{λ^*} .

1. $k = 0$, initialize λ_0 , $\phi_0 = DBL_MAX$.
2. Evolutionary (primal) search minimizing on x $L(x, \lambda_k)$. If $k > 0$, include X_{k-1}^f and X_{k-1}^i in the initial population. Stop when a point x' has been found such that $L(x', \lambda_k) < \phi_k$.
3. Add x' plus other n_f and n_i best feasible and infeasible individuals according to $L(x, \lambda_k)$ to $T_k \rightarrow T_{temp}$.
4. Simplex exact resolution of dual (AD) with T_{temp} according to formulation (1.26) $\rightarrow \lambda_{temp}$, X_k^f , X_k^i .
5. $k = k + 1$, $T_k = T_{temp}$, $\lambda_k = \lambda_{temp}$, $\phi_k = f(x_k^f) + \lambda_k g(x_k^f)$. If cumulated number of analyses $> N_{max}$, $k_{final} = k$, stop. Else go to 2.

Figure 1.11: Dual evolutionary optimizer

T_k gathers information from many potentially important points sampled by all evolutionary runs up to iteration k . Values of Lagrange multipliers are inferred from these points through an exact resolution of the approximate dual problem (AD) . Such an approach is thought to be more efficient than gradient based dual searches which change λ_k based on a local information, an approximation of $g(x_{\lambda_k})$, $x_{\lambda_k} \in X_{\lambda_k}$.

A minimal, exact, step penalty function The previous paragraph has introduced a coupled evolutionary / simplex algorithm for solving the dual problem (D) . But the goal is to tackle the primal problem (P) . In fact, problems having a saddle point at the optimum are readily solved because in this case, $X^* = \{x^*\}$, x^* unique, $X_{\lambda^*} = \{x_{\lambda^*}\}$, x_{λ^*} unique, and $x^* = x_{\lambda^*}$ ([138]). In other terms, the dual and the primal problems are equivalent. The dual evolutionary algorithm provides $X_{k_{final}}^f$

(cf. Fig. 1.11), which includes an estimate of x^* . To sum up, the Lagrangian is a valid penalty function for problems having a saddle-point.

For problems without saddle point, solving (D) does not directly provide a solution to (P), $X_{\lambda^*} \neq X^*$. This can be seen on the example of Fig. 1.12 where $X_{\lambda^*} = \{1.058, 4.58\}$, $x^f = 1.058$ and $X^* = \{4.5\}$. Problems without a saddle point require using another penalty function. Nevertheless, as will soon be seen, solving (D) still generates information for properly penalizing the constraints : X_{λ^*} contains at least one feasible element denoted x^f , $g(x^f) \leq 0$ (see [Le Riche and Guyon, T2001]).

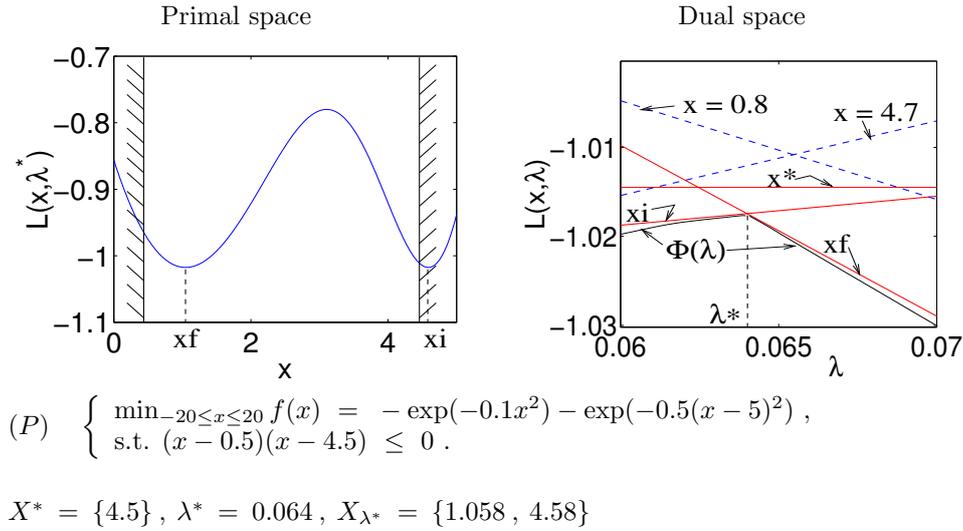


Figure 1.12: Example of dual and approximate dual functions, problem without a saddle point.

Let F_p denote any penalized objective function. The choice of the penalty function has a profound effect on the evolutionary optimization efficiency. When too high a penalty is imposed on infeasible points, the population is prematurely pushed into the feasible domain, often far from optima x^* . Subsequent convergence to x^* can be extremely slow. In evolutionary terms, penalization makes F_p deceptive. Reciprocally, if too low a penalty is enforced, the algorithm converges into the infeasible domain. The optimal penalty function is problem dependent. However, several authors have described a reasonable heuristic, the minimal penalty rule, as a remedy against penalization induced deceptiveness (Davis [38], Richardson et al. [160], Smith and Tate [178]). It says: on average, it is best to apply the smallest amount of penalty such that the algorithm converges to a feasible optimum, x^* . For calculation purposes, a more precise definition of “amount of penalty” is needed.

Définition 1 (Amount of penalty) For optimization problems without a saddle point and such that there is an infeasible solution to the dual, x^i , the amount of penalty, r , is defined as,

$$r = F_p(x^i) - f(x^i), \quad (1.30)$$

where F_p is any penalized objective function.

A class of Lagrangian based exact penalty function is now introduced.

Proposition 1 (A class of exact penalty functions) For f and g bounded, let $f_p(x; \lambda^+, x^f)$ be defined as,

$$f_p(x; \lambda^+, x^f) = f(x) + H(g(x))[\lambda^+ g(x) - \lambda^+ g(x^f) - f(x^f) + f(\widehat{x}^*) + \epsilon] , \quad (1.31)$$

where,

$$\begin{aligned} H(y) &= 0 \text{ if } y \leq 0 , H(y) = 1 \text{ otherwise,} \\ \lambda^+ &\geq \lambda^* , x^f \in X_{\lambda^*} / g(x^f) \leq 0 , \epsilon > 0 , \\ \widehat{x}^* &\text{ is the known feasible point with lowest } f . \end{aligned}$$

$f_p(x; \lambda^+, x^f)$ has an absolute minimum at $x^* \in X^*$.

This class of penalty functions contains a minimal penalty function.

Proposition 2 (A minimal step penalty function) Among exact penalty functions, L_p , based on the addition of a step, p , to a generalized Lagrangian,

$$L_p(x, \lambda^*) = f(x) + H(g(x))[\lambda^* g(x) + p] , \quad (1.32)$$

$f_p(x; \lambda^*, x^f)$ uses the smallest amount of penalty.

The proofs of Propositions 1 and 2 along with a more gentle introduction to f_p can be found in [Le Riche and Guyon, J2002 and T2001].

Proposition 1 explains how a constrained evolutionary optimization using $f_p(x; \lambda^*, x^f)$ as penalty function converges to an optimum. In addition, because it is a minimal penalty strategy (Proposition 2), it promotes fast convergence. An evolutionary optimizer for general constrained optimization problems is described in Fig. 1.13. Finally, we emphasize that no parameter of the penalty function is arbitrarily set since λ^* and x^f have a precise definition in terms of (D) . λ_0 , N_{max} , n_f and n_i control the rate of convergence in the dual space. These parameters have little influence compared to penalty parameters.

1. Run the dual evolutionary algorithm of Fig. 1.11
 $\longrightarrow \lambda_{k_{final}} , x_{k_{final}}^f , x_{k_{final}}^i .$
2. Final evolutionary search minimizing on x $f_p(x; \lambda_{k_{final}}, x_{k_{final}}^f)$.
 $x_{k_{final}}^f$ and $x_{k_{final}}^i$ are included in the initial population.

Figure 1.13: Evolutionary optimization based on f_p .

A graphical representation of the convergence of an EA on the hoop problem is given in FIG. 1.14. The feasible domain is a thin curved shell. On this problem, a reasonable penalty, $p = 10$ in the static penalty of equation (1.18), is too large and induces a premature convergence inside the feasible domain, but far from the optimum. Progress is then slow. The minimal step penalty function, $f_p(x; \lambda^*, x^f)$, authorizes short cuts through the infeasible domain and induces a faster convergence to x^* . More details and numerical tests can be found in [Le Riche and Guyon, J2002]. The experiments illustrate three claims. Firstly, the minimal penalty function promotes fast and reliable convergence as compared to arbitrarily tuned static penalty functions. Secondly, the dual evolutionary optimizer is a better strategy than an adaptive linear penalty. Thirdly, the dual evolutionary optimizer is not sensitive to its parameters setting.

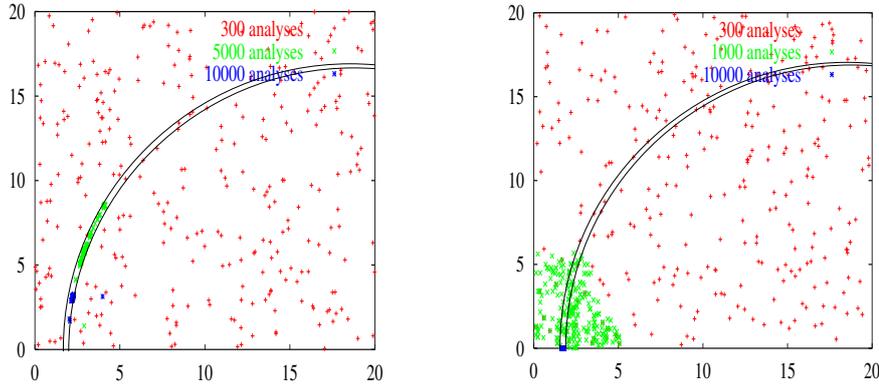


Figure 1.14: Comparison of EA convergences for varying amounts of penalty. Hoop problem : the feasible domain is inside the circles and $x^* = (1.9, 0)^T$. Left: $F_p(x) = f(x) + 10 \max(0, g(x))$; for large penalty the population is confined to the feasible domain and convergence is slow. Right: $F_p(x) = f_p(x; \lambda^*, x^f) = f(x) + 7.e - 5 \max(0, g(x))$; a minimal penalty permits a faster convergence to x^* .

1.3 GBNM

This section and the following describe approaches to global optimization we have contributed to develop which are not based on evolutionary calculation. We start here with the Globalized and Bounded Nelder-Mead algorithm (GBNM) which was first introduced in [Luersen et al., J2003].

1.3.1 A brief presentation of GBNM

As the name indicates, GBNM is based on the well-known Nelder-Mead algorithm¹⁶ ([142, 198]). The Nelder-Mead algorithm is a local search method that proceeds by simple geometric transformations of $n + 1$ points in n dimensions (a simplex). Referring back to evolutionary algorithms (section 1.1), one could see one iteration of the Nelder-Mead algorithm as a crossover operator in \mathbb{R}^n [49] from $n + 1$ parents. The Nelder-Mead algorithm is popular among engineers because *i*) it is a zeroth order method, i.e., it does not require function gradients, *ii*) it is a very simple method to program, *iii*) it is scale invariant, i.e., its convergence is not affected by any monotonous transformation of the objective function which, in turn, provides a certain robustness for poorly conditioned functions. However, the Nelder-Mead algorithm is not so popular among mathematicians: it can fail if the $n + 1$ points of the simplex are in a subspace that does not contain the local optimum; it converges slowly in more than 10 dimensions because only one of the $n + 1$ points is moved at each iteration. Moreover, it is not a global optimizer. A flowchart of the Nelder-Mead algorithm can be found at the end of [Luersen et al., J2003].

The GBNM algorithm is an attempt to improve on the Nelder Mead algorithm, while keeping a practical approach. By practical we mean a zeroth order method, whose behavior is guided by a total analyses budget, and that accounts for variables bounds and inequality constraints. With respect to the original algorithm, the components added in GBNM are the following:

- The search is made global by a probabilistic restart mechanism. The algorithm performs at least one local Nelder-Mead search. If the analysis budget is

¹⁶also known as sequential simplex method.

not exhausted, a new Nelder-Mead search is started and so on. The new starting point is, in probability, far from past starting and convergence points in order to increase the chances of visiting new basins of attraction. The restart distribution is created with Gaussian Parzen windows.

- Variable bounds are taken into account.
- General inequality constraints are handled by the linear penalty function of equation (1.27) where the Lagrange multipliers are updated using equation (1.28).
- Simplex degeneracy is checked at each Nelder-Mead iteration. A simplex is degenerated if it belongs to a subspace of dimension smaller than n , while not touching variables bounds¹⁷. The test for degeneracy is based on the determinant and the sizes of the simplex edges. If a simplex is degenerated, it is re-initialized from its best point.
- Convergence of Nelder-Mead is checked by re-initializing a Nelder-Mead search with a small simplex from the best point of the converged simplex.

As it can be seen on the GBNM flowchart of FIG. 1.15, the difficulty is to architecture these components together. The output of the GBNM algorithm is the list of Nelder-Mead convergence points. They are candidate local optima and some of them can be global optima.

At the time of this study, the GBNM clearly outperformed a steady state evolutionary algorithm with continuous crossover and fixed Gaussian mutation for functions having up to 12 variables. The readers who are looking for some details should consult [Luersen et al., J2003].

Today, it seems necessary to compare GBNM with more efficient EAs like the CMA-ES [80]. In light of accumulated experience and explanations, we also believe that the GBNM constraints handling mechanism should be improved: the estimation of the Lagrange multipliers should be changed from the simple – but unstable – updating scheme of equation (1.28) to the more robust resolution of the approximate dual problem presented in section 1.2.3.

1.3.2 Application to the design of swimming monofins

The GBNM optimizer was applied to design a swimming monofin in [Luersen et al., J2006]. Monofins, a picture of which is shown in FIG. 1.16, are the most efficient way of swimming for human beings. It is expected that further progress can be achieved because today's monofin design is empirical and studies in aquatic locomotion modes [173] and oscillating hydrofoils [2] show that more efficient swimming systems exist: fish like tuna, mackerel, sharks and marine mammals have propulsive efficiencies greater than 90% at high swimming speed in calm waters.

However, the simulation of the swimmer and monofin system is very complex: the flow is unsteady, it interacts with the fin which is a composite structure in dynamic motion with large displacements. A single simulation of such a system is a research study by itself and it is far too computationally intensive to allow subsequent optimization. The monofin optimization problem was therefore formulated in such a way so as to use efficient 2D fluid-structure simulations but also to provide a 3D design at the end of the process. Accordingly, the problem was decomposed into two subproblems: *i*) a two-dimensional optimization of the fin stiffness distribution which accounts for fluid structure interactions and *ii*) the identification of a 3D fin that is mechanically equivalent to the 2D, previously optimized, fin. Both steps involved optimizations that were performed with the GBNM algorithm.

¹⁷Simplexes are allowed to degenerate along the variables bounds to follow them.



Figure 1.15: Restarts and convergence tests linking in GBNM

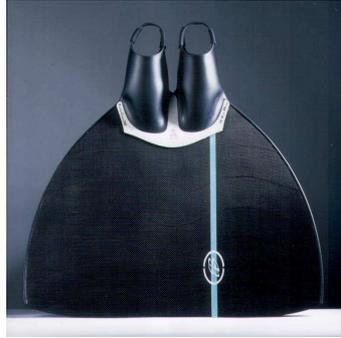


Figure 1.16: Carbon monofin by Breier©.

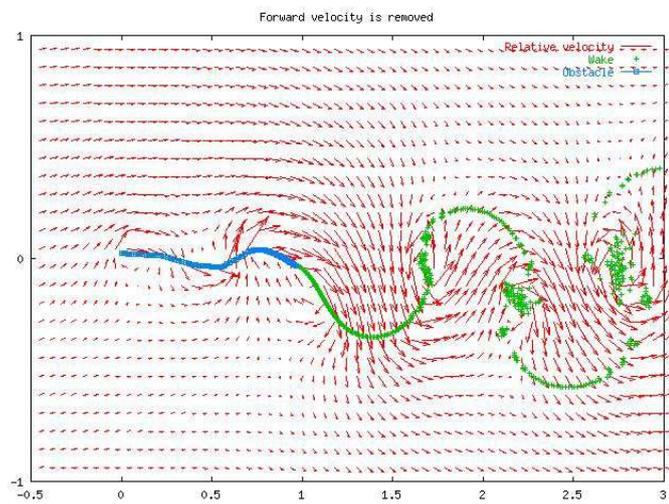


Figure 1.17: Swimmer and fin model. Blue squares are the solid obstacle (swimmer and fin), green crosses are emitted particles, red vectors are the relative speed vectors.

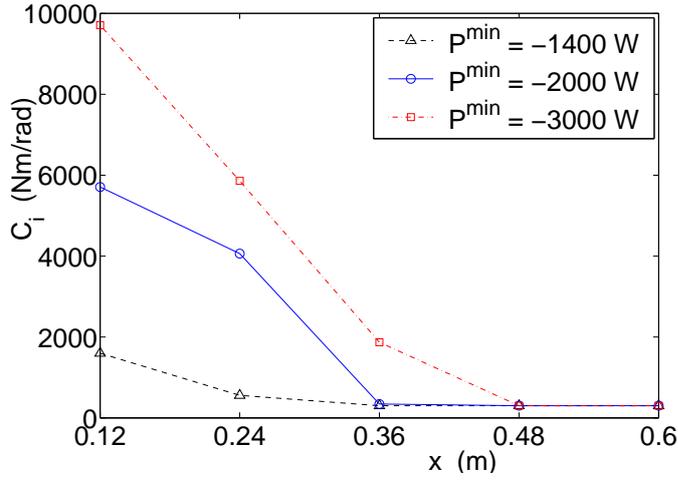


Figure 1.18: Optimal stiffness distribution for various swimmer power upper-bounds $C^{\min} = 300Nm/rad$ and $C^{\max} = 15000Nm/rad$. (The sign convention used in [Luersen et al., J2006] is the contrary of that chosen here. Therefore, powers are negative and upper-bounds become lower-bounds in this figure.).

Two-dimensional fin optimization. If one assumes an inviscid, incompressible flow that remains attached to an obstacle without thickness, the computationally efficient unsteady vortex based flow model of [132] can be used. As it can be seen in FIG. 1.17, the model is efficient because only the solid boundary and the wake need to be discretized. The swimmer is represented by four linear segments (the arms, the torso, the thighs and the tibias) with imposed displacements. The monofin is calculated as six rigid bars articulated by five torsional springs of stiffnesses C_i , $i = 1, \dots, 5$. At each iteration of the swimmer-monofin simulation, the swimmer position is updated. Next, the orientation, angular velocities and angular accelerations of the monofin's bar joints are calculated to satisfy the monofin dynamic equations. Once the equations are satisfied, the forces exerted by the fluid on the obstacle are known. One can calculate the time-averaged propulsive power of the fin, \bar{P}_{fx} , and the total power spent by the swimmer to move the fin in the water, \bar{P}_f .

The 2D design of the monofin is formulated as the maximization of the propulsive power, while the total power spent by the swimmer is bounded,

$$\begin{cases} \max_{C_i} \bar{P}_{fx} \\ \text{such that } \bar{P}_f < \bar{P}^{\max} \quad \text{and} \quad C^{\min} \leq C_i \leq C^{\max} \quad , \quad i = 1, 5 . \end{cases} \quad (1.33)$$

Problem (1.33) was solved with the GBNM method for three swimmer power upper-bounds. The optimal stiffness distributions are plotted in FIG. 1.18. They are tapered from the leading to the trailing edge. Changes in swimmer total power affect mainly the fin near the leading edge, while low stiffness is always optimal near the trailing edge.

Translation into a 3D structure. The 2D optimized stiffness distribution is now translated into a 3D structure. The mapping can be seen as an identification procedure where the “experience” is a 2D bar system whose behavior is approximated by a 3D finite element model of the fin. In its most general statement, this

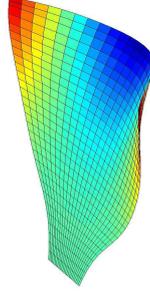


Figure 1.19: Modal FE analysis of a fin for identifying longitudinal thickness distribution.

identification problem is ill-posed since the 3D system has more degrees of freedom than its 2D counterpart. Many combinations of shape and thickness distribution can represent the 2D monofin. In practice, however, the planform shape of the monofin is dictated by manufacturing (cost of molds) and marketing considerations which has yielded forms that mimick marine mammals (cf. FIG. 1.16). Once the fin planform shape is fixed, the spring stiffnesses are mapped into a fin thickness distribution. Because the fin is manufactured using composite prepreg layup, the thickness is constant spanwise and varies chordwise at ply drops.

The equivalence between the two models can be sought in terms of static behavior, modal behavior, or a mix of static and modal behavior. The advantage of static behavior is that large displacements analyses are available. However, it neglects the fin inertia in comparison to water inertia and fin flexural stiffness. On the contrary, the modal dynamic identification accounts for both fin inertia and flexural stiffness but it is, in essence, a small displacements analysis. Furthermore, 3D non-bending modes have no pendant in the 2D system and higher natural bending frequencies are far away from the frequency of the imposed movement. For these reasons, only the first natural mode, which was empirically found for the monofin to consistently be bending (see FIG. 1.19), is considered.

The 3D thickness distribution is found by minimizing

$$J = \alpha J_{\text{static}} + (1 - \alpha) J_{\text{freq}}$$

where J_{static} is a distance between deformed shapes of the 2D model and the plane of symmetry of the 3D fin, J_{freq} is a distance between first eigenfrequencies of the 2D and 3D fins, and $0 \leq \alpha \leq 1$. J is minimized in terms of six thicknesses in [Luersen et al., J2006] using the GBNM algorithm. It is observed that taking $\alpha = 0$, i.e., accounting only for the distance in terms of first eigenfrequency, resulted in an ill-posed problem: many thickness distributions have the same first eigenfrequency. When accounting only for static displacements ($\alpha = 1$), besides a few pathological load cases that make the identification impossible because the deformed shape is not sensitive to some of the C_i s, no difference in identified thickness distribution is observed between small and large displacements formulations. The recommended approach in order to take the most physics into account and to improve the identi-

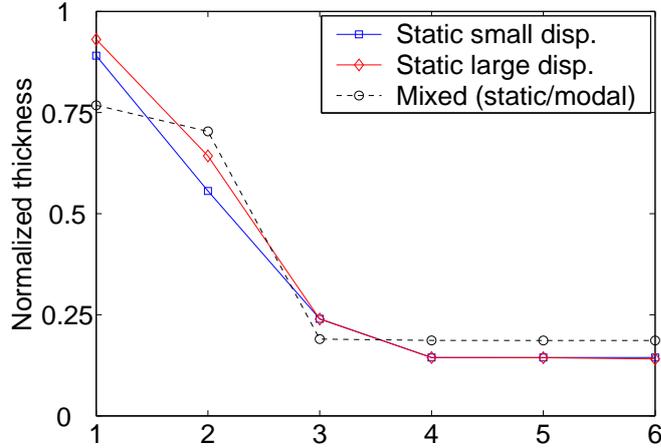


Figure 1.20: Comparison of identified monofin thickness distributions from different formulations. The x-axis is a normalized chord length, the leading edge is at $x = 1$ and the trailing edge at $x = 6$. The target 2D model was the optimum for $\bar{P}_f = 2000W$. The obtained profiles are more tapered than those already manufactured by Breier Inc.

fiability, is to mix the static and dynamic norms ($\alpha = 0.5$). Results for the various formulations are compared in FIG. 1.20.

1.4 A multi-point global optimization criterion based on Gaussian processes

1.4.1 Space filling design of experiment for optimization

We now turn to the global optimization methods proposed in [Ginsbourger et al., 2007]. These methods are based on a principle fundamentally different from that underlying EAs (with the exception of EDAs) and GBNM: whereas EAs and GBNM have search mechanisms based on randomized steps from particular space points, we consider here methods that build a metamodel of $f(x)$ in the entire search space \mathcal{S} from a list of already calculated points $(x_i, f(x_i))$, $i = 1, m$. To create optimization methods that are global in scope, it is necessary that the metamodel provides at least two responses at each point x , a prediction of f and a measure of its uncertainty. The optimizer then trades off resource allocation between high-performance and uncertain regions of the design space. An early example of such process is the DIRECT method [46] where, in simplified terms, the prediction is the function value $f(x_i)$ of the nearest already sampled point x_i , and the uncertainty is the distance to x_i . Such optimization methods typically define space filling design of experiments, the space filling property explaining why they are global in scope. Since they proceed by building global metamodels in \mathcal{S} , they are not appropriate for problems that have more than $n = 10$ dimensions.

1.4.2 Ordinary kriging

The current work lie within the framework of ordinary kriging probabilistic meta-models [33]: the function that is optimized is supposed to be a realization of the spatial random process

$$F(x) = \bar{f} + \varepsilon(x), \quad (1.34)$$

where \bar{f} is an unknown constant mean and $\varepsilon(x)$ a stationary Gaussian process of known covariance¹⁸. With these assumptions, the density of $F(x)$ knowing its realizations $F(x_i) = f(x_i)$ is analytically known¹⁹,

$$[F(x)|F(x_1) = f(x_1) \dots F(x_m) = f(x_m)] \sim \mathcal{N}(m_{OK}(x), s_{OK}^2(x)) \quad (1.35)$$

Full expressions of m_{OK} and s_{OK}^2 in terms of the $\varepsilon(x)$ covariance and the set $(x_i, f(x_i))$ can be found in [Ginsbourger et al., C2007]. At each point x , the function process conditioned by already calculated points has a Gaussian density of mean $m_{OK}(x)$ and variance $s_{OK}^2(x)$. The probabilistic model of equation (1.35) is reasonable for deterministic functions f : for all x_a and x_b in \mathcal{S} , $[F(x_a)|F(x_i) = f(x_i), i = 1, m]$ and $[F(x_b)|F(x_i) = f(x_i), i = 1, m]$ are correlated Gaussian variables (and the expression of their correlation is known); the closer x_a and x_b are, the more correlated their kriging function values are; $m_{OK}(x_i) = f(x_i)$ and $s_{OK}^2(x_i) = 0$, therefore $[F(x)|F(x_i) = f(x_i), i = 1, m]$ is interpolating. An illustration of the ordinary kriging distribution is given in FIG. 1.6. It should also be noted that the form of the covariance of $\varepsilon(x)$ is linked to the regularity of the instantiated function. For example, if

$$\text{Correlation}(F(x), F(y)) = \exp \left[\sum_{k=1}^n \theta_k |x_k - y_k|^{p_k} \right]$$

and $p_k = 2$, then f is infinitely differentiable [184].

1.4.3 Kriging-based optimization criteria

A review of optimization methods relying on metamodels can be found in [45]. It analyzes and illustrates why directly optimizing a deterministic metamodel (like a spline, a polynomial, or the kriging mean) may be dangerous, and does not even necessarily lead to a local optimum. Kriging-based sequential optimization strategies (as developed in [93]) may avoid converging to non (locally) optimal points by taking the kriging variance term, s_{OK}^2 , into account, hence encouraging the algorithms to explore unvisited zones. Such optimization algorithms produce at each iteration one point that maximizes a figure of merit (or criterion) based upon $[F(x)|F(\mathbf{x}) = \mathbf{f}]$ (where $F(\mathbf{x}) = \mathbf{f}$ is a shorthand notation for $F(x_1) = f(x_1), \dots, F(x_m) = f(x_m)$). In essence, the criteria balance kriging prediction and uncertainty.

Visiting the point with highest uncertainty: maximizing s_{OK} . The fundamental mistake of minimizing the kriging mean (m_{OK}) when globally minimizing a function is that the uncertainty associated with m_{OK} is not taken into account. At the extreme inverse, it is possible to define the next optimization iterate as the least known point in \mathcal{S} ,

$$x' = \operatorname{argmax}_{x \in \mathcal{S}} s_{OK}(x)$$

¹⁸In kriging-based optimization, one often improperly uses maximum likelihood covariance hyperparameters without taking the variance of the estimation into account. This approximation has the advantage of delivering a Gaussian posterior distribution (1.35), even if the uncertainty is slightly underestimated.

¹⁹Cf. the theory of conditioned Gaussian vectors [26].

This procedure defines a series of x 's which will fill the search space (it is dense in \mathcal{S}) and, in this sense, it will ultimately locate a global optimum x^* . Yet, since no use is made of previously obtained $f(x_i)$ information (look at the formula for s_{OK}^2 [Ginsbourger et al., C2007]), there is no bias in favor of high performance regions. Maximizing the uncertainty is inefficient in practice.

Compromises between m_{OK} and s_{OK} . The most general formulation for balancing the exploitation of previous simulations, which is supplied by m_{OK} , with the exploration, which is estimated by s_{OK} , is the two criteria problem

$$\begin{cases} \min_{x \in \mathcal{S}} m_{OK}(x) \\ \text{and } \max_{x \in \mathcal{S}} s_{OK}(x) \end{cases}$$

Let \mathcal{C} denote the Pareto set of solutions²⁰. Finding one (or many) elements in \mathcal{C} remains a difficult problem since \mathcal{C} typically contains an infinite number of points. We have mentioned earlier the DIRECT method [46]. Although DIRECT is not based on kriging, it uses a Pareto set to balance exploration (distance to the nearest neighbors) and performance (f of nearest neighbors). All subregions that are Pareto optimal receive computational resources at the next iteration. Note that [24] proposes a parallelized version of DIRECT.

Maximizing the probability of improvement At first sight, the probability of improving the function below the currently known minimum $\min(\mathbf{f}) \equiv \min(f(x_1), \dots, f(x_m))$ may seem to be crucial to an optimizer efficiency:

$$\begin{aligned} PI(x) &= P(F(x) \leq \min(\mathbf{f}) | F(\mathbf{x}) = \mathbf{f}) \\ &= \Phi \left(\frac{\min(\mathbf{f}) - m_{OK}(x)}{s_{OK}(x)} \right) \end{aligned}$$

ϕ and Φ stand for the probability density function and the cumulative distribution function of the standard normal law $\mathcal{N}(0, 1)$, respectively. $\min(\mathbf{f})$ is sometimes replaced by some arbitrary target $T \in \mathbb{R}$. However, the PI criterion is known to provide a very local search whenever the value of T is close to $\min(\mathbf{f})$. Taking several T 's is a remedy proposed in [45] to force global exploration.

Maximizing the expected improvement An alternative solution is to maximize the *expected improvement*

$$EI(x) = \mathbb{E} [(\min(\mathbf{f}) - F(x))^+ | F(\mathbf{x}) = \mathbf{f}] \quad (1.36)$$

which additionally takes into account the magnitude of the potential improvement. Above, $(u)^+$ means $\max(0, u)$. EI measures how much improvement is expected when sampling at x . In fine, the improvement will be 0 if the actual $f(x)$ is above $\min(\mathbf{f})$ and $\min(\mathbf{f}) - f(x)$ in the opposite case. Since we know the conditional distribution of $F(x)$, it is straightforward to calculate EI in closed-form (see [93]):

$$\begin{aligned} EI(x) &= \mathbb{E}[(\min(\mathbf{f}) - F(x))\mathbb{I}_{F(x) \leq \min(\mathbf{f})} | F(\mathbf{x}) = \mathbf{f}] \\ &= (\min(\mathbf{f}) - m_{OK}(x))\Phi \left(\frac{\min(\mathbf{f}) - m_{OK}(x)}{s_{OK}(x)} \right) + s_{OK}(x)\phi \left(\frac{\min(\mathbf{f}) - m_{OK}(x)}{s_{OK}(x)} \right) \end{aligned} \quad (1.37)$$

²⁰Definition of the Pareto front of $(s_{OK}, -m_{OK})$: $\forall x \in \mathcal{C}, \nexists y \in \mathcal{S} : (m_{OK}(y) < m_{OK}(x) \text{ and } s_{OK}(y) \geq s_{OK}(x)) \text{ or } (m_{OK}(y) \leq m_{OK}(x) \text{ and } s_{OK}(y) > s_{OK}(x))$

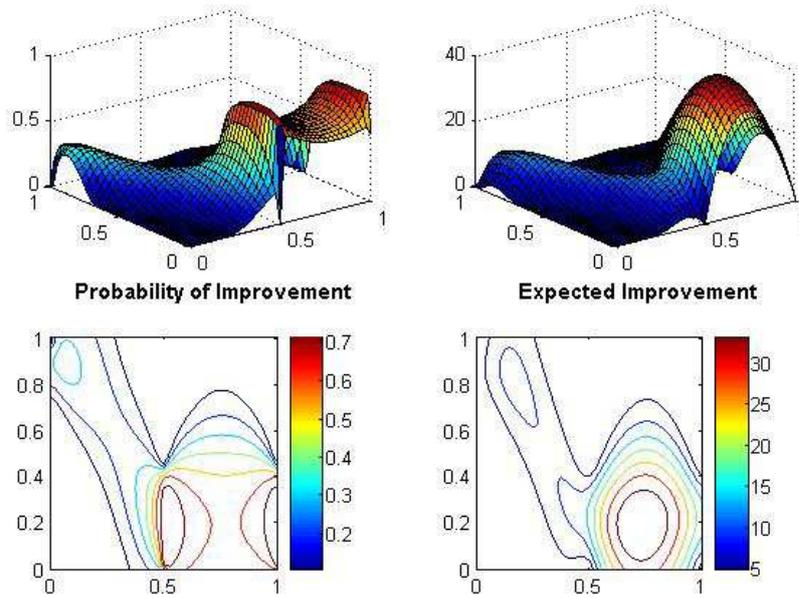


Figure 1.21: PI and EI surfaces of the Branin-Hoo function. The design of experiments is a 3×3 factorial design. The covariance is an anisotropic squared exponential with parameters estimated by gaussian likelihood maximization [33]. Maximizing PI leads to sample near the good points whereas maximizing EI leads here to sample between the good points. By construction, both criteria are null at the design of experiments, but the probability of improvement is very close to $\frac{1}{2}$ in a neighborhood of the point(s) where the function takes its best observed value.

EI represents a trade-off between promising and uncertain zones. EI has important properties for sequential exploration: it is null at the already visited sites, and positive everywhere else with a magnitude which is increasing with the kriging variance and with the decreasing kriging mean (EI maximizers are part of the Pareto front of $(s_{OK}, -m_{OK})$). The expected improvement and the probability of improvement are compared in FIG. 1.21.

The EGO algorithm ([93]) relies on the EI criterion. Starting with an initial Design \mathbf{x} (typically a Latin Hypercube), EGO visits, at each iteration, the current global maximizer of EI and updates the kriging metamodel (including hyperparameters re-estimation):

1. Evaluate f at \mathbf{x} , set $\mathbf{f} = f(\mathbf{x})$ and estimate covariance parameters of F by MLE (Maximum Likelihood Estimation)
2. While stopping criterion not met
 - (a) Compute $x' = \operatorname{argmax}_{x \in \mathcal{S}} EI(x)$, set $\mathbf{x} = \mathbf{x} \cup \{x'\}$ and $\mathbf{f} = \mathbf{f} \cup \{f(x')\}$
 - (b) Re-estimate covariance parameters by MLE

EGO was developed and applied in [126]. It was then considered as a reference and has inspired contemporary works in optimization of expensive-to-evaluate functions. For instance, [35] exposes some EGO-based methods for the optimization of noisy black-box functions. [111] proposes an adaptation of EGO to multi-objective optimization.

The Stepwise Uncertainty Reduction strategy (SUR) has been introduced in [34] and extended to global optimization in [90]. By looking at possible objective functions as conditional processes, $F(x)|\mathbf{f}$, it is possible to define $X^*|\mathbf{f}$, the random vector of the location of the minimizer of $F(x)|\mathbf{f}$, of density $p_{X^*|\mathbf{f}}(x)$. The uncertainty about the location of the optimum of $F(x)$ is measured as the entropy of $p_{X^*|\mathbf{f}}$, $H(X^*|\mathbf{f})$. $H(X^*|\mathbf{f})$ diminishes as the distribution of $X^*|\mathbf{f}$ gets more peaked. Conceptually, the SUR strategy for global optimization chooses as next iterate the point that specifies the most the location of the optimum,

$$x' = \operatorname{argmin}_{x \in \mathcal{S}} H(X^*|\mathbf{f}, F(x))$$

In practice, $p_{X^*|\mathbf{f}, F(x)}$ is estimated by Monte-Carlo sampling of $F(x)|\mathbf{f}, F(x)$ at a finite number of locations in \mathcal{S} , which may become a problem in high dimensional spaces as the number of locations must geometrically increase with the number of dimensions to properly fill the space. The SUR criterion is different in nature from the other criteria presented so far in that it does not maximize an immediate (i.e. at the next iteration) payoff defined in terms of F but rather lays the foundation of a more delayed payoff by gaining a more global knowledge on F (reduce the entropy of its optima). The multi-points expected improvement criterion introduced next also uses a delayed payoff measure.

1.4.4 The multi-points expected improvement

Definition The multi- (say q -) points expected improvement, or q -EI, generalizes the expected improvement seen in the previous section. It is the expected improvement of q points,

$$\begin{aligned} EI(x^{m+1}, \dots, x^{m+q}) &= \\ \mathbb{E} \left[\max \left((\min(\mathbf{f}) - F(x^{m+1}))^+, \dots, (\min(\mathbf{f}) - F(x^{m+q}))^+ \right) \mid F(\mathbf{x}) = \mathbf{f} \right] &= \quad (1.38) \\ \mathbb{E} \left[(\min(\mathbf{f}) - \min(F(x^{m+1}), \dots, F(x^{m+q})))^+ \mid F(\mathbf{x}) = \mathbf{f} \right] & \end{aligned}$$

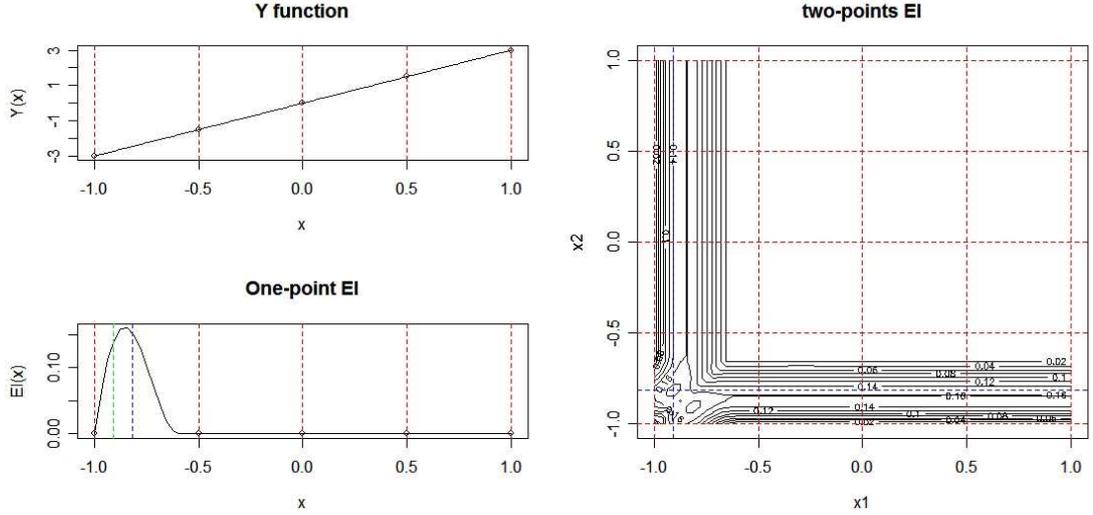


Figure 1.22: Comparison between 1-point EI (lower left) and 2-points EI (right). The learned function, $y(x) = 3x$, is known at $\mathbf{x} = \{-1, -0.5, 0, 0.5, 1\}$. The couple of points that maximize 2-EI is plotted using dashed lines on the EI curve (lower left). These two points bracket the maximizer of EI but are different from it. (The ordinary kriging has a cubic covariance with parameters $\sigma^2 = 10$, scale = 1.4.)

q -EI was first defined in [126]. It was later developed in D. Ginsbourger's doctoral thesis and [Ginsbourger et al., C2007] within the framework of (simple and ordinary) kriging. The q -EI criterion is a function of q vectors, x^{m+1}, \dots, x^{m+q} . Like the SUR criterion, q -EI does not target an immediate payoff but rather a delayed return after q objective function calculations. We believe this is an important property for finite cost global optimizers: if it is known that q analyses will be computed, maximizing q times an immediate payoff is not the best strategy because some time can be devoted to better exploring the search space. Another advantage of q -EI is that maximizing it yields q points whose objective functions can then be calculated in parallel. An optimizer using the q -EI criterion may therefore be labelled a *parallelized EGO*.

In order to better understand and calculate q -EI, one needs to say that the variables $[F(x^{m+1})|F(\mathbf{x}) = \mathbf{f}], \dots, [F(x^{m+q})|F(\mathbf{x}) = \mathbf{f}]$ are not independent: any piece of information obtained at a certain point in space carries over to neighboring points in a way accurately described by kriging. The statistical law followed by the conditioned vector of objective function values is analytically known,

$$[(F(x^{m+1}), \dots, F(x^{m+q}))|F(\mathbf{x}) = \mathbf{f}] \sim \mathcal{N}((m_{OK}(x^{m+1}), \dots, m_{OK}(x^{m+q})), S_q) \quad (1.39)$$

The expression of S_q is given in [Ginsbourger et al., C2007]. An analytical expression of the two-points expected improvement was obtained in [Ginsbourger et al., C2007]. It was used to draw the plots of FIG. 1.22 which show, on a simple unidimensional linear function, that maximizing EI and 2-EI does not yield the same points: the two maximizers of 2-EI, which are shown on the EI lower left curve by dashed lines, are in high EI regions, but none of them is at the EI optimum. Furthermore, the two 2-EI optima need to be away from each other to collectively maximize the expected improvement.

The analytical calculation of q -EI for q greater than two gives a complex expres-

sion that depends on q -dimensional Gaussian cumulative density functions whose computation is necessarily approximated as soon as q is larger than four. Therefore, when q is larger than two, q -EI is estimated by Monte Carlo simulations:

```

For  $i = 1, n_{\text{sim}}$  do,
  instantiate  $f_i(x)|F(\mathbf{x}) = \mathbf{f}$ 
  calculate  $I_i(x^{m+1}, \dots, x^{m+q}) =$ 
     $(\min(\mathbf{f}) - \min(f_i(x^{m+1}), \dots, f_i(x^{m+q})))^+ |F(\mathbf{x}) = \mathbf{f}$ 
End For
 $qEI_{\text{sim}} = \frac{1}{n_{\text{sim}}} \sum_{i=1}^{n_{\text{sim}}} I_i(x^{m+1}, \dots, x^{m+q})$ 

```

The central limit theorem can then be invoked to control the precision of qEI_{sim} by changing n_{sim} . Like the SUR method, the Monte Carlo estimation of q -EI requires simulating the conditioned Gaussian process $F(x)|F(\mathbf{x}) = \mathbf{f}$ on a grid of points, which will inevitably become costly as search space dimension, n , increases.

Approximated q -EI maximization Ideally, one would like to maximize the q -points expected improvement,

$$(x^{*m+1}, \dots, x^{*m+q}) = \arg \max_{\mathbf{x} \in \mathcal{S}^q} EI(x^{m+1}, \dots, x^{m+q}) \quad (1.40)$$

but the problem becomes computationally intractable as n and q grow: there are $n \times q$ unknowns and the Monte Carlo approximation to q -EI is expensive when n increases. Two heuristics for approximating the solution of the q -EI maximization were proposed in [Ginsbourger et al., C2007]. Both heuristics, the *constant liar* and the *kriging believer*, boil down to solving a series of modified EI maximizations:

```

 $x^{m+1} = \arg \max_{x \in \mathcal{S}} \mathbb{E}[(\min(\mathbf{f}) - F(x))^+ |F(\mathbf{x}) = \mathbf{f}]$ 
For  $i = 2, q$  do,
   $x^{m+i} = \arg \max_{x \in \mathcal{S}} \mathbb{E}[(\min(\mathbf{f}) - F(x))^+ |F(\mathbf{x}) = \mathbf{f}, F(x^{m+1}) = L_1, \dots,$ 
     $F(x^{m+i-1}) = L_{i-1}]$ 
End For

```

In the constant liar, L_i is set equal to a constant which is either $\min(\mathbf{f})$ or $\text{mean}(\mathbf{f})$ or $\max(\mathbf{f})$. Since the final goal of the procedure is to minimize f , $L_i = \max(\mathbf{f})$ generates more repulsion from already visited x^{m+i} than $L_i = \text{mean}(\mathbf{f})$ which, in turn, generates more repulsion than $L_i = \min(\mathbf{f})$. The repulsion from already visited points is synonymous of a more explorative algorithm. In the kriging believer, L_i has the value of the kriging predictor knowing $F(\mathbf{x}) = \mathbf{f}$, $F(x^{m+1}) = L_1, \dots$, $F(x^{m+i-1}) = L_{i-1}$.

Tests were performed in [Ginsbourger et al., C2007] on the two-dimensional Branin-Hoo function and on 4000 realizations of Gaussian processes with various covariance structures. The aforementioned heuristics were compared to q -point designs generated by Latin Hypercube Sampling (LHS) and random UNIFORM sampling (UNIF). FIG. 1.23 presents the results on the Branin-Hoo function. It is seen that the constant liars achieve much better multi-point expected improvements than the LHS and UNIF random designs, which confirms that these heuristics approximately optimize the q -EI. With the Gaussian processes test functions, it was observed that the min constant liar is statistically more likely to yield good results than the max constant liar but it is also more likely to fail: indeed, the max constant liar is more exploratory than the min constant liar. The experiments done with the kriging believer offer another example of No Free Lunch: The kriging believer performs well – typically better than the constant liars – when optimizing Gaussian

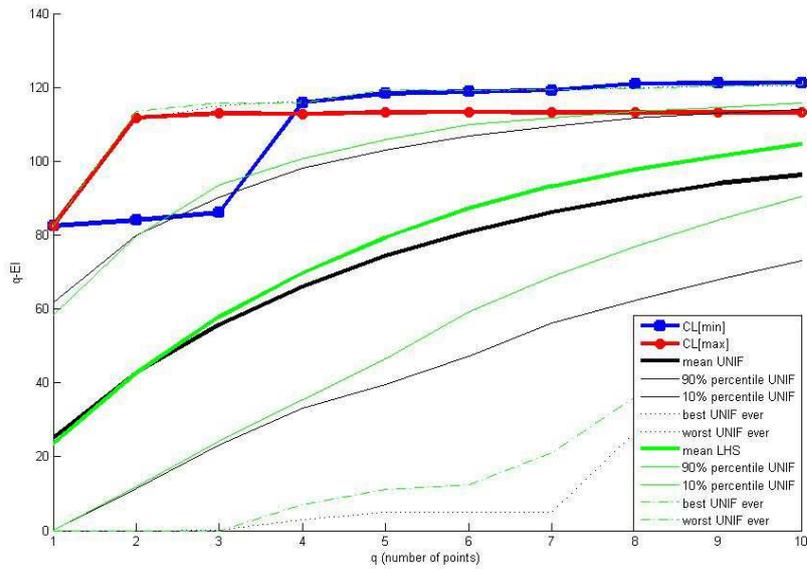


Figure 1.23: Comparison of the q -points EI achieved by constant liars (min and max), LHS and uniform random samples for the Branin-Hoo function and $q \in [1, 10]$. q -EI is estimated as qEI_{sim} ; LHS and the uniform random samples' performances are based on 2000 realizations.

processes because the objective functions match perfectly well its covariance structure; however, the kriging believer is clearly beaten by the constant liar heuristics when optimizing the Branin-Hoo function because it wrongly predicts values well below $\min(\mathbf{f})$ at the first iterations and prematurely focuses on a few regions of the search space.

Chapter 2

Specialized evolutionary optimization algorithms

2.1 How to specialize EAs

Evolutionary algorithms belong to the class of global optimization methods. The globality of the search is obtained thanks to probabilistic transitions and the use of a population, two features that, in turn, make EAs expensive optimization methods¹. Nevertheless, EAs can be adapted in many ways to the problem at hand to make them more efficient.

Customizing coding and operators. Firstly, the EA coding and variation operators can account for domain-specific knowledge. In the case of non-parametric optimization (defined in section 1.1.1), adapting coding and variation operators to the problem is the rule. For example, in [167] and [155, 156] genetic programming has been adapted to the non-parametric identification of solids behavior laws. An in depth study of codings and variation operators for topological optimization of structures can be found in [101, 100, 102, 99, 78, 79]. Our work on evolving composite laminates, which will soon be summarized (in section 2.2), provides other examples of specialized coding and variation operators.

Coupling EAs and heuristics. Secondly, the EA's structure leaves many possibilities open for coupling them with other search strategies. Such coupling is appealing with local optimization methods or improvement rules of thumb. In order to simplify the terminology, we will henceforth call any method that is supposed to improve points a "heuristic", irrespective of whether it is mathematically or empirically founded, and irrespective of whether it is local or global. The EA makes the search global, while the heuristic may increase its convergence speed. Drawing once again on the biological metaphor, if an EA is the adaptation, a heuristic could be the education. The synergy between EAs and heuristics has been acknowledged early on (see for example [39], chap. 4). It has greatly contributed to the success of the EA in practical applications since, if the analysis cost allows it, one can often improve the best known resolution strategy by coupling it with an EA.

There are three typical couplings between EAs and heuristics. The heuristic can be used to introduce high performing points in the initial population. It can also be added as a specialized mutation operator. Finally, it can improve some points of

¹Again, to give a cost estimate, no evolution is possible below 1000 calls of the objective function and constraints.

the last population. These couplings are presented in the following EA flowchart.

```

t ← 0, Population initialization [with a heuristic]
Evaluate the pop. (f)
While continue
  t ← t+1
  Selection.
  Variations (crossover, mutation, [heuristic]).
  Evaluate the children.
  Replace some parents by some children.
end while.
[heuristic applied to the best individuals]

```

While coupling an EA and a heuristic, the respective roles of the different search strategies must be balanced as a function of the total analyses budget. At one extreme, for large resources, it is possible to apply the heuristics to every population point until they converge. This idea underlies memetic evolutionary algorithms [84, 85]. The evolutionary memetic algorithm then searches in the set of the heuristic convergence points. If the heuristic is a local optimizer, this set contains the local optima. At the other extreme, the heuristic can be applied as a post-treatment of the best individual found by the EA. Between these extremes, calls to the heuristic must be controlled. The main criterion remains computation resources. There is also a risk that the heuristic induces premature convergence, for example by giving a determining advantage early in the evolution to only a few locally high performing individuals. Finally, the heuristic may make individuals less recombinable with each other. In these cases, the heuristic should be applied with caution.

2.2 Evolutionary algorithms for composite structures

2.2.1 Specializing the coding and the variation operators

We will now describe an evolutionary algorithm specifically designed to optimize the stacking sequences of composite laminates [Le Riche and Haftka, J1995]. The objective is to design structural elements made by stacking layers of unidirectional composite material. Each of the layers, or ply, is composed of fibers, typically glass or graphite fibers, aligned in a preferential direction θ , and glued together by an epoxy matrix. The structural elements are simply supported rectangular plates subjected to in-plane compressive forces N_x and N_y (cf. FIG. 2.1). Such simple elements can be analyzed by an analytical simulation (the classical lamination theory, [18, 71]). Hence, they are computationally inexpensive and it is possible to optimize them with an EA.

The composite laminate optimization problem aims at choosing the best number of layers and the best fiber orientations in each layer. Because of manufacturing constraints, possible ply orientations are taken from $\{0^\circ, +45^\circ, -45^\circ, 90^\circ\}$. In order to obtain a laminate with a symmetric behaviour, one further imposes that *i*) the stacking sequence be symmetric with respect to its middle and *ii*) that there are as many plies turned by $+\theta^\circ$ as plies turned by $-\theta^\circ$ (the laminate is then “balanced”). The symmetry of the sequence eliminates the in-plane / flexural coupling, which is written $B = 0$ with the notations of classical lamination theory (CLT) [18, 71]. The plies balance removes the in-plane extensional-shear coupling, i.e., in CLT terms, $A_{16} = A_{26} = 0$. The symmetry and balance constraints can directly be accounted

for in the coding: only one half of the plies is described, the other half being obtained by symmetry; each allele is defined as a pair of plies², $x_i = \pm \theta^\circ$, $\theta \in \{0, 45, 90\}$. Therefore, any ordered list of x_i 's describes balanced stacking sequences. If the indices denote the number of adjacent layers, and noticing that $+90^\circ$ and -90° designates the same ply, alleles take on values in $\{0_2, \pm 45, 90_2\}$. For example, the chromosome $x = [\pm 45, 90_2, \pm 45]$ corresponds to the 12 ply laminate whose stacking sequence is $[45^\circ / -45^\circ / 90^\circ / 90^\circ / +45^\circ / -45^\circ / -45^\circ / +45^\circ / 90^\circ / 90^\circ / -45^\circ / +45^\circ]$.

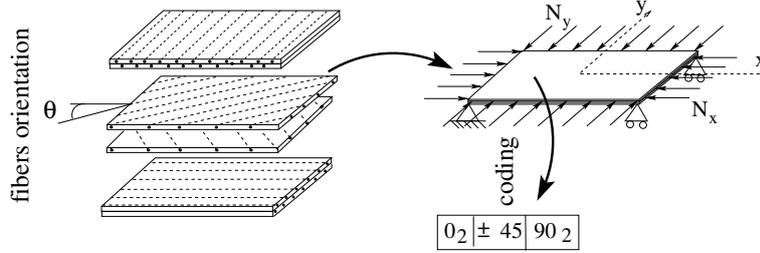


Figure 2.1: Composite plate coding, from [Le Riche and Haftka, J1993 and J1995]. A laminate is a superposition of plies in which fibers have the orientation θ . It is common to enforce a symmetry constraint in the θ 's with respect to the laminate mid-plane and a balance constraint, i.e., as many θ 's as $-\theta$'s. These constraints can be handled through the coding by describing one half of the laminate and by grouping plies in balanced alleles $\pm\theta$. The number of alleles is variable for changing the laminate thickness. The varying number of alleles makes the problem non-parametric.

The design of composite laminates can be formulated as

$$\begin{cases} \min n, \\ \text{by changing } x \in \{0_2, \pm 45, 90_2\}^n \quad (n \text{ variable}), \\ \text{such that } \lambda_{\text{bu}}(x) \geq 1, \quad \lambda_{\text{cs}}(x) \geq 1 \quad \text{and} \quad n_c(x) \leq 4. \end{cases} \quad (2.1)$$

Including n in the variables makes the problem non-parametric. λ_{bu} is the critical buckling load factor, λ_{cs} is the critical strength failure load factor (calculated with the maximum principal strains criterion) and n_c is the number of contiguous plies with the same fiber orientation. The two first constraints guarantee that the laminate does not undergo buckling or strength failure under the (N_x, N_y) loads. The upperbound on the number of contiguous plies with the same orientation is an expert rule for preventing matrix cracking. These three constraints are handled through static penalties (the subject of constraints handling and penalties has been debated in section 1.2).

Stacking sequence problems are non-linear and are likely to have local minima because of the overabundance of variables (the number of plies can be in the order of 100). In addition, the discrete and non-parametric features of \mathcal{S} are an invitation to tackle the problem with an evolutionary algorithm. Many versions of EAs for composite design have been proposed, e.g. in [Le Riche et Haftka, J1993, J1995] and [112, 113, 179]. We will now present the experiments made in [Le Riche et Haftka, J1995].

² $+\theta$ and $-\theta$ layers are thus always next to each other with this coding. This is a restriction on all possible balanced laminates since it prohibits having other layers in-between. In addition to enforcing the balance condition, this coding minimizes the laminate flexion-twist coupling. In CLT terms, it minimizes D_{16} and D_{26} . For more than 20 layers, such an assumption should not significantly deteriorate the performance of the solutions. For thin laminates, this assumption should be removed, but in this case the design space size is sufficiently small to allow enumerative strategies.

A criterion is defined to compare different EAs for stacking sequence design. It is the *cost* of the search and is equal to the average number of analyses (f and g calculations) necessary to reach a near optimal solution with 80% probability³. Near optimal solutions have the same number of plies as the global optimum x^* and $\min(\lambda_{bu}, \lambda_{cs}) > 0.999 \min(\lambda_{bu}(x^*), \lambda_{cs}(x^*))$. The global optimum was the best ever point found, resulting from thousands of optimizations. The cost was estimated by averaging 200 independent runs for three load cases (N_x, N_y) , that is, 600 independent runs were made for each EA version. The contribution of each EA component (mutation, crossover, ...) to the cost also depends on the other components because they all contribute to the overall exploration-exploitation EA balance (cf. section 1.1.3). In order to be rigorous, we therefore cannot discuss the effect of one component without accounting for its interactions with the other components. Nevertheless, testing all the combinations of the components to meta-optimize the EA (e.g., [68]) was deemed computationally too complex to be conducted without losing intuition. In particular, note that not only the EA parameters (population sizes and probabilities of the variation operators) were tuned, but also different variation operators were investigated. The experimental process chosen in [Le Riche and Haftka, J1993 and J1995] was to find, by successive enumerations on each EA parameter of each EA component, a local optimum⁴ in the space of the EA. The starting EA of this empirical meta-optimization was an elitist genetic algorithm with its archetypal parameters settings (population size $\mu = 100$, two-points crossover with $p_c = 0.6$, $p_m = 0.01$). Selection was ranked-based. The optimized EA had $\mu = 8$ individuals with new variation operators. Its search cost was 1450 analyses for designing laminates that were 48 layers thick, i.e., laminates coded by 12 alleles. The efficiency of the algorithm can be appreciated by comparing the cost to the search space size, larger than $3^{12} = 531441$: less than 0.2% were sampled on average.

Composite laminates formae

In order to understand the laminate specialized variation operators, it is useful to have in mind the simple mechanical features on which we will rely. Remember from section 1.1.5 that formae are equivalence classes that are defined by equivalence relations. Relevant equivalence relations for composite laminates are:

- laminate A [*has fewer plies than*] / [*has as many plies as*] / [*has more plies than*] laminate B.
- laminate A [*is made of the same layers as*] laminate B, irrespective of the layers' relative positions. A weaker formulation of this relation is [*has the same in-plane stiffness as*], where the in-plane stiffness designates the A matrix of classical lamination theory. For constant material laminates, an equivalent statement is [*has the same extensional lamination parameters V as*], see [Grosset et al., J2006] and section 2.2.2.
- laminate A [*has the same layer(s) at position(s) ... as*] laminate B. For the stacking sequence coding introduced and n fixed, this equivalence relation defines schemata (see section 1.1.4). For example, [$* \dots * 90_2$] is the order 1 schema (subset) of all balanced and symmetric laminates that have a 90_2 stack adjacent to their midplane. In the Appendix of [Le Riche, T1994], a strong correlation was observed in the case of composite laminates between order 1

³Mathematically, it is the 80% percentile of the random variable “number of analyses to find a near optimal solution”.

⁴This is a local optimum only for searches that vary one variable at a time. It may not be a local optimum if more than one parameter is changed simultaneously.

Parent 1 :	$[x_1^1 \ x_2^1 \ x_3^1 \ x_4^1 \ \square \ \square \ \square]$
Parent 2 :	$[x_1^2 \ x_2^2 \ x_3^2 \ x_4^2 \ x_5^2 \ x_6^2 \ x_7^2]$
X1-thin, $ic = 2$:	$[x_1^1 \ x_2^1 \ x_3^2 \ x_4^2 \ x_5^2 \ x_6^2 \ x_7^2]$
X1-thick, $ic = 5$:	$[x_1^1 \ x_2^1 \ x_3^1 \ x_4^1 \ \square \ x_6^2 \ x_7^2]$
i.e.,	$[x_1^1 \ x_2^1 \ x_3^1 \ x_4^1 \ x_5^2 \ x_6^2 \ x_7^2]$
X2-thin, $ic = 1$ and 3 :	$[x_1^1 \ x_2^2 \ x_3^2 \ x_4^1]$
X2-thick, $ic = 1$ and 6 :	$[x_1^1 \ x_2^2 \ x_3^2 \ x_4^2 \ x_5^2 \ x_6^2]$

Figure 2.2: Examples of crossovers for composite laminates. ic designates the breaking points' indices.

schemata average performance and problem easiness. Also, a strong correlation was seen between order 1 schemata performance variance and problem difficulty.

Crossover

Five crossovers have been compared, X1-thin, X1-thick, X2-thin, X2-thick and uniform crossover. These crossovers require two parents to make one offspring. If the two parents do not have the same thickness, the chromosome of the thinner parent is patched with “void” alleles, \square , until it has the same size as the thicker parent. Whenever a void appears inside a chromosome, it is deleted and the stacking sequence is compacted. The X1-... crossover series has one breaking point. X1-thin has its breaking point occurring in the filled part of the chromosome of the thinnest laminate. Hence, the child laminate always has the thickness of one of its parents. X1-thick takes its breaking point in the filled part of the chromosome of the thicker parent. The produced laminate can have any thickness between and including those of the parents. The X2-... series of crossovers follows the same logic with two breaking points. Uniform crossover has already been described in section 1.1.2. Crossover examples can be found in FIG. 2.2.

The better performing operator for designing laminates is X1-thick. Contrary to X...-thin crossovers, it properly mixes the parents' thicknesses since it is able to produce any intermediate thickness between those of the parents. On top of that, it benefits from having only one breaking point because it transmits large parts of the parents' sequences unchanged. This last property is an advantage because, as will soon be seen, intensive use is made of the permutation operator. Without permutation, it is likely that more sequence mixing would have been demanded from the crossover.

Mutation(s)

Three local perturbations of the stacking sequence can be imagined: a change of ply orientation, a change of ply position in the sequence, and a change of the number of plies (or laminate thickness). It is important to isolate each of these mutations in order to best tune the EA. As a demonstration, let us consider the counterexample of directly applying a standard EA to the design of composite laminates. One would resort to the discrete mutation of equation (1.8) with the coding $x_i \in \{0_2, \pm 45, 90_2, \square\}$, $i = 1, n$, n fixed a priori (parametric framework). If all

alleles have the same appearance probability ($1/(A - 1)$), the three aforementioned mutation events have implicitly varying occurrence probabilities that are functions of n and the laminate thickness. For example, the thinner a laminate is, the more \square 's its chromosome has, and the higher its chances of becoming thicker by mutation (mutation of a \square into a full allele, 0_2 , ± 45 , or 90_2).

For stacking sequence design, the mutation is therefore decomposed into three operators: ply orientation variation, ply order variation and thickness variation.

Ply orientation variation was trivially implemented by changing a full allele (0_2 , ± 45 or 90_2) into another full allele with a given probability. An optimal static probability of 0.01 (1%) per allele was found in the context of these experiments.

Ply order variation is given the name *permutation* operator. With a probability p_p per individual, it exchanges the positions of two alleles (each allele codes a balanced stack of two plies):

$$\begin{array}{l} \text{Before permutation : } [x_1 \ x_2 \ x_3 \ x_4 \ \dots \ x_n] \\ \text{Permutation of 2 and 4 : } [x_1 \ x_4 \ x_3 \ x_2 \ \dots \ x_n] \end{array}$$

The permutation operator makes a crucial contribution to an efficient evolution of the stacking sequences. It is applied to every child sequence, $p_p = 1$. Permutation has the advantage of exploring sequences that have new numbers of contiguous plies (n_c in equation (2.1)) and new critical buckling loads (λ_{bu}), while keeping the strength of the laminate λ_{cs} and the number of plies n unchanged. In terms of optimization criteria landscapes, permutation permits movements along the contour lines⁵ of n and λ_{cs} . Once the constraint on buckling is satisfied, permutation allows intensive exploration of the other criteria, while remaining in the feasible domain for λ_{cs} .

Thickness variation was performed with two operators. The first, and most straightforward, thickness mutation adds or deletes a two-ply stack anywhere in the sequence with a given probability per individual. Empirically, average values of 0.05 (5%) of adding and, independently, deleting, were found to be optimal.

In a second phase, such blind thickness variation was complemented by a *scaling mutation* operator. When a laminate is subjected to scaling mutation, its buckling and strength critical load factors, λ_{bu} and λ_{cs} , are calculated. Use is then made of the scaling relations

$$\lambda_{cs} \propto h \quad , \quad \lambda_{bu} \propto h^3$$

that come from strength of materials. h is the plate thickness. New “buckling” and “strength thicknesses” can be defined using these proportional relations in order for the corresponding load factors to equal one,

$$\begin{aligned} h_{bu} &= h/\sqrt[3]{\lambda_{bu}} \\ h_{cs} &= h/\lambda_{cs} \end{aligned}$$

The new thickness is the largest of h_{bu} and h_{cs} rounded to a multiple of the basic stack of four-ply thickness⁶. Stacks deletions or additions are made so as to reach the new thickness.

By lowering the rate of blind thickness variations to 1% and compensating by applying scaling mutation to 10% of the children, the price of the search was further decreased to 1310 analyses.

⁵The term “contour line” is an abuse of the language in the current discrete space, but we think it helps to visualize the explanation.

⁶A stack of four plies is the allele unit thickness here because of the balance and symmetry handling.

2.2.2 The Double Distribution Optimization Algorithm

The Double Distribution Optimization Algorithm (DDOA) is a particular Estimation of Distribution Algorithms (EDAs) which have already been discussed in section 1.1.6. DDOA was introduced in [Grosset et al., J2006] and is an example of the specialization of an evolutionary algorithm in the design of composite laminates. We would like to mention that the ideas arising from the DDOA algorithm can easily be generalized to apply to many engineering optimization problems.

General presentation of the algorithm

The general principle of the DDOA is to estimate the distribution $p^t(x)$ of high performing points from two distributions, firstly, $p_x^t(x)$, a function of primal x variables and secondly, $p_v^t(v)$, a function of auxiliary v variables.

The auxiliary variables v depend on x through $v(x)$. They provide a way to incorporate a priori knowledge into the search. Auxiliary variables can be high level information on the physics of the considered point x . For example, in mechanics, the v 's can be flexural rigidities, masses or moments of inertia, while the x 's are the structural element details. The v 's could also be drag coefficients in aerodynamics or permeabilities in the study of porous media. Note that in these examples, the v 's are integral quantities of the x 's and their calculation is computationally much simpler than that of a full simulation (a calculation of f and g). The auxiliary variables sometimes provide complete information, that is, f and g are functions of the v 's only. But the DDOA is still applicable when the v 's supply only partial information, i.e., f and g are functions of both v and x . Note that when the v 's are integral quantities of the x 's, the inverse map, $v = v^{-1}(v)$ does not exist.

```

t ← 0.
Initialize  $p_x^t$  and  $p_v^t$ .
Do while continue( $\mathcal{P}, t$ )
  t ← t + 1
  Sampling:
    create  $\lambda$  target  $v$ 's from  $p_v^t$ 
    and  $\nu \geq \lambda$  candidate  $x$ 's from  $p_x^t$ ,
    calculate the  $\nu$   $v(x)$ 's,
    keep the  $\lambda$  candidates closest to the targets.
  (Apply mutation.)
  Evaluate  $f$  (and  $g$ ) of the  $\lambda$  children.
  Select  $\mu < \lambda$  points based on performance
  (the new population  $\mathcal{P}$ ).
  Re-estimate  $p_x^t$  and  $p_v^t$  from  $\mathcal{P}$ .
End.

```

Figure 2.3: Generic flowchart of a DDOA algorithm.

The flowchart of the DDOA algorithm is shown in FIG. 2.3. At each iteration t , $\nu \geq \lambda$ candidate points are created in the x space by sampling from p_x^t . Independently, λ target points are created in the v space by sampling from p_v^t . The auxiliary variables v of the candidates are calculated. The candidate points whose v 's are closest to the target points are selected (no copies) and constitute the next population of children. The rest of the algorithm is the usual EDA except that two distributions are estimated instead of one. Note that the DDOA may be applied even if f and g are not completely determined by the auxiliary variables because

all λ children have their primal variables x defined (they have first been sampled in the primal space).

Although it is not standard, we find it necessary to complement EDAs with a mutation in order for them to perform well. Hence, a mutation instruction figures in the flowchart. The importance of mutation in EDAs was observed experimentally during Laurent Grosset's PhD and can be understood as follows: EDAs without mutation do not work well because their distributions are static pictures of already known good regions of \mathcal{S} . In contrast to learning high performance positions in \mathcal{S} , as EDAs do, the state-of-the-art CMA-ES learns successful steps. EDA's distributions play the same role as the crossover operator: they are an exploitation mechanism that infers new points from a population of already known good points. But, without mutation, an exploration mechanism is missing in EDAs.

An analytical expression of the distribution p^t sampled by the DDOA algorithm is not known to date. It is a function of p_x^t, p_v^t and is parameterized by ν . When $\nu = \lambda$, all the candidate points are kept so that $p^t \equiv p_x^t$. When $\nu \rightarrow \infty$ and all points in the v space have non zero probability $p_v^t(v)$, some candidate points are arbitrarily close to the target points and only the auxiliary distribution drives the sample, $p^t \equiv p_v^t$. For $\nu > \lambda$, p^t is an unknown compromise⁷ between p_x^t and p_v^t .

Two advantages are expected using the DDOA method:

i) the functional expression of p_x^t and p_v^t can be very simple, yet the sampled distribution p^t accounts for variable couplings. In [Grosset et al., J2006], p_x^t only describes independent variables,

$$p_x^t(x) = \prod_{i=1}^n p_i(x_i) \quad (2.2)$$

but the dependencies are reintroduced in p^t through p_v^t since $v(x)$ is a function of many of the x_i 's. Simple distributions are not flexible. As already discussed in section 1.1.6, this is advantageous because simple distributions need few data points to be identified. This means that small population sizes can be used, which is an important condition in making EAs less expensive.

ii) The auxiliary variables are a way to introduce high level information in the search at a negligible computational cost (the v calculation time should be much smaller than those of f and g). Typically, the dimension of the auxiliary space is smaller than that of the primal space. One half of the DDOA thus evolves in a space of reduced dimensions.

Application to composite laminates

In [Grosset et al., J2006], the DDOA is applied to the optimization of composite laminate stacking sequences. The variables are defined as in section 2.2.1: x_i is the fiber orientation of the i -th stack of two plies of a symmetric laminate, i.e., the laminate's stacking sequence is $[\pm x_n^\circ / \dots / \pm x_1^\circ]_s$. The fiber angles are taken in $\{0^\circ, \pm 22.5^\circ, \pm 45^\circ, \pm 67.5^\circ, 90^\circ\}$. The four auxiliary variables are extensional and flexural lamination parameters,

$$\begin{aligned} V_{\{1,3\}}^* &= \frac{1}{n} \sum_{k=1}^n \{\cos 2x_k, \cos 4x_k\}, \\ W_{\{1,3\}}^* &= \frac{1}{n^3} \sum_{k=1}^n [(n-k+1)^3 - (n-k)^3] \{\cos 2x_k, \cos 4x_k\}, \end{aligned} \quad (2.3)$$

respectively [186, 137]. Within the assumptions of the classical lamination theory, for given fiber and matrix materials and symmetric and balanced laminates, these four lamination parameters fully determine the overall extensional and flexural behaviour of the plate (i.e., the A and D matrices). This property is an important

⁷A linear approximation could be $p^t(x) = \alpha(\nu)p_x^t(x) + (1 - \alpha(\nu))p_v^t(v(x))$ where $0 \leq \alpha(\nu) \leq 1$, $\alpha(\nu) \rightarrow 1^-$ when $\nu \rightarrow \lambda^+$, and $\alpha(\nu) \rightarrow 0^+$ when $\nu \rightarrow \infty$.

source of inspiration in laminate design because there can be some 100 ply angles, while the number of lamination parameters remains constant at four [Harrison et al., C1995],[71, 185].

The distribution p_x^t assumes independent variables (equation (2.2)) and the $p_i(x_i = \theta)$ are simply the frequency of the appearance of θ at the i th variable in the last population. Although the lamination parameters take discrete values that follow bounded regular geometric patterns [186, 137], they are described in the DDOA by an unbounded continuous distribution. Indeed, they can take a large number of well spread values in the (V^*, W^*) space and tests have shown that accounting for the bounds slowed down the DDOA [Grosset et al., J2006]. The density in the auxiliary space is a sum of Gaussian kernels,

$$p_v^t(v) = \frac{1}{\mu} \sum_{i=1}^{\mu} K(v - v^i) \quad , \quad K(u) = \frac{1}{(2\pi)^{d/2} \sigma^d} \exp\left(-\frac{u^T u}{\sigma^2}\right) \quad (2.4)$$

where v^i is the vector of lamination parameters of an individual in the population, d is the number of lamination parameters taken into account and σ is a bandwidth (adjusted by maximum likelihood).

Three problems were investigated. The in-plane stiffness problem is to maximize the transverse in-plane stiffness with bounds on the effective Poisson's ratio,

$$\begin{cases} \max_x A_{22}(x) \\ \text{such that } \nu_{\text{low}} \leq \nu_{\text{eff}}(x) \leq \nu_{\text{upp}} . \end{cases} \quad (2.5)$$

The two extensional lamination parameters are sufficient to fully describe the solutions because A_{22} and ν_{eff} are only functions of V_1^* and V_3^* ($d = 2$ in this case). The extensional/flexural problem is to minimize the longitudinal coefficient of thermal expansion with a lower bound on the first natural frequency,

$$\begin{cases} \min_x |\bar{\alpha}_x(x)| \\ \text{such that } \omega_1(x) \geq \omega_{\text{min}} . \end{cases} \quad (2.6)$$

In this problem, the four lamination parameters are necessary to fully describe the solution because $\bar{\alpha}_x$ is a function of V_1^* and V_3^* , and ω_1 of W_1^* and W_3^* . The last problem is to maximize the laminate strength. The strength load factor is calculated using a first-ply maximum strain failure criterion,

$$\max_x \lambda_s(x) = \min_{k=1, \dots, n} \left\{ \min \left[\max \left(\frac{\epsilon_1^t}{\epsilon_1(k)}, \frac{-\epsilon_1^c}{\epsilon_1(k)} \right), \max \left(\frac{-\epsilon_2^c}{\epsilon_2(k)}, \frac{\epsilon_2^t}{\epsilon_2(k)} \right), \frac{\gamma_{12}^{\text{ult}}}{|\gamma_{12}(k)|} \right] \right\} \quad (2.7)$$

For this strength definition, the extensional lamination parameters provide only partial information: all angles need to be individually known to calculate ply failures.

The DDOA is compared to a genetic algorithm with a two-points crossover and $p_c = 0.6$, and to a UMDA. All algorithms have a population size of $\mu = 30$ individuals. UMDA and DDOA have $\lambda = 60$ children. DDOA has $\nu = 120$. Each of these algorithms uses a local mutation at a rate $p_m = 0.02$ per allele. In essence, these algorithms are the same with the exception of the exploitation mechanism (crossover or p^t) which is therefore the studied component. The algorithms' performances are estimated by their reliability, defined as the probability of finding the optimum after a certain number of analyses. Reliability is obtained from 50 independent optimizations. For all design problems, the improvement brought by the DDOA over the GA and the UMDA can be observed. It becomes particularly clear when the number of dimensions increases. An example result is given in FIG. 2.4. For more details, the reader is referred to [Grosset et al., J2006].

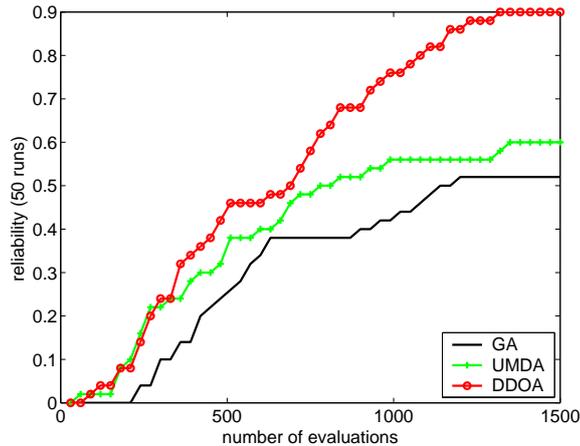


Figure 2.4: Reliability of the GA, UMDA and DDOA on the extensional/flexural design problem, $n = 12$.

2.3 Evolutionary algorithms for shape optimization

An example where an EA is coupled to a heuristic for shape optimization was developed in [Le Riche and Cailletaud, J1998]. The optimization goal is to minimize a fan disk volume, V , with bounds on Von Mises and hoop stresses, σ_{vm} and $\sigma_{\theta\theta}$ respectively. These constraints are handled by a static linear penalty, so that the final function to minimize is⁸

$$V + p \max \left[\max_{\text{fan}} (\sigma_{vm} - \sigma_{vm}^{\text{allow}}, \sigma_{\theta\theta} - \sigma_{\theta\theta}^{\text{allow}}), 0 \right]$$

where p is a penalty parameter (set to 0.5 in the experiments described below).

The shape is parameterized by B-splines. The optimization variables, $x \in \mathbb{R}^n$, are the positions of the B-splines control points. In order to encode realistic shapes (prevent B-splines loops and control overall fan dimensions), each control point is constrained to remain inside a convex polygon. The fan shape coding is depicted in FIG. 2.5. The disk and its loading are assumed to be axisymmetric. Once the positions of the B-splines' control points are set, the contours of a radial cross-section of the disk are known. A free mesh generator is called and the stresses are calculated by finite elements. The disk is subjected to centrifugal forces and to F_{blades} . F_{blades} replaces the blades' centrifugal forces that would be passed on the disk. Other boundary conditions are shown in FIG. 2.5. The mesh contains around 2000 degrees of freedom and the material is supposed to be elastic linear. The Zebulon finite element software was used for this study [44]. A complete fan analysis, including B-splines and mesh generation, finite element calculations and post-processing, took about 10 seconds CPU at the time of the study (in 1997).

The shape improvement heuristic devised in [Le Riche and Cailletaud, J1998] is called *penalized generalized biological growth*. Biological growth [131, 144, 50] locally adds material at the solid surface if neighboring Von Mises stresses exceed an allowable value. In addition, generalized biological growth locally removes material if Von Mises stresses are below the allowable value. Penalized biological

⁸The function that was really minimized also had normalization constants for the stresses and the volume. To simplify the text, we do not reproduce them here but they are useful in setting p . See [Le Riche et Cailletaud, J1998].

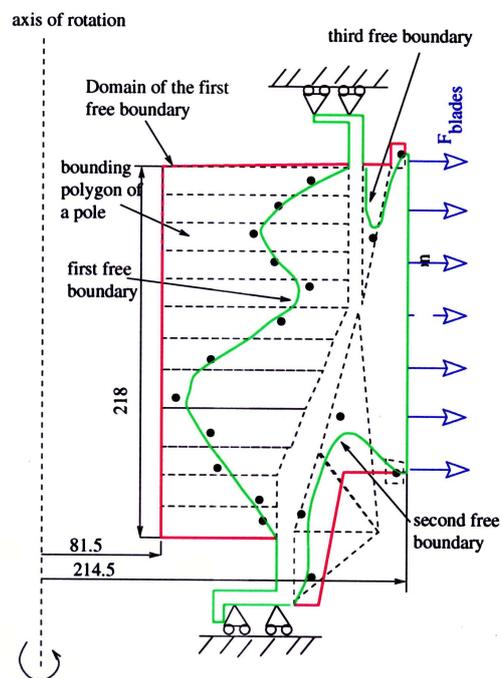


Figure 2.5: Coding of a blade fan disk shape with B-splines. The fan is axisymmetric, subjected to centrifugal forces and to F_{blades} . The B-splines control points remain inside convex polygons. (From [Le Riche and Caillaud, J1998].)

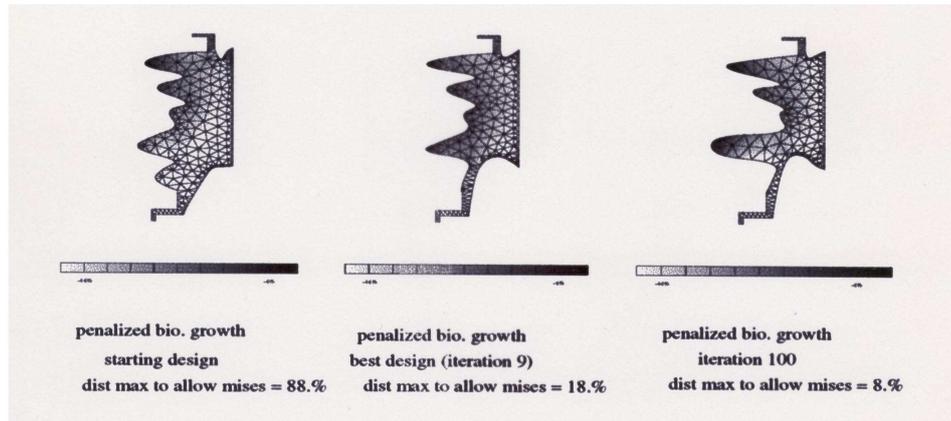


Figure 2.6: Penalized generalized biological growth heuristic for improving the shape of a fan disk. The shape is parameterized by B-splines (from [Le Riche and Cailletaud, J1998])

growth controls how much material is removed depending not only on the local but also on the global stress state. FIG. 2.6 illustrates the effect of the penalized generalized biological growth of a fan disk.

In a coupled biological growth - evolutionary approach, new designs are created either by the usual crossover and mutation operators or by the growth heuristic: biological growth needs the stress state of the design and cannot directly follow the variation operators. Several strategies for coupling the shape improvement heuristic and the evolutionary algorithm are compared in [Le Riche and Cailletaud, J1998]: biological growth is applied to the initial population, to the worst individual of the current population or to the entire population. The frequency of the application of the heuristic, i.e., the number of designs created by biological growth over the total number of calculated fans is varied between 1 and 1/10.

Experiments were carried out with a steady-state genetic algorithm composed of a flat crossover (applied at rate $p_c = 1$) and a fixed anisotropic Gaussian mutation ($p_m = 0.1$ per variable and $\sigma_i^2 = (x_i^{\max} - x_i^{\min})^2/16$). There were 50 individuals per population and the search stopped after 2000 analyses (i.e., 5.5 hours CPU). Due to the length of a single optimization, each strategy was tested by repeating the search only three times. The tests show that it is better to apply the shape improvement heuristic to the entire population at a frequency of 1/10, i.e., 9 individuals are generated by the variation operators for 1 by generalized penalized biological growth. This strategy outperforms an EA alone, an EA with more mutation ($p_m = 0.2$), a random search and the heuristic alone. The optimum design found is reproduced in FIG. 2.7.

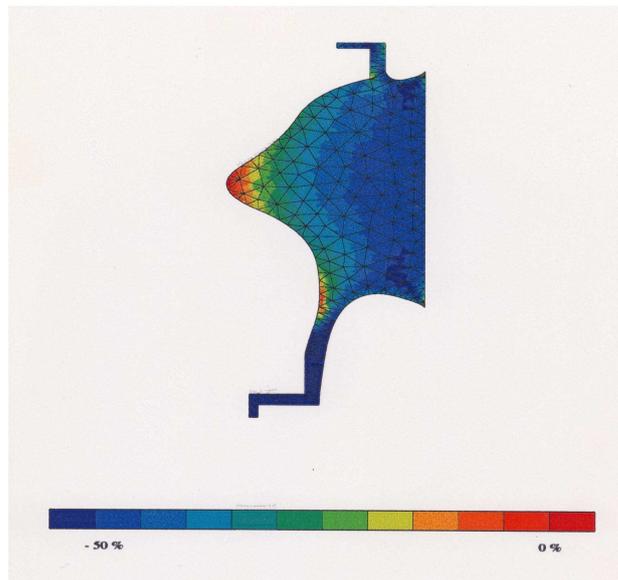


Figure 2.7: Optimum fan disk shape obtained by coupling an EA with a shape improvement heuristic, the penalized generalized biological growth. The material is off center with respect to the rotation axis in order to increase disk flexural stiffness. Stress constraints are violated by less than 1%. Cf. [Le Riche and Cailletaud, J1998].

Chapter 3

Perspectives

Optimization and uncertainties. It is important for researchers in engineering optimization to realize that the simulators, y , they are working with are imperfect representations of the actual systems. These imperfections can often be translated into uncertain model parameters which should be accounted for in the optimization [104, 180]. Some of these uncertainties are of a deterministic nature, like uncontrolled experimental parameters. Other uncertainties are random in nature, for example wind loadings. Therefore, designing and identifying in the presence of random uncertainties are two perspectives to our work which are discussed in sections 3.1 and 3.2.

Accounting for simulation cost. We have already mentioned in the “bottle-necks” introductory part to this manuscript that the simulation cost is, and will remain, a determining factor of any optimization approach. A large number of recent optimization studies address this issue by replacing some of the high-fidelity numerical simulations by computationally less expensive physically-based or learned models [150, 177, 74]. Another of our perspectives, further described in section 3.3, will be to explicitly include the cost of the simulations in the formulation of the optimization problem.

Distributed optimization. In the last 20 years, the total high performance computing capacity available worldwide has doubled each year. Starting from a teraflops in 1994, it has reached a petaflops in 2005. At the end of 2007, it is estimated at 7 petaflops, 0.5 of which belongs to the current most powerful computer. This increase in computing capacity is not only attributed to large integrated systems, but also to grid-computing, which allows the creation of virtual super-computers from a network of loosely-coupled computers (e.g., the European EGEE grid, [53]).

Distributed computing technologies are a way of addressing the simulation cost barrier. They are rapidly adopted by computationally demanding domains (climatology, fluid mechanics, genetics, astrophysics). Engineering optimization is also turning towards distributed computation [190, 199, 105, 143, 193], not only as a remedy to the computational cost of repeated simulations, but also because the design of complex systems typically involves many experts, softwares and computers that are geographically distant from each other [67]. However, in design, the change to distributed computing is still at an experimental stage. Most optimization methods have been devised in a sequential frame of mind. I think it is now important for optimization researchers to envision asynchronous methods. This is one of the conditions for having methods that scale well with the number of computing nodes. Other obstacles to the development of distributed optimization are the complex-

$$(x, U) \longrightarrow \theta \longrightarrow \boxed{\text{numerical model}} \longrightarrow y$$

Figure 3.1: Parameters definitions: x are the design variables, U the random variables, θ the numerical model parameters and y its output.

ity and instability of software (middleware) components¹ and, in multidisciplinary applications, the amount of data exchanged between nodes. The perspectives that will be described in Sections 3.1, 3.2 and 3.3 will favor methods which are prone to being distributed.

3.1 Optimum design accounting for uncertainties

There are two principal types of uncertainties in optimization²: parametric uncertainty is due to randomness in some of the parameters of a numerical model; model uncertainty refers to the approximations made when describing a physical system with a numerical model. For example, representing a structure with an elastic material when a non-linear material behavior is expected introduces a model uncertainty. Reducing the number of elements of a finite elements model is another example of model uncertainty introduction. In the sequel, we focus on parametric uncertainties.

Let y be a numerical model whose parameters are grouped in the vector θ . θ depends on the deterministic design parameters $x \in \mathcal{S}$ and on the random parameters U taken in a probability space. (See FIG. 3.1.) Note that, as usual, capital letters will denote random variables and lower cases realizations of these variables. The model is $y(\theta(x, U))$ or, in shorthand notation, $y(x, U)$. A typical example in design is when dimensions are supposed to be perturbed by additive Gaussian manufacturing errors:

$$\begin{aligned} \theta &\equiv \text{dimension} \\ x &= (m, \sigma) = (x_1, x_2) \\ U &\sim \mathcal{N}(0, 1) \\ \theta &= x_1 + x_2 U \end{aligned}$$

m is the average dimension and σ is associated to the product quality class. In this example the noise is controlled through x_2 . In other cases, the noise is not directly controlled, e.g. when U represents a random aerodynamic load.

The uncertainty introduced by U propagates through the optimization criteria f and g which become random functions,

$$\begin{aligned} F(x) &\equiv f(x, U) \equiv f(x, U, y(x, U)) \\ G(x) &\equiv g(x, U) \equiv g(x, U, y(x, U)) \end{aligned}$$

In robust and reliability based design, one looks for a unique x (a design) which guarantees a certain level of performance knowing the law followed by U (but one does not consider particular realizations u). The deterministic optimization problem (2) is reformulated using statistical measures of the criteria. A particularly relevant formulation is

$$\begin{cases} \min_{x \in \mathcal{S}} p_\alpha(F(x)|G(x) \leq 0) \\ \text{such that } \text{Prob}(G(x) \leq 0) > \beta \end{cases} \quad (3.1)$$

¹Luckily, many projects and products address the middleware issue, e.g., ProActive [89].

²The word ‘‘uncertainty’’ may also be used to describe errors generated by computer finite arithmetics. The question of whether a local solution is global or not is sometimes called ‘‘the fundamental uncertainty’’.

where p_α is the α -th percentile of the performance of feasible designs, hence the conditioning. β is a lower bound on the probability that *all* constraints are satisfied. It should be noted that this formulation accounts for all statistical dependencies between the f 's and the g 's³.

Although the criteria in (3.1) have been made deterministic by averaging out the uncertainties in the percentiles and the probabilities, the problem is much more difficult than the original (2): estimating percentiles and probabilities is a complex task. If the statistics are empirically estimated by Monte Carlo simulations with N analyses each, the cost of the optimization is multiplied by N . If one remembers that the computational cost is already a barrier in deterministic optimization, it is clear that including simple Monte Carlo simulations inside an optimization loop becomes rapidly out of computational reach. Therefore, optimizing while accounting for uncertainties was approached in the past

1. by replacing the probabilities by reliability indices, i.e., distances to the most probable failure points (e.g., FORM and SORM methods, [133, 154]),
2. by constructing metamodels of $f(x, u)$ (idem with g) and subsequently estimating statistical measures with them (either through Monte Carlo [118, 15, 124] or analytically [28, 51]),
3. by constructing metamodels of the statistical measures in \mathcal{S} (typically the average and the variance of $f(x, U)$ and $g(x, U)$) [189, 15],
4. by approximating the random processes $F(x)$ and $G(x)$ by polynomial chaos expansions [60, 109].

The problem (3.1) is of contemporary interest because there is a tendency to account for more and more uncertainty sources at early design stages (variations in boundary conditions, variations in material parameters, manufacturing tolerances). Moreover, probability measures are more accurate than safety factors and might allow structural mass savings [97]. We propose to study methods for solving (3.1) which are based on the following principles:

- Uncertainty is propagated by improved Monte Carlo methods; importance sampling [175], recycling of the simulations⁴, common random numbers [104]. An important feature for later use in optimization algorithms is to have confidence measures for p_α and the probability of being feasible⁵. With Monte Carlo simulations, these confidence measures are functions of the number of simulations at each x .
- Early in the search, the statistical measures of f and g do not need to be accurate. They only need to allow progress towards high performing regions of \mathcal{S} . Therefore, at the beginning of the search, the empirical estimates of the performance statistics can be based on fewer analyses than later. The number of analyses allocated to each Monte Carlo simulation can be controlled by progressively increasing the confidence levels during the optimization.
- The optimizer that decides which x 's are analyzed should be able to work with noisy functions since empirical estimates of performance statistics are noisy. Evolutionary algorithms are good candidates for such a task.

³Statistical dependency seems to be often neglected in the literature. For example, in [108, 59], the probabilities of satisfying the constraints are handled separately which guarantees that a point is feasible in $\beta\%$ of the cases only if the constraints are statistically independent.

⁴For example, if $\theta = x + U$, a simulation $y(x_1 + u_1)$ done at x_1 may be re-used as a Monte Carlo realization at x_2 , $y(x_2 + x_1 - x_2 + u_1)$, i.e., $u_2 = x_1 - x_2 + u_1$.

⁵More precisely, for rank-based optimization algorithms such as evolutionary algorithms and pattern-search methods, confidence levels on the relative ranks of points x in \mathcal{S} are needed, e.g., $\text{Prob}(p_\alpha(x_1) < p_\alpha(x_2))$.

- The computational burden of optimizing while accounting for uncertainties is due to the imbricated iterations of the optimization and the statistics. Methods that both optimize and estimate statistical measures at the same time were proposed. To our knowledge, they all focus on optimizing the expectation of $f(x, U)$: The Robbins-Monro method in stochastic approximation [180] and evolutionary strategies for robust optimization in [22]. An early try at generalizing the Robbins-Monro formula to variance, α -percentile and probability estimations can be found in [Pujol et al., J2008].
- In order to take advantage of the emerging distributed-computing technologies for tackling the computational cost barrier, methods proposed for optimizing noisy functions should be as asynchronous as possible. This argument is in favor of Monte Carlo methods for estimating the statistics.

3.2 Identification accounting for uncertainties in solid mechanics

Contrary to design, when identifying parameters from measures, realizations of the uncertainties occur before the optimization is done: boundary conditions and measures have a certain value, even if this value is not always known. Identification in the presence of uncertainties is an example of “closed-loop” problem where an optimum can be sought for each occurrence of the noise:

$$\begin{cases} \theta^*(u) = \arg \min_{\theta \in \mathcal{S}} f(\theta, u) \\ \text{such that } g(\theta^*(u), u) \leq 0 \end{cases} \quad (3.2)$$

Note that in this section about identification in solid mechanics, we temporarily change the notation for optimization variables (here identification parameters) from x to θ because x will be needed as the usual symbol for spatial positions. Typically here, θ are material parameters and f is a distance between a model, y^{mod} , and experimental outputs, y^{exp} . The solutions of problem (3.2) for all uncertainties define the random vector, $\theta^*(U)$, of all identified parameters. Today, estimating confidence intervals about $\theta^*(U)$ (e.g., the parameters one looks for are in $[\theta^{\text{low-bnd}}, \theta^{\text{upp-bnd}}]$ with 99% confidence) and, a fortiori, estimating the probability density function of $\theta^*(U)$, is becoming important in identification.

3.2.1 Identification from full-field measurements

In the last decade, progress in optical full-field measurement techniques and the appearance of affordable CCD cameras have enabled the acquisition of entire displacement and strain fields, as opposed to point measurements (e.g., by strain gages). Full-field measurement techniques are typically based on image correlation, moiré and speckle interferometry, and grid methods. They provide from 1000 to 10000 independent measurement points. These developments, combined to those in solid mechanics simulation (finite elements in particular), have opened the way to the direct identification of material parameters from structural tests (e.g., [Besson et al., J1998; Molimard et al., J2005; Silva et al., J2007-2]). The older, but still standard, identification methodologies work with specimen designed to create locally (arguably) homogeneous strain and stress fields. Global material parameters are then estimated from analytical models. Identification based on structural tests allows heterogeneous strain and stress fields. Because these fields provide more information, fewer tests are necessary and local material parameters can be estimated. Moreover, in situ identification can serve to control structural safety.

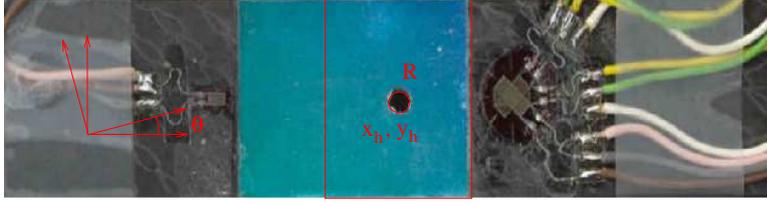


Figure 3.2: Identification of experimental parameters in addition to the material constitutive parameters. The experimental parameters are the specimen tilt angle, θ , and the hole position and radius, x_h , y_h and R , respectively. The material parameters are the four in-plane elastic moduli, E_{xx} , E_{yy} , G_{xy} , ν_{xy} . The specimen is a plate with a hole made of NC2© reinforcement from Hexcel Composite with epoxy resin manufactured in a RTM mold. The stacking sequence is $[(0/90)_3]_s$. Cf. [Molimard et al., J2005].

A lot of attention has been devoted to formulating a distance between the simulated and measured fields. Five types of distances were described in the recent survey [8] by a French “identification” working group (the CNRS GDR 2519): a least squares distance from a finite element model, the constitutive equation gap method, the virtual fields method, the equilibrium gap method and the reciprocity gap method. Contrary to the finite element technique, the last four methods are directly related to the variational formulation. They gain additional information (e.g., explicit minimizations in the constitutive equation gap) and possibilities (e.g., choose a virtual field which makes the identification insensitive to unknown boundary load distributions) at the expense of restricted application cases (e.g., the entire strain field must be known in the virtual field method) and more complex implementations. Our previous work about identification considered least squares distances to finite elements models and studied the numerical aspects of iterative identification: we addressed the computational complexity of fields comparison in [Silva et al., J2007-1], the implementation of the Levenberg-Marquardt algorithm for scaling and bounding variables [Le Riche and Guyon, T1999; Molimard and Le Riche, J2003] and the effect of distance norms on identifiability [Molimard et al., J2005].

A characteristic feature of full-field identification problems is that they are largely overdetermined: there are much more independent data points than parameters to determine. Current identification practice is still often restricted to superimposing experimental and simulated responses and minimizing the distance between them by changing material parameters. Full advantage is not taken from all information provided by the fields. In particular, full-field techniques allow to analyze possible experimental errors and estimate material confidence intervals.

Identification of experimental parameters

Because full-field identification problems are overdetermined, it is possible to add to the list of unknown constitutive parameters other parameters that describe uncertain aspects of the experimental setup. In [Molimard et al., J2005], the four in-plane elastic properties (E_{xx} , E_{yy} , G_{xy} , ν_{xy}) of a thin orthotropic composite laminate were identified from a plate-with-a-hole test. As illustrated in FIG. 3.2 the precise position of the hole, its radius and the specimen misalignment angle were added to the identified parameters. An on-going work done in G. Silva’s PhD is to systematically calculate the displacement field created by probable deterministic experimental errors. The following error sources were analyzed: mismatching of the experimental and numerical coordinate systems, rigid body motion, camera

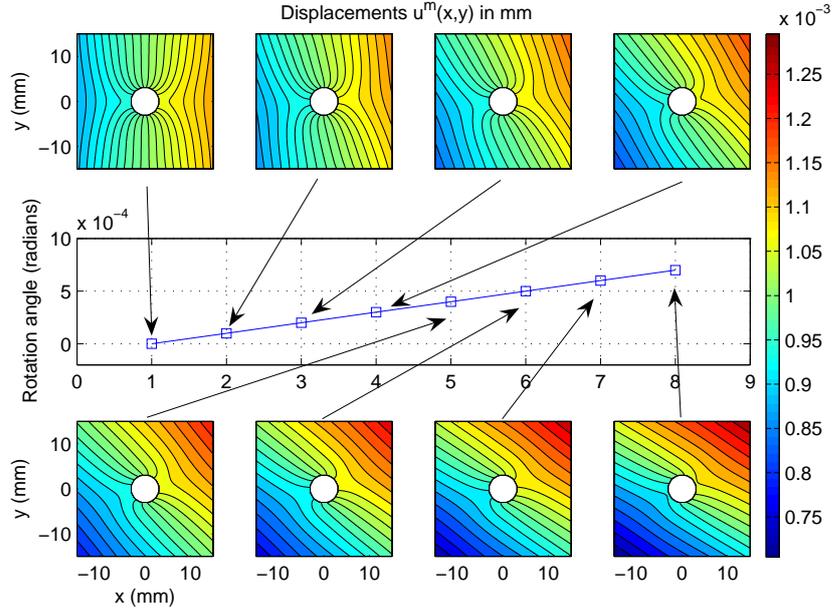


Figure 3.3: Addition of a rigid body rotation onto the u displacement field of a plate with a hole in traction (from G. Silva PhD Thesis, [176]).

misalignment and image distortion. FIG. 3.3 shows an example of rigid body rotation field superimposed onto the longitudinal displacement field of a plate with a hole.

We propose to study how the contributions of several experimental errors sources can be identified from field measurements before, at the same time as, or after the constitutive parameters. New error sources, stemming from propagating uncertainties through the optical acquisition chain and the image processing, should be accounted for. Care should be taken to discuss the identifiability, i.e., the possibility to separate the contribution of each material and error parameter.

Damage identification

Another important application of full-field identification is finding heterogeneous material properties, i.e., locally damaged areas. For linear elasticity, if the heterogeneity takes the form of a scalar field,

$$\mathbf{A}(x) = c(x)\mathbf{A}_0$$

where $\mathbf{A}(x)$ is the local elasticity Hooke tensor, the equilibrium gap method or the virtual field method should be used [30]. (In this case, note that the experimental displacement field must be known at each mesh node.) When this is not possible, the identification can be tackled from outside the simulation, through the ill-posed inclusion identification problem [166]. A first attempt at such an identification was done in G. Silva's PhD [176]: as it can be seen in FIG. 3.4, once the homogeneous plate properties have been estimated, local damage often has a local signature on the residuals map (in particular on the strain map). When such a signature is visible, the domain is partitioned along the residuals map frontiers. Next, each subpartition is assigned a different set of material parameters and the identification is repeated.

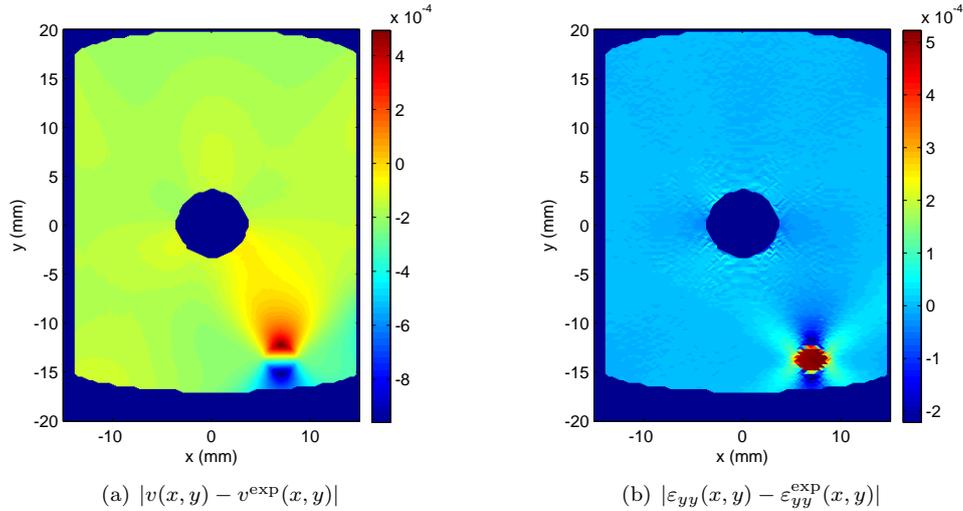


Figure 3.4: Distance maps of a plate with a hole in traction with an elliptical local material aberration. $E_1 = 19.49\text{GPa}$, $E_2 = 13.65\text{GPa}$, $G_{12} = 11.42\text{GPa}$, $\nu_{12} = 0.60$ excepted in the ellipse where $E_1 = 10.69\text{GPa}$, $E_2 = 16.69\text{GPa}$, $G_{12} = 11.06\text{GPa}$, $\nu_{12} = 0.62$. The identified homogeneous properties are $E_1 = 19.45\text{GPa}$, $E_2 = 12.11\text{GPa}$, $G_{12} = 12.49\text{GPa}$, $\nu_{12} = 0.57$. From [Silva et al., C2006].

We propose to continue this work and consider more difficult cases where strains differences do not clearly partition the domain. This identification problem includes looking for partitions with different material parameters. This is a difficult non-parametric optimization problem. It was already approached in [166] using Voronoi cells but this identification problem is so fundamental that it deserves further attention.

Identifying the distribution of parameters

An important perspective of full-field identification is to estimate, from one or a few experiments, the density of the identified parameters $\theta^*(U)$ (cf. (3.2)). (Of course, all deterministic experimental errors that can be removed before identifying the constitutive parameters should be corrected.) We consider here homogeneous material properties and note that, typically, our models (displacements and strains) are non-linear in terms of material and experimental parameters. Two approaches can be taken.

Firstly, one can linearize the field with respect to the θ parameters around a set of identified parameters. Then, under the assumption that the model is linear, the parameters distributions (or only confidence intervals) can be calculated using classical regression analysis [47]. In this case, a noise model, for example

$$y^{\text{exp}}(U) = y^{\text{mod}}(\theta^{\text{target}}) + U \quad , \quad \text{where } U \sim \mathcal{N}(0, \sigma^2 \mathbf{I}) \quad , \quad (3.3)$$

will be required. The noise model may stem from propagating uncertainties through the measurements chain or from analyzing the residuals. The parameter density obtained with the linearity assumption should be checked a posteriori. Advantage could be taken of the large number of data by repeating the identification using subsets of all measurements: the variances of the identified parameters should be larger (multiplied by about m/m' where m is the total number of measurement and m' the size of the control subsets) and the law should remain the same.

Secondly, one can stay within a non-linear regression framework and rely on re-sampling (bootstrap) strategies [52]. Resampling strategies involve computationally expensive repeated identifications. However, the identifications can (and should) be distributed among several computers. The idea is to take advantage of the growing computational capacity to estimate standard errors that are not directly derivable from theory. Once several realizations $\theta^*(u_1), \dots, \theta^*(u_m)$ of $\theta^*(U)$ have been calculated, statistics can be performed to infer confidence intervals or, if m is large enough, a density. A first trial at bootstrapping for estimating parameters standard errors was published in [Silva et al., J2007-2]. An additive noise model with stationary and independent errors such as (3.3) was assumed. In the future, a more careful analysis of the noise model will be carried out: spatial dependency of the errors U could be described by Gaussian processes whose covariance structures could be inferred by maximizing their likelihood knowing $[y^{\text{exp}}(u) - y^{\text{mod}}(\theta^*(u))]$.

3.2.2 Bayesian identification

Bayesian identification is a method for directly estimating the probability density function of θ without solving the optimization problem (3.2). As the name suggests, it is based on Bayes rule applied to continuous random vectors

$$p(\theta|y^{\text{exp}}(u)) = \frac{1}{K} p(y^{\text{exp}}(u)|\theta) p(\theta) \quad (3.4)$$

where the $p()$'s are probability density functions (pdf's) and K is a normalization constant such that the left-handside integrates to 1, as every pdf should. $p(\theta)$ is called the *prior* distribution. It is our a priori knowledge about the distribution of parameters. $p(y^{\text{exp}}(u)|\theta)$ is the probability of observing the measures $y^{\text{exp}}(u)$ when the parameters have values θ , or likelihood of θ knowing $y^{\text{exp}}(u)$. $p(\theta|y^{\text{exp}}(u))$ is the a posteriori density of θ knowing the experimental results $y^{\text{exp}}(u)$. In the following, we will shorten the notation from $y^{\text{exp}}(u)$ to y^{exp}

In [Gogu et al., J2008], advantages of Bayesian methods over other identification approaches were shown through a truss and a plate examples. FIG. 3.5 is a graphical comparison of least squares and Bayesian results for a Young's modulus identification from strain measurements on a three bar truss. There is a gradual sophistication from simple least squares (LS) to Bayesian identification:

$$\begin{aligned} \text{Simple LS;} & \quad \theta^* = \arg \min_{\theta} \frac{1}{2} (y^{\text{mod}}(\theta) - y^{\text{exp}})^T (y^{\text{mod}}(\theta) - y^{\text{exp}}) \\ \text{Generalized LS;} & \quad \theta^* = \arg \min_{\theta} \frac{1}{2} (y^{\text{mod}}(\theta) - y^{\text{exp}})^T C(\theta)^{-1} (y^{\text{mod}}(\theta) - y^{\text{exp}}) \\ \text{Max. likelihood;} & \quad \theta^* = \arg \max_{\theta} p(y^{\text{exp}}|\theta) \\ \text{Bayes;} & \quad p(\theta|y^{\text{exp}}) = \frac{1}{K} p(y^{\text{exp}}|\theta) p(\theta) \end{aligned}$$

Compared to the simple least squares, the generalized formulation appropriately handles normalization as well as measurements correlations thanks to the variance-covariance matrix $C(\theta)$. For example, if two measures are strongly correlated, their collective influence is decreased through the C weighting so that they do not hide information present in other measurements. In addition, measurements exhibiting strong deviations are underweighted. The maximum likelihood also has these advantages. In addition, it can handle non Gaussian distributions and the likelihood provides of measure of the confidence one has in the identified parameters. Bayesian identification also properly normalizes and accounts for variance-covariances of the

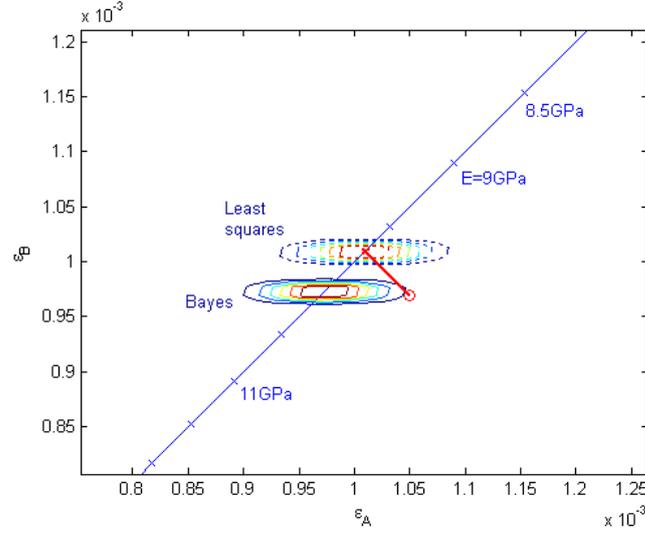


Figure 3.5: Graphical comparison of the least squares and Bayesian identifications for the three bar truss example of [Gogu et al., J2008]. Different uncertainty levels in the loads translate into different uncertainties in the bars strains. The red circle is the experimental measurement and its orthogonal projection (red cross) the least squares identified modulus. The ellipses are the contour plots of the strain distribution, $p([\varepsilon_A, \varepsilon_B]|E)$. $E = E_{LS}^*$, the least squares identified modulus, for the dashed ellipses. $E = E_{\text{bayes}}^*$, where $E_{\text{bayes}}^* = \arg \max_E p(E|\varepsilon_A, \varepsilon_B)$, for the full line ellipses. Note that the likelihood of $E = E_{\text{bayes}}^*$ is higher than that of E_{LS}^* for the measurements.

measurements. It can be applied to non-Gaussian measurements. Moreover, it incorporates previous knowledge about the parameters with the prior distribution and it yields a complete distribution of the parameters from which confidence intervals can be calculated.

In most cases, an analytical expression of the likelihood function $p(y^{\text{exp}}|\theta)$ is not known. It follows that the apparently simple expression (3.4) for Bayesian identification hides in fact a large computational cost: for each choice of θ , the pdf $p(y(U)|\theta)$ needs to be estimated by Monte Carlo sampling of the U 's in order to, next, calculate the likelihood of θ , $p(y^{\text{exp}}|\theta)$. Again, the “double loop” seems to haunt an estimation problem in the presence of uncertainties. This computational barrier is increasingly daunting with the dimension of y .

Future work on Bayesian identification will focus on finding strategies for computing the likelihood $p(y^{\text{exp}}|\theta)$. An interesting target application is to be able to carry out the Bayesian identification of homogeneous anisotropic material parameters from full-field measurements. In this case, y has dimension 1000 to 10000 ! Four research directions can be taken:

- Reducing the dimension of the original problem, for example by proper orthogonal decomposition (POD) [104, 192];
- Improving the efficiency of the Monte Carlo sampling procedure by turning to Monte Carlo Markov Chains [174];
- Replacing the original model y^{mod} by a metamodel which is computationally faster and, if its functional form allows it, provides an analytical expression

for the likelihood⁶.

- Developing the above strategies by keeping in mind that the computations will be distributed on a large number of processors.

3.3 Finite-cost optimization

3.3.1 A sequential finite-cost algorithm based on kriging

Formulation of the finite-cost optimization problem. Optimization methods, to the exception of the SUR method [90] and the multi-points expected improvement [Ginsbourger et al., C2007] (cf. Section 1.4.4), choose a next iterate that maximizes an immediate figure of merit which is a mixture of expected f and possible progress, e.g., the expected improvement [93]. Such algorithms behave as if the search ended at the next iterate. For global optimization purposes, this is a mistake as it prevents these algorithms from taking the time they have to correctly explore the design space.

If it is known that q analyses can still be performed before the search ends, an appropriate formulation of the unconstrained optimization problem is

$$\min_{(x^1, \dots, x^q) \in \mathcal{S}^q} \min[f(x^1), f(x^2), \dots, f(x^q)] \quad (3.5)$$

that is, one looks for an optimum in the next q evaluations of f . During the optimization, the search points x^i are chosen sequentially according to a predicted payoff criterion, w (for worth), which is maximized. The specificity of “finite-cost optimization methods” is to include all the remaining search points in the worth of the next iterate, i.e., they implement,

$$\max_{x^1 \in \mathcal{S}} w[x^1, x^2(x^1), \dots, x^q(x^1)] \quad (3.6)$$

Note that the optimization is carried out with x^1 , the next iterate, and that it is assumed that future search points, x^2, \dots, x^q , depend on x^1 . The choice of x^1 paves the way for later optimization. The critical component of any method that implements (3.6) is the relation between an iterate, x^i , and its successor x^{i+1} . In general, since the real $f(x^i)$ is not known in advance, use will be made of probabilities and there is not a single x^{i+1} associated to x^i , but rather a distribution $X^{i+1}(x^i)$. An example of this is proposed by D. Ginsbourger in his doctoral work where the worth is based on kriging. We now outline this idea. Readers are referred to Section 1.4 for an introduction to kriging.

The two-points sequential expected improvement. It is clearer to start with $q = 2$ points to introduce the idea. In the kriging context, the function f is known at certain points \mathbf{x} and it is globally seen in \mathcal{S} as a conditioned Gaussian process, $[F(x)|F(\mathbf{x}) = \mathbf{f}]$. The improvement at a point x is

$$I(x) = [(\min(\mathbf{f}) - F(x))^+ | F(\mathbf{x}) = \mathbf{f}] .$$

It is extended here to newly evaluated points,

$$I(x|f_1) = [(\min(\mathbf{f}) - \min(F(x), f(x^1)))^+ | F(\mathbf{x}) = \mathbf{f}, F(x^1) = f_1] .$$

The notation will be further compacted by writing $F_1 \equiv F(x^1)$.

⁶For example, if the metamodel can be decomposed into the product of one-dimensional functions of the uncertainties u_i , then the expectation and the variance can simply be calculated by one-dimensional integrals, [92].

The 2-points Sequential Expected Improvement, 2-SEI, is a payoff criterion (an example of w) that predicts the effect of calculating f at x^1 on the overall improvement at the second iterate x^2 :

$$2\text{-SEI}(x^1) = \mathbb{E}_{F_1} \left[\max_{x^2} \mathbb{E}_{F_2} I(x^2 | F_1) \right] \quad (3.7)$$

In algorithmic form, the 2-SEI has the pseudo-code,

```

1. function  $s = 2\text{-SEI}(x^1)$ 
2.    $s = 0$ 
3.   for  $i = 1, M1$  do
4.     sample  $f_1 \sim F(x^1)$ 
5.      $I^* = \max_{x^2} \mathbb{E} I(x^2 | f_1)$ 
6.      $s = s + I^*$ 
7.   end for
8. return  $s/M1$ 

```

Note that, on line 5, $\mathbb{E} I(x^2 | f_1)$ is a one-point expected improvement and therefore its analytical expression is known. The 2-SEI should not be mixed up with the two-points expected improvement, 2-EI, already discussed in Section 1.4: 2-EI involves two independent points while 2-SEI involves only the next iterate and a prediction on its effect one step later. Mathematically, the difference is made clear by rewriting 2-EI,

$$\begin{aligned} EI(x^1, x^2) &= \mathbb{E}[(\min(\mathbf{f}) - \min(F(x^1), F(x^2)))^+ | F(\mathbf{x}) = \mathbf{f}] \\ &= \mathbb{E}_{F_1} [\mathbb{E}_{F_2} I(x^2 | F_1)] . \end{aligned}$$

The dependency on x^2 can be removed by maximization,

$$M_2 EI(x^1) = \max_{x^2} EI(x^1, x^2) = \max_{x^2} \mathbb{E}_{F_1} [\mathbb{E}_{F_2} I(x^2 | F_1)] .$$

This last equation is different from (3.7) because the expectation and the maximum operators do not permute.

The q -points sequential expected improvement. A q -points sequential expected improvement, q -SEI, can be defined recursively:

$$\begin{cases} I_q^* = \max_{x^q} \mathbb{E}_{F_q} [I(x^q | F_1, \dots, F_{q-1})] \\ I_{k-1}^* = \max_{x^{k-1}} \mathbb{E}_{F_{k-1}} [I_k^*] \quad , \quad k = 3, \dots, q \\ q\text{-SEI}(x^1) = \mathbb{E}_{F_1} [I_2^*] \end{cases} \quad (3.8)$$

Of course, the computational cost of q -SEI is tremendous because of the imbricated expectations (performed in general by Monte Carlo simulations) and maximizations. It is even higher than the cost of q -EI. However, problem (3.5) expresses the real global optimization challenge, i.e., finding the best possible point within a limited number of function evaluations. We therefore think that approximations to problem (3.5) should be investigated, in the same spirit as approximations to q -EI were proposed in [Ginsbourger et al., C2007] (section 1.4).

3.3.2 A finite-cost, multi-fidelity optimization problem

Problem (3.5) includes the computational cost of the simulations in the optimization problem formulation in a rough way, through the number of simulations q that can still be performed. To be more accurate, we propose to account for the predicted

computational cost of the model. In fact, we will say models because, in engineering optimization, there are usually many models available. These models represent various compromises between an accurate description of the physics and an efficient computer simulation. One speaks of various fidelity levels. For example, a finite element model is more accurate when its number of elements increases, but its numerical cost grows simultaneously; a solid mechanics model that accounts for visco-plasticity is physically more accurate than an elastic model but it takes longer to be simulated.

To account for the fidelity of the models in the optimization problem, a variable z , which identifies the model used, is added to the x variables. The model becomes $y(x, z)$ and, similarly, the optimization criteria become $f(x, z) \equiv f(y(x, z))$ and $g(x, z) \equiv g(y(x, z))$. z is a finite discrete number when there are different separate simulators available, or it is a natural number when the model fidelity is parameterized. The two principal examples of the latter are *i*) z is the number of elements of a finite elements model, or, *ii*) z is the number of simulations of a Monte Carlo analysis. In addition to f and g , one assumes that an accuracy function $a(x, z)$ and a computational cost function $c(x, z)$ are known. To shorten notations, the extended variable $s = (x, z)$ is introduced which belongs to $\mathcal{M} = \{S \cup \mathcal{D}\}$ or $\mathcal{M} = \{S \cup \mathbb{N}\}$.

A general finite-cost, multi-fidelity optimization problem can be written,

$$\begin{cases} \min_{q, (s^1, \dots, s^q) \in \{\mathbb{N} \cup \mathcal{M}\}^q} f(s^*) , \\ \text{where } s^* = \arg \min_{s^i / a(s^i) \geq a^{\min}} [f(s^1), f(s^2), \dots, f(s^q)] , \\ \text{such that } \sum_{i=1}^q c(s^i) \leq C^{\max} , \\ \text{(and } g(s^*) \leq 0 \text{ for constrained problems)} \end{cases} \quad (3.9)$$

This problem is non-parametric because q , the number of points to be calculated, is unknown. Out of the q points, one considers the performance of the best point that satisfies a minimum accuracy condition, $a(x^i, z^i) \geq a^{\min}$. The constraint put on $\sum_{i=1}^m c(x^i, z^i)$ says that the total cost of the search should be lower than an allowable computation budget, C^{\max} . Problem (3.9) is the finite-cost problem (3.5) with additional cost constraints that lead to a multi-fidelity formulation.

It is a very complex problem and it will be necessary to simplify it. A first step is to solve problem (3.9) in batches of r points, $r < q$, where the r points are chosen simultaneously. The simplification is that one does not have to exhibit the dependency of the points, $x^{i+1}(x^i)$, as it was done in section 3.3.1. The problem is formulated like in (3.9) with r instead of q . An implementation to solve it could use a kriging metamodel with nugget effect to account for the accuracy a and it would consist in

$$\begin{cases} \max_{s^1, \dots, s^r} EI(s^1, \dots, s^r) , \\ \text{such that } a(s^*) \geq a^{\min} , \text{ where } s^* = \arg \max_{i=1, r} EI(s^i) , \\ \text{and } \sum_{i=1}^r c(s^i) \leq C^{\max} . \end{cases}$$

Chapter 4

Bibliography

4.1 External bibliography

Bibliography

- [1] N. Alexandrov and R. M. Lewis. Analytical and computational aspects of collaborative optimization. Technical Report NASA TM 2000-2101, NASA, Langley Research Center, 2000.
- [2] J. Anderson, K. Streitlien, D. Barret, and M. Triantafyllou. Oscillating foils of high propulsive efficiency. *Journal of fluids mechanics*, 360:41–72, 1998.
- [3] J. Antonisse. A new interpretation of schema notation that overturns the binary encoding constraint. In J. D. Schaffer, editor, *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 86–91. Morgan Kaufmann, June 1989.
- [4] D. V. Arnold and H.-G. Beyer. Local performance of the $(\mu/\mu_i, \lambda)$ -es in a noisy environment. In W. N. Martin and W. M. Spears, editors, *Foundations of Genetic Algorithms 6*, pages 127–141. Morgan Kaufmann, 2001.
- [5] J.S. Arora and Lee T.H. *Structural optimization: status and promise*, volume 150 of *Progress in aeronautics and astronautics*, chapter Design sensitivity analysis of nonlinear structures III: shape variation of viscoplastic structures, pages 447–486. AIAA, 1993. Kamat M.P. ed.
- [6] A. Auger. *Contributions théoriques et numériques à l’optimisation continue par algorithmes évolutionnaires*. PhD thesis, Université Paris 6, December 2004. in French.
- [7] A. Auger, M. Schoenauer, and O. Teytaud. Local and global order 3/2 convergence of a surrogate evolutionary algorithm. In K. Deb et al., editor, *Proceedings of the Genetic and Evolutionary Conference 2004*. LNCS 3102 and 3103, Springer Verlag, 2004.
- [8] S. Avril, M. Bonnet, A.-S. Bretelle, M. Grédiac, F. Hild, P. Ienny, F. Latourte, D. Lemosse, S. Pagano, E. Pagnacco, and F. Pierron. Overview of identification methods of mechanical parameters based on full-field measurements. *Experimental Mechanics*, 2008. to appear.
- [9] T. Bäck. Optimal mutation rates in genetic search. In S. Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 2–8. Morgan Kaufmann, 1993.
- [10] T. Bäck. Selective pressure in evolutionary algorithms: a characterization of selection mechanisms. In Z. Michalewicz, J. D. Schaffer, H.-P. Schwefel, D. B. Fogel, and H. Kitano, editors, *Proceedings of the First IEEE International Conference on Evolutionary Computation*, pages 57–62, 1994.
- [11] T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford Univ. Press, New-York, 1996.

- [12] T. Bäck, M. Schütz, and S. Khuri. A comparative study of a penalty function, a repair heuristic and stochastic operators with set covering problem. In J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, editors, *Artificial Evolution*, number 1063 in LNCS. Springer Verlag, 1995.
- [13] S. Baluja. Population-based incremental learning : a method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, Pittsburg, 1994.
- [14] S. Baluja. An empirical comparison of seven iterative and evolutionary function optimization heuristics. Technical Report CMU-CS-95-193, Carnegie Mellon University, 1995.
- [15] R. A. Bates, R. S. Kennet, D. M. Steinberg, and H. P. Wynn. Achieving robust design from computer simulations. *Quality Technology & Quantitative Management*, 3:161–177, 2006.
- [16] J.C. Bean and A.B. Hadj-Alouane. A dual genetic algorithm for bounded integer programs. Technical Report TR 92-53, Dept. of Industrial and Operations Eng., The Univ. of Michigan, 1992.
- [17] S. BenHamida. *Algorithmes Évolutionnaires : Prise en Compte des Contraintes et Application Réelle*. PhD thesis, Université de Paris 11, Orsay, France, 29 March 2001. (in French).
- [18] J.-M. Berthelot. *Matériaux Composites: comportement mécanique et analyse des structures*. Technique & Documentation, Paris, 3 edition, 1999.
- [19] H.-G. Beyer. An alternative explanation for the manner in which genetic algorithms operate. *BioSystems*, 41:1–15, 1997.
- [20] H.-G. Beyer. *The Theory of Evolution Strategies*. Springer, 2001.
- [21] H.-G. Beyer and H.-P. Schwefel. Evolution strategies : a comprehensive introduction. *Natural Computing*, 1:3–52, 2002.
- [22] H.-G. Beyer and B. Sendhoff. Evolution strategies for robust optimization. *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 1346–1353, 2006.
- [23] P. A. N. Bosman and D. Thierens. Expanding from discrete to continuous estimation of distribution algorithms: The IDEA. In M. Schoenauer et al., editor, *Proceedings of the 6th Conference on Parallel Problems Solving from Nature*, pages 767–776. Springer-Verlag, LNCS 1917, 2000.
- [24] Baker C.A., Watson L. T., Grossman B., Mason W. H., and Haftka R. T. Parallel global aircraft configuration design space exploration. *Practical parallel computing*, pages 79–96, 2001.
- [25] A.H. Aguirre C.A. Coello Coello and E. Zitzler, editors. *Proceedings of Evolutionary Multi-Criterion Optimization'05*. Springer Verlag, LNCS 3410, 2005.
- [26] Rasmussen C.E. and Williams K.I. *Gaussian Processes for Machine Learning*. M.I.T. Press, 2006.
- [27] R. Cerf. *Une théorie asymptotique des algorithmes génétiques*. PhD thesis, Université de Montpellier II, France, 1994.

- [28] W. Chen, R. Jin, and A. Sudjianto. Analytical variance-based global sensitivity analysis in simulation-based design under uncertainty. *ASME Journal of Mechanical Design*, 127:875 – 886, 2005.
- [29] K.K. Choi. *Structural optimization: status and promise*, volume 150 of *Progress in aeronautics and astronautics*, chapter Design sensitivity analysis of nonlinear structures II, pages 407–446. AIAA, 1993. Kamat M.P. ed.
- [30] D. Claire, F. Hild, and S. Roux. A finite element formulation to identify damage fields. *Int. J. Num. Meth. in Eng.*, 61:189–208, 2004.
- [31] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, 2002.
- [32] N.J. Cramer. A representation for the adaptive generation of simple sequential programs. In J. J. Grefenstette, editor, *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 183–187. Laurence Erlbaum Associates, 1985.
- [33] N. A. C. Cressie. *Statistics for spatial data*. Wiley, revised edition, 1993.
- [34] Geman D. and Jedynek B. An active testing model for tracking roads in satellite images. Technical report, Institut National de Recherches en Informatique et Automatique (INRIA), December 1995.
- [35] Huang D., Allen T.T., Notz W., and Zheng N. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of Global Optimization*, 34:441–466, 2006.
- [36] G.B. Dantzig and P. Wolfe. The decomposition algorithm for linear programming. *Econometrica*, 29(4):767–778, 1961.
- [37] L. Davis. Applying adaptive algorithms to epistatic domains. In *Proc. Intl. Joint Conference on Artificial Intelligence*, 1985.
- [38] L. Davis. *Genetic Algorithms and Simulated Annealing*. Pitman, London, 1987.
- [39] L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.
- [40] K. Deb. *Multi-objective optimization using genetic algorithms*. Wiley, 2001.
- [41] K. Deb and H.-G. Beyer. Self-adaptive genetic algorithms with simulated binary crossover. *Evolutionary Computation*, 2001.
- [42] K. A. DeJong. Are genetic algorithms function optimizers ? In R. Manner and B. Manderick, editors, *Proceedings of the 2nd Conference on Parallel Problems Solving from Nature*, pages 3–13. North Holland, 1992.
- [43] K.A. DeJong. *The Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, Ann Harbor, 1975. *Dissertation Abstract International*, 36(10), 5140B. (University Microfilms No 76-9381).
- [44] Ecole Nationale Supérieure des Mines de Paris. Zebulon 7. Technical Report User’s manual, Centre des matériaux, France, 1998.
- [45] Jones D.R. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21:345–383, 2001.

- [46] Jones D.R., Pertunen C.D., and Stuckman B.E. Lipschitzian optimization without the lipshitz constant. *Journal of Optimization Theory and Application*, 79, October 1993.
- [47] N. R. Draper and H. Smith. *Applied regression analysis*. Wiley, 3 edition, 1998.
- [48] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, 2 edition, 2001.
- [49] N. Durand and J.-M. Alliot. A combined nelder-mead simplex and genetic algorithm. Technical report, ENAC (ISAE), 1999. available at <http://www.recherche.enac.fr/opti/papers/>.
- [50] A. J. Durelli and K. Rajaiah. Optimum hole shapes in finite plates under uniaxial loads. *ASME J. Appl. Mech.*, 46:691–695, 1976.
- [51] R. Duval, M. Martinelli, and C. Praveen. Uncertainty propagation in cfd using surrogates and automatic differentiation. In *5th European Congress on Computational Methods in Applied Sciences and Engineering*, Venice, Italy, June 2008.
- [52] B. Efron and R. Tibshirani. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical science*, 1(1):54–77, 1986.
- [53] EGEE. Enabling Grids for E-science, 2008. <http://eu-egee.org>.
- [54] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Springer Verlag, 2003.
- [55] D. B. Fogel. *Evolving artificial intelligence*. PhD thesis, University of California, San diego, CA, 1992.
- [56] D. B. Fogel. *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ, 1995.
- [57] L. J. Fogel. Autonomous automata. *Industrial Research*, 4:14–19, 1962.
- [58] C.M. Fonseca, P.J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors. *Proceedings of Evolutionary Multi-Criterion Optimization'03*. Springer Verlag, LNCS 2632, 2003.
- [59] J. R. Fonseca, M. I. Friswell, and A. W. Lees. Efficient robust design via Monte Carlo sample reweighting. *Int. J. Numer. Meth. Engng.*, 2007.
- [60] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer-Verlag, New York, 1991.
- [61] D. E. Goldberg. *The Design of Innovation : Lessons from and for Competent Genetic Algorithms*. Kluwer, 2002.
- [62] D. E. Goldberg, K. Deb, and J. H. Clark. Genetic algorithms, noise, and the sizing of populations. Technical Report IlliGAL 90007, Univ. of Illinois at Urbana-Champaign, 1991.
- [63] D. E. Goldberg, B. Korb, and K. Deb. Messy genetic algorithms: Motivations, analysis and first results. *Complex Systems*, 3:493–530, 1989.

- [64] D. E. Goldberg, B. Korb, and K. Deb. Messy genetic algorithms revisited: Nonuniform size and scale. *Complex Systems*, 4:415–444, 1990.
- [65] D. E. Goldberg and R. E. Smith. Nonstationary function optimization using genetic algorithms with dominance and diploidy. In *Proceedings of the 2nd International Conference on Genetic Algorithms*, pages 59–68. Morgan Kaufmann, 1987.
- [66] David E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, 1989.
- [67] SAS JMP Gorgia Tech, Boeing. ASDL: Aerospace systems design lab, 2008. <http://www.asdl.gatech.edu>.
- [68] J. J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-16(1):122–128, 1986.
- [69] L. Grosset. *Optimization of Composite Structures by Estimation of Distribution Algorithms*. PhD thesis, Univ. of Florida and Ecole Nationale Supérieure des Mines de Saint-Etienne, September 2004.
- [70] L. Grosset, R. Le Riche, and R. T. Haftka. A double-distribution statistical algorithm for composite laminate optimization. *Structural and Multidisciplinary Optimization*, 31(1):49–59, January 2006.
- [71] Z. Gürdal, R.T. Haftka, and P. Hajela. *Design and optimization of laminated composite materials*. Wiley, 1999.
- [72] R.B. Haber, D.A. Tortorelli, and C.A. Vidal. *Structural optimization: status and promise*, volume 150 of *Progress in aeronautics and astronautics*, chapter Design sensitivity analysis of nonlinear structures I: large-deformation hyperelasticity and history-dependent material response, pages 369–406. AIAA, 1993. Kamat M.P. ed.
- [73] A.B. Hadj-Alouane and J.C. Bean. A genetic algorithm for the multiple-choice integer program. Technical Report TR 92-50, Dept. of Industrial and Operations Eng., The Univ. of Michigan, 1992.
- [74] R. Haftka, E. P. Scott, and J. R. Cruz. Optimization and experiments: A survey. *Applied Mechanics Review*, 51(7):435–448, 1998.
- [75] R.T. Haftka and R.V. Grandhi. Structural shape optimization – a survey. *Comp. Methods Applied Mech. Eng.*, 57:91–106, 1986.
- [76] R.T. Haftka and Z. Gürdal. *Elements of structural optimization*. Kluwer, 3rd revised and expanded edition, 1991.
- [77] R.T. Haftka and M.P. Kamat. Simultaneous nonlinear structural analysis and design. *Computational Mechanics*, 4(6):409–416, 1989.
- [78] H. Hamda, F. Jouve, E. Lutton, M. Schoenauer, and M. Sebag. Compact unstructured representations in evolutionary topological optimum design. *Applied Intelligence*, 16:139–155, 2002.
- [79] H. Hamda and M. Schoenauer. Toward hierarchical representations for evolutionary topological optimum design. In J. Périaux et al., editor, *Eurodays 2000, in memoriam of B. Mantel*. John Wiley, 2002.

- [80] N. Hansen. *Towards a new evolutionary computation. Advances in estimation of distribution algorithms*, chapter The CMA Evolution Strategy: A Comparing Review, pages 75–102. Springer, 2006.
- [81] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaption. In *Proceedings of the Third IEEE International Conference on Evolutionary Computation*, pages 312–317. IEEE Press, 1996.
- [82] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
- [83] N. Hansen, A. Ostermeier, and A. Gawelczyk. On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In L. J. Eshelman, editor, *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 57–64. Morgan Kaufmann, 1995.
- [84] W. E. Hart, N. Krasnogor, and J.E. Smith. Recent advances in memetic algorithms. In *Series: Studies in Fuzziness and Soft Computing*, volume 166. Springer, 2005.
- [85] W.E. Hart, N. Krasnogor, and J. E. Smith. Special issue on memetic algorithms. In *Evolutionary Computation*. MIT Press, 2004.
- [86] John H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1975.
- [87] S. Howe. New conditions for exactness of a simple penalty function. *SIAM Journal of Control*, 11(2):378–381, 1973.
- [88] C. Igel and M. Toussaint. A no-free-lunch theorem for non-uniform distributions of target functions. *Journal of Mathematical Modelling and Algorithms*, 3:313–322, 2004.
- [89] INRIA. ProActive Parallel Suite. Technical report, Activeeon, 2008. <http://proactive.inria.fr>.
- [90] Villemonteix J., Vazquez E., and Walter E. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 2007. accepted for publication.
- [91] P. Trompette J.-L. Marcelin and R. Dornberger. Optimization of composite beam structures using a genetic algorithm. *Structural Optimization*, 9(3/4):236–244, 1995.
- [92] R. Jin, W. Chen, and A. Sudjianto. Analytical metamodel-based sensitivity analysis and uncertainty propagation for robust design. In *SAE Transactions*, Detroit, MI, 2004. paper 2004-01-0429.
- [93] D.R. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.
- [94] T. Jones. *Evolutionary algorithms, fitness landscapes and search*. PhD thesis, University of New Mexico, 1995.
- [95] K. A. De Jong. *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, Ann Arbor, MI, 1975.

- [96] K. A. De Jong. *Evolutionary Computation: a unified approach*. MIT Press, 2006.
- [97] A. Kale, E. Acar, T. Haftka, R., and J. Stroud. Why are airplanes so safe structurally? effect of various safety measures on structural safety of aircraft. In *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Palm Springs, CA, April 2004. AIAA-2004-1629.
- [98] L. Kallel. *Convergence des algorithmes génétiques : aspects spatiaux et temporels*. PhD thesis, Ecole Polytechnique, Palaiseau, France, Feb. 1999.
- [99] L. Kallel and M. Schoenauer. Alternative random initialization in genetic algorithms. In Th. Bäck, editor, *Proceedings of the 7th International Conference on Genetic Algorithms*, pages 268–275. Morgan Kaufmann, 1997.
- [100] C. Kane, F. Jouve, and M. Schoenauer. Structural topology optimization in linear and nonlinear elasticity using genetic algorithms. In *Proceedings of the ASME 21st Design Automation Conference*. ASME, Boston, Sept. 1995.
- [101] C. Kane and M. Schoenauer. Genetic operators for two-dimensional shape optimization. In J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, editors, *Artificial Evolution*, number 1063 in LNCS. Springer Verlag, Septembre 1995.
- [102] C. Kane and M. Schoenauer. Topological optimum design using genetic algorithms. *Control and Cybernetics*, 25(5):1059–1088, 1996.
- [103] C. Kane and M. Schoenauer. Optimisation topologique de formes par algorithmes génétiques. *Revue Française de Mécanique*, 4:237–246, 1997.
- [104] A. J. Keane and P. B. Nair. *Computational Approaches for Aerospace Design*. Wiley, 2005.
- [105] P. Kere and J. Lento. Design optimization of laminated composite structures using distributed grid resources. *Composite Structures*, 71(3-4):435–438, december 2005. Fifth International Conference on Composite Science and Technology - ICCST/5.
- [106] S. Kern, N. Hansen, S. Müller, D. Büche, J. Ocenasek, and P. Koumoutsakos. Distributions in continuous evolutionary algorithms - review and comparison. *Natural Computing*, 3(1):77–112, 2004.
- [107] J.-H. Kim and H. Myung. Evolutionary programming techniques for constrained optimization. *IEEE Trans. on Evolutionary Computation*, pages 129–140, July 1997.
- [108] N. H. Kim, H. Wang, and N. V. Queipo. Adaptive reduction of design variables using global sensitivity in reliability-based optimization. *International Journal of Reliability and Safety*, 1:102 – 119, 2006.
- [109] N. H. Kim, H. Wang, and N. V. Queipo. Efficient shape optimization under uncertainty using polynomial chaos expansions and local sensitivities. *AIAA JOURNAL*, 44(5):1112–1115, May 2006. technical note.
- [110] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [111] J. Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE transactions on evolutionary computation*, 2005.

- [112] N. Kogiso, L.T. Watson, Z. Gürdal, and R.T. Haftka. Genetic algorithms with local improvement for composite laminate design. *Structural Optimization*, 7(4):207–218, 1994.
- [113] N. Kogiso, L.T. Watson, Z. Gürdal, R.T. Haftka, and Nagendra S. Design of composite laminates by a genetic algorithm with memory. *Mechanics of Composite Materials and Structures*, 1(1):95–117, 1994.
- [114] J. R. Koza. *Genetic Programming: On the Programming of Computers by means of Natural Evolution*. MIT Press, Massachusetts, 1992.
- [115] J. R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Massachusetts, 1994.
- [116] J. R. Koza and al. *Genetic Programming III: Automatic Synthesis of Analog Circuits*. MIT Press, Massachusetts, 1999.
- [117] I. Kroo. Distributed multidisciplinary design and collaborative optimization. In *VKI lecture series on Optimization Methods and Tools for Multi-criteria/Multidisciplinary Design*. VKI, November 2004.
- [118] A. Kumar, P. B. Nair, A. J. Keane, and S. Shahpar. Robust design using bayesian monte carlo. *Int. J. Numer. Meth. Engng.*, 73:1497–1517, 2008.
- [119] A. H. Land and A. G. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28:497–520, 1960.
- [120] P. Larranaga and J. A. Lozano. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2001.
- [121] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Archiving with guaranteed convergence and diversity in multi-objective optimization. In W.B. Langdon & al., editor, *Proceedings of the Genetic and Evolutionary Conference 2002*, pages 39–447. Morgan Kaufmann, 2002.
- [122] E. L. Lawler and D. E. Wood. Branch-and-bound methods: a survey. *Operations research*, 14:699–719, 1966.
- [123] R. Le Riche, C. Knopf-Lenoir, and R. T. Haftka. A segregated genetic algorithm for constrained structural optimization. In *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA95)*, pages 558–565. Morgan Kaufman, 1995.
- [124] K.-H. Lee and G.-J. Park. A global robust optimization using kriging based approximation model. *JSME International Journal*, 49(3):779–788, 2006.
- [125] S. Lin and B. Kernighan. An efficient heuristic procedure for the traveling salesman problem. *Operations Res.*, 21:498–516, 1973.
- [126] Schonlau M. *Computer Experiments and Global Optimization*. PhD thesis, University of Waterloo, 1997.
- [127] S. W. Mahfoud. *Niching Methods for Genetic Algorithms*. PhD thesis, University of Illinois, Urbana Champaign, 1995. also Illigal Report 95001.
- [128] S. W. Mahfoud and D. E. Goldberg. Parallel recombinative simulated annealing: a genetic algorithm. *Parallel Computing*, 21:1–28, 1995.

- [129] W. N. Martin, J. Liening, and J. P. Cohoon. *Evolutionary Computation 2: Advanced Algorithms and Operators*, chapter Island (migration) models: evolutionary algorithms based on punctuated equilibria, pages 101–124. Inst. of Physics Editor, Bristol, UK, 2000. T. Bäck, D. B. Fogel and Z. Michalewicz, eds.
- [130] M. Martinelli, C. Praveen, and R. Duvigneau. On the estimation of drag uncertainty. In *43rd AAAF Congress on Applied Aerodynamics*, Poitiers, France, 2008.
- [131] C. Mattheck and S. Burkhard. A new method of structural shape optimization based on biological growth. *Int. J. Fatigue*, 12:185–190, 1990.
- [132] O. Le Maître, S. Huberson, and E. Souza de Curci. Unsteady model of sail and flow interaction. *Journal of Fluids and Structures*, 13:37–59, 1999.
- [133] R. E. Melchers. *Structural reliability analysis and prediction*. Wiley, 2 edition, 1999.
- [134] A. Meyer. *Filengrene : un logiciel d'analyse et de fabrication d'engrenages*. INSA de Rouen, France, 2005-2008.
- [135] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, 3 edition, 1992-1996.
- [136] Z. Michalewicz and M. Schoenauer. Evolutionary algorithms for constrained parameter optimization. *Evolutionary Computation*, 4(1):1–32, 1997.
- [137] M. Miki. Optimum design of laminated composite plates subject to axial compression. In K. Kawabata, S. Umekawaand, and A. Kobayashi, editors, *Proc. Japan-U.S. CCM-III*, pages 673–680, 1986.
- [138] M. Minoux. *Mathematical programming: Theory and Algorithms*. Wiley, 1986.
- [139] H. Mühlenbein. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5:303–346, 1998.
- [140] H. Mühlenbein and T. Mahnig. *Theoretical aspects of evolutionary computing*, chapter Evolutionary algorithms: from recombination to search algorithms, pages 135–174. Springer, 2000.
- [141] S. Nagendra, D. Jestin, Z. Gürdal, R.T. Haftka, and L. T. Watson. Improved genetic algorithms for the design of stiffened composite panels. *Computers & Structures*, 58(3):543–555, 1996.
- [142] J.A. Nelder and R. Mead. A simplex for function minimization. *Computer J.*, 7:308–313, 1965.
- [143] F. Nieto and S. Hernandez. Distributed computing for design optimization of the messina bridge considering aeroelastic constraints. *Journal of Wind Engineering*, 108:757–760, 2006.
- [144] J. Oda and K. Yamazaki. On a technique to obtain an optimum strength shape of axisymmetric body by the finite element method. *Bull. JSME*, 20:1524–1532, 1977.
- [145] Faye P., Paluch B., and Grediac M. Optimisation de structures en matériaux composites par algorithmes génétiques. Technical Report ONERA-TP - 03-68, ONERA, 2003. in French.

- [146] J. Paredis. Co-evolutionary constraint satisfaction. In Y. Davidor, H.-P. Schwefel, and R. Manner, editors, *Proceedings of the 3rd Conference on Parallel Problems Solving from Nature*, pages 46–55. Springer-Verlag, LNCS 866, 1994.
- [147] Jan Paredis. *Constraint satisfaction with coevolution*, pages 359–366. McGraw-Hill Ltd., UK, Maidenhead, UK, England, 1999.
- [148] M. Pelikan, D.E. Goldberg, and E. Cantu-Paz. Boa: the bayesian optimization algorithm. In D.E. Goldberg & al., editor, *Proceedings of the Genetic and Evolutionary Conference 1999*, pages 525–532. Morgan Kaufmann, 1999.
- [149] D. Powell and M.M. Skolnick. Using genetic algorithms in engineering design optimization with non-linear constraints. In *Proc. of the Fourth International Conference on Genetic Algorithms*, pages 424–431, San Mateo CA, 1991. Morgan Kaufmann.
- [150] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1–28, January 2005.
- [151] Spallino R. and Thierauf G. Thermal buckling optimization of composite laminates by evolution strategies. *Computers & Structures*, 78(5):691–697, 2000.
- [152] N. J. Radcliffe. Genetic set recombination. In Foundations of Genetic Algorithms 2, editor, *Foundations of Genetic Algorithms 2*, pages 203–220. Morgan Kaufmann, 1993.
- [153] N. J. Radcliffe and P. D. Surry. Fitness variance of formae and performance prediction. In L. D. Whitley and M. D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 51–72. Morgan Kaufmann, 1995.
- [154] P. Ramu, X. Qu, B.D Youn, and K.K. Choi. Inverse reliability measures and reliability-based design optimization. *Int. J. Reliability and Safety*, 1(1/2):187–205, 2006.
- [155] A. Ratle and M. Sebag. Grammar-guided genetic programming and dimensional consistency: Application to non-parametric identification in mechanics. *Applied Soft Computing*, 1(3):105–118, 2001.
- [156] A. Ratle and M. Sebag. Avoiding the bloat with stochastic grammar-based genetic programming. In P. Collet, E. Lutton, M. Schoenauer, C. Fonlupt, and J.-K. Hao, editors, *Artificial Evolution'01*, pages 254–266. Springer Verlag, LNCS 2310, 2002.
- [157] I. Rechenberg. *Cybernetic solution path of an experimental problem*. Royal Aircraft Establishment, Farnborough, Library Translation 1122, 1965.
- [158] I. Rechenberg. *Evolutionstrategie: Optimierung technischer System nach Principen der Biologischen Evolution*. Frommann-Holzboog, Stuttgart, 1973.
- [159] C. Reeves and C. Wright. Epistasis in genetic algorithms: an experimental design perspective. In L. J. Eshelman, editor, *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 217–230. Morgan Kaufmann, 1995.

- [160] J.T. Richardson, M.R. Palmer, G. Liepins, and M. Hilliard. Some guidelines for genetic algorithms with penalty functions. In J. D. Schaffer, editor, *Proc. of the Third International Conference on Genetic Algorithms*, pages 191–197, San Mateo CA, George Mason Univ., 1989. Morgan Kaufmann.
- [161] R.T. Rockafellar. Lagrange multipliers in optimization. In R.W. Cottle and C.E. Lemke, editors, *Nonlinear Programming, SIAM-AMS Proc.*, volume 9, pages 145–168, New York, 1976.
- [162] G. Rudolf. Finite markov chain results in evolutionary computation: A tour d’horizon. *Fundamenta Informaticae*, 35(1-4):67–89, 1998.
- [163] L. A. Schmit. Structural design by systematic synthesis. In ASCE, editor, *Proceedings of the 2nd conference on electronic computation*, pages 105–132, 1960.
- [164] F. Schoen. Stochastic techniques for global optimization: a survey of recent advances. *Journal of Global Optimization*, 1:207–228, 1990.
- [165] M. Schoenauer. Bio-inspired continuous optimization: the coming of age. In *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, Singapore, 2007.
- [166] M. Schoenauer, F. Jouve, and L. Kallel. *Evolutionary Computation in Engineering*, chapter Identification of mechanical inclusions, pages 477–494. Springer Verlag, 1997. D. Dasgupta and Z. Michalewicz, ed.
- [167] M. Schoenauer, M. Sebag, F. Jouve, B. Lamy, and H. Maitournam. Evolutionary identification of macro-mechanical models. In P. J. Angeline and Jr K. E. Kinneer, editors, *Advances in Genetic Programming II*, pages 467–488, Cambridge, MA, 1996. MIT Press.
- [168] H.-P. Schwefel. *Numerical Optimization of Computer Models*. John Wiley & Sons, New-York, 1981. 1995 – 2nd edition.
- [169] H.-P. Schwefel and G. Rudolf. Contemporary evolution strategies. In F. Morán et al., editor, *Proc. of the 3rd Int. Conf. on Artificial Life*, pages 893–907. Springer, 1995. Lecture Notes in Artificial Intelligence 929.
- [170] B. Schölkopf, C. Burges, and A. Smola. *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.
- [171] M. Sebag and A. Ducoulombier. Extending population-based incremental learning to continuous search spaces. In Th. Bäck, A.E. Eiben, M. Schoenauer, and H.-P. Schwefel, editors, *Proceedings of the 5th Conference on Parallel Problems Solving from Nature*, pages 418–427. Springer Verlag, 1998.
- [172] M. Sebag, M. Schoenauer, and M. Peyral. Revisiting the memory of evolution. *Fundamenta Informaticae*, 38:1–39, 1998.
- [173] M. Sfakiotakis, D. Lane, and B. Davies. Review of fish swimming modes for aquatic locomotion. *IEEE Journal of Oceanic Eng.*, 24(2):237–252, 1999.
- [174] M.-H. Sheng, Q.-M. Shao, and J. G. Ibrahim. *Monte Carlo Methods in Bayesian Computation*. Springer, 2000.
- [175] Y. A. Shreider. *The Monte Carlo method*. Pergamon Press, Oxford, 1966.

- [176] G. Silva. *Identification from full field measurements of local and global properties using F.E.M.U.* PhD thesis, Ecole National Supérieure des Mines de Saint-Etienne, Saint-Etienne, France, Oct. 2008.
- [177] T. W. Simpson, J. Peplinski, P. N. Koch, and J. K. Allen. Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with Computers*, 17(2):129–150, 2001.
- [178] A.E. Smith and D.M. Tate. Genetic optimization using a penalty function. In R. K. Belew and L. B. Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 499–505, San Mateo, CA, 1991. Morgan Kaufmann.
- [179] G. Soremekun, Z. Gürdal, R.T. Haftka, and L.T. Watson. Composite laminate design optimization by genetic algorithm with generalized elitist selection. *Computers & Structures*, 79(2):131–143, 2001.
- [180] J. C. Spall. *Introduction to stochastic search and optimization*. Wiley, 2003. <http://www.jhuapl.edu/ISSO/>.
- [181] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. Technical Report Kan-GAL Report 2005005, IIT Kanpur, India, 2005. see <http://www.bionik.tu-berlin.de/user/niko/cec2005.html>.
- [182] M.-J. Tahk and B.-C. Sun. Co-evolutionary augmented lagrangian methods for constrained optimization. *IEEE Trans. on Evolutionary Computation*, February 1999. submitted for publication.
- [183] Tropics team. Tapenade : Online automatic differentiation engine. Technical report, INRIA Sophia Antipolis, 2008. <http://www-sop.inria.fr/tropics>.
- [184] Santner T.J., Williams B.J., and Notz W.J. *The Design and Analysis of Computer Experiments*. Springer, 2003.
- [185] A. Todoroki and R.T. Haftka. Lamination parameters for efficient genetic optimization of the stacking sequence of composite panels. In *Proc. 7th AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis and Optimization Symposium*, pages 870–879, 1998.
- [186] S.W. Tsai and N.J. Pagano. Invariant properties of composite materials. In S.W. Tsai, J.C. Halpin, and N.J. Pagano, editors, *Composite Materials Workshop*, 1968.
- [187] F. Van Keulen, R.T Haftka, and N.H. Kim. Review of options for structural design sensitivity analysis. part 1: Linear systems. *Computer Methods in Applied Mechanics and Engineering*, 194(30-33):3213–3243, 2005.
- [188] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.
- [189] G. G. Vining and R. H. Myers. Combining taguchi and response surface philosophies: A dual response approach. *Journal of Quality Technology*, 22:38–45, 1990.
- [190] B.K. Voon and M. Austin. Structural optimization in a distributed computing environment. Technical Report ISR; TR 1991-104, Digital Repository at the University of Maryland, 1991. <http://hdl.handle.net/1903/5150>.

- [191] M. D. Vose. *The Simple Genetic Algorithm*. MIT Press, Cambridge, MA, 1999.
- [192] F. De Vuyst and C. Audouze. Physics-based metamodeling for parameterized pde problems using space-parameter space principal component analysis. Technical Report october 8-10, CEA-EDF-INRIA School, INRIA Rocquencourt, France, 2007.
- [193] M. Wahib, A. Munawar, M. Munetomo, and K. Akama. Mhgrid: Towards an ideal optimization environment for global optimization problems using grid computing. In *Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies*, pages 167–168. IEEE Computer Society, 2007.
- [194] L. D. Whitley. Fundamental principles of deception in genetic search. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 221–241. Morgan Kaufmann, 1991.
- [195] L. D. Whitley, T. Starkweather, and D. Shaner. *Handbook of Genetic Algorithms*, chapter Traveling salesman and sequence scheduling : quality solutions using genetic edge recombination, pages 350–372. Van Nostrand Reinhold, New York, 1991. L. Davis, ed.
- [196] D. H. Wolpert and W. G. MacReady. No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fe Institute, Santa Fe, NM, 1995. <http://www.no-free-lunch.org>.
- [197] D. H. Wolpert and W. G. MacReady. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, april 1997.
- [198] M.H. Wright. Direct search methods: Once scorned, now respectable. In *Dundee Biennial Conference in Numerical Analysis*, pages 191–208, Harlow, UK, 1996.
- [199] B. A. Wujek, J. E. Renaud, S. M. Batill, and J. B. Brockman. Concurrent subspace optimization using design variable sharing in a distributed computing environment. *Concurrent Engineering*, 4(4):361–377, 1996.
- [200] E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne, editors. *Proceedings of Evolutionary Multi-Criterion Optimization'01*. Springer Verlag, LNCS 1993, 2001.

4.2 Personal bibliography

The letters P, J, C, S, T and V in front of the year refer to patents, journal articles, conference articles, seminars, technical reports (including contract reports) and popularizing articles respectively. Table 4.1 summarizes the number of publications per year and per type. The following facts should be stressed:

- Two patents have been registered.
- The communication [Le Riche, S2004] is an invited talk in a workshop devoted to state-of-the-art presentations in numerical methods.
- The article [Le Riche et Guyon, C2001] obtained the best paper price at “Artificial Evolution 01”.

	Patents	Journals	Conf. + P + R	Conf. + P	Seminars	Others
2008	0	1	1	2	0	1
2007	1	4	3	1	1	4
2006	0	2	1	1	1	1
2005	1	1	0	0	0	0
2004	0	2	5	1	1	4
2003	0	4	3	1	1	3
2002	0	3	3	1	4	1
2001	0	1	3	0	0	4
2000	0	0	0	0	1	3
1999	0	0	1	0	0	1
1998	0	3	0	1	1	1
1997	0	1	3	0	0	0
1996	0	1	1	0	0	2
1995	0	1	2	1	0	1
91-94	0	1	3	0	1	0
Total	2	25	29	9	11	26

Table 4.1: Number of publications and communications per year. “Conf.+P+R” designates conferences with proceedings and reviewing, “Conf.+P” are conferences with proceedings but without reviewing, “Seminars” are all the presentations without proceedings, and “Others” include technical reports, contract reports, popularizing articles and dissertations. Cf. section 4.2 for an extensive list.

- The article [Le Riche et Haftka, J1993] is cited 179 times on GoogleTM Scholar¹, [Michalewicz et al., J1996] is cited 160 times, [Le Riche et al., C1995-1] is cited 74 times, [Le Riche and Haftka, J1995] is cited 59 times, [Harisson et al., J1995] is cited 21 times.

4.2.1 Patents

[Gualandris et al., P2007] Gualandris, D., Le Riche, R. and Rakotomamonjy, A., *Reconstruction d'historiques temporels du couple moteur*, (re-estimation of instantaneous engine torque) patent INPI No. 05/03/748 (France), submitted on 14 April 2005, accepted on 20 July 2007.

[Bay et al., P2005] Bay, X., Dorel, L., Leclercq, Y., Le Riche, R., Martin, S. and Serval, F., *Régulation de la planéité d'une bande métallique à la sortie d'une cage de laminoir*, (flatness control of a metal sheet in a rolling-mill) patent INPI No. 04-13753 (France), submitted on 22 December 2004.

4.2.2 Journals

[Pujol et al., J2008] G. Pujol, R. Le Riche, O. Roustant and X. Bay, *L'incertitude en conception: formalisation, estimation*, to be published as a chapter in the book *Optimisation Multidisciplinaire de Systèmes Mécaniques : multi-niveaux et robustesse*, Mécanique et Ingénierie des Matériaux series, Hermes publisher, Dec. 2008 (in French).

¹Estimation made in July 2008 on the research engine GoogleTM Scholar, <http://scholar.google.com>, using the key word “le riche”.

- [Gogu et al., J2008] Gogu, C., Haftka, R.T., Le Riche, R., Molimard, J., Vautrin, A., and Sankar, B. *Limits of basic least squares identification - an introduction to the Bayesian statistical approach applied to elastic constants identification*, to appear in the Journal of Physics: Conference Series, an open-access web-based journal published by the Institute of Physics, May 2008.
- [Silva et al., J2007-1] Silva, G., Le Riche, R., Molimard, J. and Vautrin, A., *Exact and Efficient Interpolation using Finite Elements Shape Functions*, to appear in European Journal of Computational Mechanics (Revue Européenne de Mécanique Numérique), Lavoisier Ed., October 2007.
- [Silva et al., J2007-2] Silva, G., Le Riche, R., Molimard, J., Vautrin, A. and Galerne, C., *Identification of Material Properties using FEMU: Application to the Open Hole Tensile Test*, Journal of Applied Mechanics and Materials, vol.7-8, Sept. 2007, pp. 73-78.
- [Rakotomamonjy et al., J2007] A. Rakotomamonjy, R. Le Riche, D.Gualandris and Z. Harchaoui, A comparison of statistical learning approaches for engine torque estimation, Control Engineering Practice, Vol.16, No. 1, Jan. 2008, pp. 43-55 (available online in May 2007 at <http://dx.doi.org/doi:10.1016/j.conengprac.2007.03.009>).
- [Le Riche et al., J2007] Le Riche, R., Schoenauer, M. and Sebag, M., Un état des lieux de l'optimisation évolutionnaire et de ses implications en sciences pour l'ingénieur, chapter in Modélisation Numérique: défis et perspectives, 2nd volume, Traité Mécanique et Ingénierie des Matériaux, P. Breitenkopf and C. Knopf-Lenoir eds., Hermes publ., Feb. 2007, pp. 187-259 (in French).
- [Luersen et al., J2006] Luersen, M.A., Le Riche, R., Lemosse, D. and Le Maître, O., *A computationally efficient approach to swimming monofin optimization*, Structural and Multidisciplinary Optimization, paper no. SMO-05-0038, electronic issue : DOI 10.1007/s00158-006-0001-6, URL <http://dx.doi.org/10.1007/s00158-006-0001-6>, Apr. 2006, pp. 1-9.
- [Grosset et al., J2006] Grosset, L., Le Riche, R. and Haftka, R.T., *A double-distribution statistical algorithm for composite laminate optimization*, Structural and Multidisciplinary Optimization, Vol. 31, No. 1, Jan. 2006, pp. 49-59 (published online in Oct. 2005 as paper no. 1268).
- [Molimard et al., J2005] Molimard, J., Le Riche, R., Vautrin, A. and Lee, J.R., *Identification Of The Four Orthotropic Plate Stiffnesses Using a single Open Hole Tensile Test*, Experimental Mechanics, SAGE Publications, Vol. 45, No. 5, 2005, pp. 404-411.
- [Grosset et al., J2004] Grosset, L., Le Riche, R. and Haftka, R.T., *A study of the effects of dimensionality on stochastic hill climbers and estimation of distribution algorithms*, Lecture Notes in Computer Science, Vol. 2936, selected papers from the 6th International Conference on Artificial Evolution (EA03), Springer Verlag publ., May 2004, pp. 27-38.
- [Luersen and Le Riche, J2004] Luersen, M.A. and Le Riche, R., *Globalized nelder-mead method for engineering optimization*, Computers & Structures, special issue Computational Structures Technology, Edited by Z. Bittnar and B.H.V. Topping, Vol. 82, No. 23-26, pp. 2251-2260, 2004 (received Dec. 2002).
- [Molimard and Le Riche, J2003] Molimard, J. and Le Riche, R., *Identification de piézoviscosité en lubrification*, Mécanique et Industrie, Elsevier publ., Vol.4, No.6

, November-December 2003, pp. 645-653 (in French).

[Luersen et al., J2003] Luersen, M., Le Riche, R. and Guyon, F., *A Constrained Globalized and Bounded Nelder-Mead Algorithm for Engineering Optimization*, Journal of Structural and Multidisciplinary Optimization, Springer, Vol.25, No.4, October 2003 (electronic version, paper version in Vol. 27, No. 1-2, May 2004, pp. 43 - 54).

[Le Riche et al., J2003-2] Le Riche, R., Saouab, A. and Bréard, J., *Coupled compression RTM and composite layup optimization*, Composites Science and Technology, Vol.63, No.15, Elsevier publ., July 2003, pp. 2277-2287.

[Le Riche et al., J2003-1] Le Riche, R., Gaudin, J. and Besson, J., *An Object Oriented Simulation-Optimization Interface*, Computers & Structures, Elsevier publ., Vol.81, No.17, june 2003, pp.1689-1701.

[Rakotomamonjy et al., J2002] Rakotomamonjy, A., Le Riche, R., Gualandris, D. and Canu, S., *Comparaisons de stratégies de discrimination de masses de véhicules automobiles*, Extractions, Connaissances et Apprentissage, Vol.6, numéro special "Méthodes d'optimisation pour l'extraction de connaissances et l'apprentissage", Hermès publ., 2002, pp. 753-784 (in French).

[Saouab et al., J2002] Saouab, A., Le Riche, R. and Bréard, J., *Couplage procédé-propriétés : étude d'une plaque composite*, Revue des Composites et Matériaux Avancés, Hermès publ., Vol.12, No. 3, 2002, pp.515-529 (in French).

[Le Riche and Guyon, J2002] Le Riche, R., and Guyon, F., *Dual Evolutionary Optimization*, Artificial Evolution, Lecture Notes in Computer Science, No. 2310, selected papers of the 5th International Conference, Evolution Artificielle, Collet, P., Lutton, E., Schoenauer, M., Fonlupt, C. and Hao, J.-K., eds, Springer Verlag publ., pp. 281-294, 2002.

[Le Riche et al., J2001] Le Riche, R., Gualandris, D., Thomas, J.-J., and Hemez, F., *Neural Identification of Nonlinear Dynamic Structures*, Journal of Sound and Vibration, Vol.248, No.2, November 2001, pp. 247-265.

[Le Riche and Cailletaud, J1998] Le Riche, R., and Cailletaud, G., *A Mixed Evolutionary / Heuristic Approach to Shape Optimization*, International Journal of Numerical Methods in Engineering, Vol.41, 1998, pp. 1463-1484.

[Le Riche and Gaudin, J1998] Le Riche, R., and Gaudin, J., *Design of Dimensionally Stable Composites by Evolutionary Optimization*, Composite Structures, Vol.41, 1998, pp. 97-111.

[Besson et al., J1998] Besson, J., Le Riche, R., Foerch, R., and Cailletaud, G., *Object Oriented Programming applied to the Finite Element Method - Part II : Application to Material Behavior*, European Journal of Finite Elements (Revue Européenne des Eléments Finis), P. Breitenkopf Ed., Hermes Publ., Vol.7, No.5, August 1998, pp. 567-588.

[Le Riche and Haftka, J1997] Le Riche, R., and Haftka, R.T., *Evolutionary Optimization of Composite Structures*, Evolutionary Algorithms in Engineering Applications, D. Dasgupta and Z. Michalewicz ed., Springer Verlag Publ., ISBN 3-540-62021-4, 1997, pp. 87-103.

[Michalewicz et al., J1996] Michalewicz, Z., Dasgupta, D., Le Riche, R., and Schoenauer, M., *Evolutionary Algorithms for Constrained Engineering Problems*,

Computers & Industrial Engineering Journal, Vol.30, No.4, 1996, pp. 851-869.

[Le Riche and Haftka, J1995] Le Riche, R., and Haftka, R.T., *Improved Genetic Algorithm for Minimum Thickness Composite Laminate Design*, Composites Engineering, Vol.5, No.2, 1995, pp. 143-161.

[Le Riche and Haftka, J1993] Le Riche, R., and Haftka, R.T., *Optimization of Laminate Stacking Sequence for Buckling Load Maximization by Genetic Algorithm*, AIAA Journal, Vol.31, No.5, May 1993, pp. 951-970.

4.2.3 Conferences

Conferences with reviewing

[Gogu et al., C2008] Gogu, C., Haftka, R.T., Le Riche, R., Molimard, J., Vautrin, A., and Sankar, B., *Comparison between the basic least squares and the Bayesian approach for elastic constants identification*, accepted at the 6th Int. Conf. on Inverse Problems in Engineering, ICIPE 2008, Dourdan (Paris), June 15-19 2008.

[Ginsbourger et al., C2007] Ginsbourger, D., Le Riche, R. and Carraro, L., *A Multi-points Criterion for Deterministic Parallel Global Optimization based on Kriging*, Intl. Conf. on Nonconvex Programming, NCP07, Rouen, France, 17-21 december 2007.

[Silva et al., C2007-2] Silva, G., Le Riche, R., Molimard, J., Vautrin, A. and Galerne, C., *Identification of Material Properties using FEMU: Application to the Open Hole Tensile Test*, British Society for Strain Measurement Conference 2007, 4-6th September 2007, University of Manchester, UK.

[Silva et al., C2007-1] Silva, G., Le Riche, R., Molimard, J., Vautrin, A. and Galerne, C., *Aspects numériques de la comparaison éléments finis - mesures de champs: application à l'identification de propriétés de matériaux*, 8ième colloque national en calcul de structures, Giens, France, May 21-25 2007 (in French).

[Park et al., C2006] Park, C.-H., Saouab, A., Bréard, J. and Le Riche, R., *Simple models for mold filling stage in liquid composite molding and their applications to structure-process coupled optimization*, 8th Int. Conf. on Flow Processes in Composite Materials (FPCM8), Douai, France, 11 - 13 July, 2006, (article 16, accepted for presentation and publication in the proceedings).

[Luersen et al., C2004-2] Luersen, M.A., Le Riche R. and Lemosse, D., *Identification of a 3D shape from a 2D design : application to a swimming monofin*, Proc. of the 7th International Conference on Computational Structures Technology CST/ECT 2004, B.H.V. Topping and C.A. Mota Soares ed., Civil-Comp Press, Stirling, England, article no. 30, Lisboa, Portugal, 7-9 Sept. 2004.

[Luersen et al., C2004-1] Luersen, M., Le Riche, R., Lemosse, D. and LeMaître, O., *Swimming Monofin Optimization*, Proc. of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, article AIAA no. 2004-4426, 30 Aug. - 1 Sept., 2004.

[Grosset et al., C2004-2] Grosset, L., Le Riche, R., and Haftka, R.T., *A Two-Tier Estimation of Distribution Algorithm for Composite Laminate Optimization*, Proc. of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, article AIAA no. 2004-4529, 30 Aug - 1 Sept., 2004.

[Molimard et al., C2004-1] J. Molimard, R. Le Riche, A. Vautrin, and J. R. Lee, *Identification of Orthotropic Plate Stiffness Using Open Hole Tensile Test*, 10th SEM International Congress, Costa Mesa, CA, USA, 7-10 June, 2004.

[Grosset et al., C2004-1] Grosset, L., Le Riche, R., and Haftka, R.T., *A Double-Distribution Statistical Algorithm for Composite Laminate Optimization*, 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference, Palm Springs, CA - article AIAA no. 2004-1910, April 2004.

[Grosset et al., C2003] Grosset, L., Le Riche, R. and Haftka, R.T., *A comparison of an estimation of distribution algorithm and a stochastic hill-climber for composite optimization problems*, 18th Annual Technical Conference American Society for Composites, Gainesville, FL, article no. 168, Oct. 19-22, 2003.

[Luersen et al., C2003] Luersen, M.A., Le Riche, R., LeMaître, O. and Breier, E., *Optimisation de palmes de nage*, Proc. of the 6th Colloque en Calcul de Structures, Vol.3, CSMA / AFM publ., Giens, France, May 2003, pp. 207-214 (in French).

[Le Riche et al., C2003-1] Le Riche, R., Gaudin, J. and Vinet, A., *Un logiciel orienté objet d'analyse et de conception de plaques composites*, Proc. of the 6th Colloque en Calcul de Structures, Vol.1, CSMA / AFM publ., Giens, France, May 2003, pp. 87-94 (in French).

[Grosset et al., C2002-2] Grosset, L., Le Riche, R. and Haftka, R.T., *Accounting for plies couplings in composite laminate statistical optimization via lamination parameters*, IUTAM Symposium on Evolutionary Methods in Mechanics, Cracow, Poland, september 24-27, 2002.

[Grosset et al., C2002-1] Grosset, L., Le Riche, R. and Haftka, R.T., *Probability-based genetic algorithm for composite laminate design*, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference, article AIAA no. 2002-5462, september 2002, Atlanta, USA.

[Luersen and Le Riche, C2002] Luersen, M. and Le Riche, R., *Globalized Nelder-Mead Method for Engineering Optimization*, Proc. of the 3rd Intl. Conf. on Engineering Computational Technology (Civil-Comp Press), Paper 65, 4-6 september 2002, Prague, Czech Republic.

[Le Riche and Guyon, C2001] Le Riche, R., and Guyon, F., *Dual Evolutionary Optimization*, Actes de la 5ième conférence internationale sur l'Evolution Artificielle, EA'01, Collet, P., Lutton, E., Schoenauer, M., Fonlupt, C. and Hao, J.-K., editors, Le Creusot, France, oct. 2001, pp. 123-134. (best paper price).

[Le Riche et al., C2001] Le Riche, R., Gualandris, D., Thomas, J.-J. and Hemez, F., *Identification Neuronale de Systèmes Dynamiques Non Linéaires : Application au Monoroue*, Actes du 5ième Colloque en Calcul des Structures, Giens, France, Teknea Publ., 2001, pp. 131-138 (in French).

[Gualandris et al., C2001] Gualandris, D., Thomas, J.-J., Laugerette, M., and Le Riche, R., *Neural Identification of Non Linear Dynamic Structures*, Proc. of the 7th ATA Conference and Exhibition, Florence, Italy, 23 - 25 Mai 2001.

[Deocon et al., C1999] J. Deocon, L. Salvo, P. Lemoine, D.Landru, Y. Brechet and R. Le Riche, *Optimum Design of Metal Foam in Sandwich Structures*, Proceedings of Int. Conf on Metal Foams and Porous Metal structures, Breme, 14-16 june 1999.

[Le Riche et al., C1997-2] Le Riche, R., Feyel, F., Besson, J., Cailletaud, G., Gutmann, M., and Foerch, R., *L'Objet Matériau, de l'Identification au Calcul de Structures*, Actes du Troisième Colloque en Calcul des Structures, Presses Académiques de l'Ouest Publ., Giens, 1997, pp. 583-588 (in French).

[Le Riche and Cailletaud, C1997] Le Riche, R., and Cailletaud, G., *A Mixed Evolutionary / Heuristic Approach to Shape Optimization*, Advances in Computational Eng. Sc., S.N. Atluri and G. Yagawa ed., Tech Science Press Publ., San Jose, Costa Rica, Mai 4-9, 1997, pp. 514-519.

[Le Riche et al., C1997-1] Le Riche, R., Gaudin, J., and Renard, J., *Optimisation de Composites à Haute Stabilité Dimensionnelle par Algorithmes Evolutionnistes*, Actes du Troisième Colloque en Calcul des Structures, Presses Académiques de l'Ouest Publ., Giens, 1997, pp. 811-816 (in French).

[Le Riche et al., C1996] Le Riche, R., Knopf-Lenoir, C., and Haftka, R.T., *Evolutionary Optimization in Mechanics*, Proceedings of the 1996 International Conference on Integrated Design and Manufacturing in Mechanical Engineering, IDMME'96, École Centrale de Nantes, France, 15-17 Avril 1996.

[Le Riche et al., C1995-1] Le Riche, R., Knopf-Lenoir, C., and Haftka, R.T., *A Segregated Genetic Algorithm for Constrained Structural Optimization*, L.Eschelman ed., Genetic Algorithms: Proceedings of the Sixth International Conference on (ICGA95), Morgan Kaufman, San Francisco, CA, 1995, pp. 558-565.

[Harrison et al., C1995] Harrison, P.N., Le Riche, R., and Haftka, R.T., *Design of Stiffened Composite Panels by Genetic Algorithms and Response Surface Approximations*, Proceedings of the 36th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, AIAA-95-1163-CP, New-Orleans, LO, USA, 1995, pp. 58-68.

[Le Riche and Haftka, C1994] Le Riche, R., and Haftka, R.T., *Genetic Algorithms for Minimum Thickness Design of Composite Plates*, Proceedings of the First International Conference on Composite Engineering, ICCE1, New Orleans, LO, USA, 1994.

[Haftka et al., C1994] Haftka, R.T., Le Riche, R. and Harrison, P., *Genetic Algorithms for the Design of Composite Panels*, NATO Advanced Workshop : Emergent Computing Methods in Engineering Design, Napfolio, Greece, 25-27 Aug., 1994.

[Le Riche and Haftka, C1992] Le Riche, R., and Haftka, R.T., *Optimization of Composite Stacking Sequence by Genetic Algorithms*, Proceedings of 33rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, CP AIAA-92-2314, Dallas, Texas, 1992.

Conferences without reviewing

[Gogu et al, C2008-1] Gogu, C., Haftka, R.T., Le Riche, R., Molimard, J. and Vautrin, A., *A Bayesian Framework for Orthotropic Elastic Constants Identification Accounting for both Error and Variability*, accepted at the 8th. World Congress on Computational Mechanics (WCCM8) and 5th. European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008), Venice, Italy, July 2008.

[Gogu et al, C2008-2] Gogu, C., Haftka, R.T., Le Riche, R., Molimard, J. and

Vautrin, A., *Illustrating the advantages of a Bayesian over a least squares approach to material properties identification*, presented at the ESM Centennial Mechanics Conference: Liviu Librescu Session, Blacksburg, VA, May 2008.

[Ginsbourger and Le Riche, C2007] D. Ginsbourger and R. Le Riche, *Utilisation d'un méta-modèle de krigeage en optimisation*, communication at the CSMA meeting on new trends in multi-disciplinary optimization, Université de Technologie de Compiègne, France, March 30th 2007, cf. <http://www.utc.fr/lrm/csma300307> (in French).

[Silva et al., C2006] G. Silva, J. Molimard, R. Le Riche and A. Vautrin, *Integrated Strategy for Identification of Material Properties using OFFMs Methods*, Photomechanics meeting, Clermont-Ferrand, France, 10 to 12th of July 2006, pp. 184-185.

[Molimard et al., C2004-2] J. Molimard, R. Le Riche, A. Vautrin, and J. R. Lee, *Application of the Open Hole Tensile Test to the Identification of the in-plane Characteristics of Orthotropic Plates*, 2nd International Conference on Composites Testing and Model Identification, Comptest 2004, Dept. of Aerospace Eng., Univ. of Bristol, England, 21-23 September, 2004.

[Saouab et al., C2003] Saouab, A., Le Riche, R. and Bréard, J., *Coupled Optimization Processes-Properties of Composite Plates*, Comptes rendus des 13ièmes Journées Nationales des Composites, JNC13, Y. Rémond et J. Lamon ed., Univ. de Strasbourg, France, 12-14 march 2003, Vol.1, pp.101-110.

[Molimard and Le Riche, C2002] Molimard, J. and Le Riche, R., *Identification de piézoviscosité en lubrification*, proceedings of the AFM (Association Française de Mécanique) meeting: Identification des propriétés mécaniques à partir de mesures de champs, LERMES/LaRAMA, univ. Clermont II, Clermont Ferrand, France, April 2002 (in French).

[Le Riche et al., C1998] Le Riche, R., Gaudin, J., and Galantini, P., *Evolutionary Approach to Multiobjective Optimization of Dimensionally Stable Composites*, Proceedings of the European Conference on Spacecraft Structures, Materials, and Mechanical Testing, Braunschweig, Germany, 4 - 6 November 1998.

[Le Riche et al., C1995-2] Le Riche, R., Knopf-Lenoir, C., and Haftka, R.T., *Optimisation de Structures Composites par Algorithmes Génétiques*, Actes de la journée PRIMECA de Toulouse, INSA de Toulouse, France, 4 July 1995.

4.2.4 Technical and contract reports

[Le Riche R. et al., T2008] Le Riche, R., Boyer, A., Silva, G., Monnatte, J. and Molimard J., *Préconception d'arcs à relâchement d'effort*, contract report Wildsteer SARL Company, May 2008 (in French).

[Le Riche R. et al., T2007-2] R. Le Riche, A. Meyer, and D. Lemosse *Optimisation de transmissions de puissance par engrenage*, 2007 contract report for the CETIM Foundation, December 2007 (in French).

[Le Riche R. et al., T2007-1] R. Le Riche, G. Pujol, X. Bay and O. Rousstant, *Optimisation en présence d'incertitudes, état de l'art*, contract report of the OMD project, ANR (National Research Agency), France, jan. 2007 (available at <http://omd.lri.fr> for consortium members, in French).

- [Pujol and Le Riche, T2007] G. Pujol and R. Le Riche, *Spécification pour les développements logiciels du projet OMD et réalisation d'une maquette*, contract report of the OMD project, ANR (National Research Agency), France, june 2007 (available at <http://omd.lri.fr> for consortium members, in French).
- [Silva et al., T2007] G. Silva, R. Le Riche, J. Molimard and A. Vautrin, *Exact and efficient interpolation using finite elements shape functions*, technical report hal-00122640, available from <https://hal.archives-ouvertes.fr/hal-00122640>, Jan. 2007.
- [Le Riche and Meyer, T2006] R. Le Riche and A. Meyer, *Optimisation de transmissions de puissance par engrenage*, 2006 contract report for the CETIM Foundation, December 2006 (in French).
- [Le Riche and Bay, T2004] R. Le Riche and X. Bay, *Régulation de planéité par critère crête-à-crête*, contract report for Vai-Clecim Company, December 2004 (in French).
- [Luersen et al., T2004] M. Luersen, R. Le Riche and E. Breier, *Identification de la position des sauts de plis lors d'un changement de tissu dans la fabrication des monopalmes*, contract report for Breier S.A.S. Company, May 2004 (in French).
- [Rakotomamonjy et al., T2004] Rakotomamonjy, A., Le Riche, R., and Gualandris, D., *Reconstruction de couple moteur par apprentissage statistique*, contract report for PSA Company (DRIA / SARA / MATE), March 2004 (in French).
- [Le Riche, T2003-3] Le Riche, R., *Identification de la diffusion dans LAMKIT*, contract report for EADS-CCR, September 2003 (in French).
- [Le Riche, T2003-2] Le Riche, R., *Assistance de maintenance et développement de LAMKIT*, contract report for EADS-CCR, Aug. 2003 (in French).
- [Le Riche, T2003-1] Le Riche, R., *Identification de cycles de vieillissement hydrique accélérés*, contract report for EADS-CCR, june 2003 (in French).
- [Le Riche et al., T2002] Le Riche, R., Le Maître, O., Luersen, M., Besson, J. and Sorbier, A., *Optimisation de palmes de nage en matériaux composites*, contract report for Breier S.A.S. Company, december 2002 (in French).
- [Rakotomamonjy et al., T2001] Rakotomamonjy, A., Le Riche, R. and Gualandris, D., *Estimation de masse de véhicule à partir de mesures d'accélération*, contract report for PSA Company, Nov. 2001 (in French).
- [Le Riche and Gaudin, T2001-2] Le Riche, R. and Gaudin, J., *Interfaçage dynamique dans LAMKIT*, contract report EADS-CCR, June 2001 (in French).
- [Le Riche and Gaudin, T2001-1] Le Riche, R. and Gaudin, J., *Etude du flambage dans LAMKIT*, contract report EADS-CCR, May 2001 (in French).
- [Le Riche and Guyon, T2001] Le Riche, R. and Guyon, F., *Dual Evolutionary Optimization*, technical report no. 01/2001, Laboratoire de Mécanique de Rouen, INSA de Rouen, France, 2001.
- [Le Riche, T2000] Le Riche, R., *Approches neuronales de l'identification de systèmes dynamiques, application au monoroue*, contract report for PSA Company, April 2000 (in French).
- [Le Riche et al., T2000] Le Riche, R., Gaudin, J. and Dutertre, C., *OPTILAM, un logiciel d'optimisation de stratifiés composites*, user's manual, Jan. 2000 (in

French).

[Vandenger et al., T2000] Vandenger, M., Lemosse, D., and Le Riche, R., *Modélisation et Optimisation de Supports de Câbles*, contract report for CES Company, Jan. 2000.

[Le Riche and Guyon, T1999] Le Riche, R., and Guyon, F., *Least Squares Parameter Estimation and the Levenberg-Marquardt Algorithm : Deterministic Analysis, Sensitivities and Numerical Experiments*, technical report no. 041/99, Laboratoire de Mécanique de Rouen, INSA de Rouen, France, 1999 (available from <http://www.emse.fr/~leriche/publications.html>).

[Le Riche and Kroo, T1998] Le Riche, R., and Kroo, I., *New Ideas for Nonlinear Control: Part 1, Numerical Tools*, technical report no. 121/97, Laboratoire de Mécanique de Rouen, INSA de Rouen, France, 1998.

[Cailletaud et al., T1996] Cailletaud, G., Lovato, G., and Le Riche, R., *Étude de l'Alliage AS5U3G avec Applications aux Culasses*, contract report for MONTUPET Company (MONTUPET-ARMINES No. 250312H), Feb. 1996 (in French).

[Le Riche et al., T1995] Le Riche, R., Cailletaud, G., and Pilvin, P., *Optimisation Topologique de Pièces Mécaniques*, contract report SNECMA Company-ARMINES No. 764621, December 1995 (in French).

[Le Riche and Cailletaud, T1996] Le Riche, R., et Cailletaud, G., *Simulation de l'Essai de Choc Thermique*, contract report GDF company - ARMINES No. E 602 850 GDF, December 1996 (in French).

[Le Riche, T1994] Le Riche, R., *Optimization of Composite Structures by Genetic Algorithms*, Ph.D. thesis, Dept. of aerospace eng., Virginia Polytechnic Institut and State University, Blacksburg, VA, 24060, Etats-Unis, Octobre 1994.

4.2.5 Popularizing texts

[Apnéa, V2007] Citation in Apnéa, *Eric Breier, le sorcier des composites*, april 2007, pp.24-27 (in French).

[La Tribune, V2006] E. Babuin, *Rodolphe Le Riche, un parcours atypique*, La Tribune - Le Progrès, 29 March 2006 (in French).

[Breier et al., T2004] E. Breier, R. Le Riche, M. Luersen, J. Monnatte, D. Lemosse et O. LeMaître, *Ils vous parlent du carbone*, Apnea, mai 2004, pp. 46-49 (in French).

4.2.6 Seminars

[Filomeno et al., S2007] R. Filomeno Coelho, R. Le Riche, P. Breitkopf and C. Vayssade, *Projet OMD : Concepts et Application à une Aile d'Avion*, invited speaker at STIC 2007, Paris, La Villette, Nov. 5-7, 2007 (in French).

[Le Riche R., S2006] R. Le Riche, *Les méthodes FORM et SORM en calcul de fiabilité*, seminar of the 3MI group, Ecole des Mines de Saint-Etienne, 15 May 2006 (in French).

[Le Riche, S2004] Le Riche, *Quoi de neuf dans les algorithmes génétiques ? Un bilan de 15 ans d'optimisation évolutionnaire*, invited speaker at the symposium

“30 ans de Modélisation Numérique à l’UTC : défis et perspectives”, Univ. de Technologie de Compiègne, France, 25-26 November 2004 (in French).

[Le Riche et al., S2003] Le Riche, R., Vinet, A. and Didierjean, S., *Identification of accelerated wet-ageing cycles*, presented at EUROMECH 453, “Internal Stresses in Polymer Composite Processing and Service Life”, Saint-Étienne, France, 1-3 Décembre, 2003.

[Le Riche, S2002-3] Le Riche, R., *Optimisation par algorithmes évolutionnaires en mécanique*, Plan pluri-formation “Simulation numérique des conséquences métallurgiques et mécaniques induites par les procédés mettant en jeu les hautes températures”, Saint Galmier, France, 6 december 2002.

[Le Riche, S2002-2] Le Riche, R., *Introduction à l’optimisation par algorithmes évolutionnaires*, SPIN seminar of the Ecole des Mines de Saint-Étienne, France, 12 novembre 2002 (in French).

[Saouab et al., S2002] Saouab, A., Le Riche, R. et Bréard, J., *Couplage Procédés-Propriétés: analyse statistique d’une plaque composite*, Journée Scientifique et Technique (JST - AMAC): Mise en Forme des Matériaux Composites, ENSAM, Paris, 3 October 2002 (in French).

[Le Riche, S2002-1] Le Riche, R., “Une comparaison de stratégies de discrimination des masses de véhicules automobiles”, seminar of the optimization group at the Ecole des Mines de Saint-Étienne, France, 21 June 2002 (cf. <http://www.emse.fr/~leriche/GTopti.html>) (in French).

[Le Riche, S2000] Le Riche, R., *Identification de lois de comportement*, invited speaker at the DMSE dept., ONERA Chatillon, France, 22 novembre 2000 (in French).

[Le Riche, S1998] Le Riche, R., *Pénalisation en calcul évolutionnaire*, talk at the 2nd journées de calcul évolutionnaire, CMI, Marseilles, France, 29 June – 1st July 1998 (in French).

[Le Riche, S1991] Le Riche, R., *Buckling Optimization of Laminates with Strain Constraints by Genetic Algorithms*, presentation at the 10th ASCE Conference on Electronic Computation, Indianapolis, Indiana, USA, 1991.

Chapter 5

Other professional activities

5.1 Project management

5.1.1 The OMD project

Since June 2006, I have lead the French national “Optimisation MultiDisciplinaire” project. This three years long project has an overall budget of 3 Meuros and was granted 1.3 Meuros by the French Agence Nationale de la Recherche (ANR). (A summary of the main acronyms used in this chapter can be found in table 5.1.) It is a national effort to foster the development of optimization in industry and academics. 14 entities are taking part to the OMD project, 10 public laboratories and 4 private companies. The project is structured into 8 research operations: metamodels for optimization, specialized metamodels in solid mechanics, specialized metamodels in fluid mechanics, optimization using multi-level variables, multi-fidelity optimization, optimization with model uncertainties, collaborative optimization and software development. Three industrial design applications are targetted: a supersonic business jet (in collaboration with Dassault Aviation), a car engine air inlet (in collaboration with Renault) and a satellite launcher (in collaboration with Astrium Space Transportation). The project developments are prototyped in the Scilab language¹ because it is an opensource clone of the popular Matlab language. Collaboration between the participants is fostered by the use of a common programming language and by communications through the collaborative web site <http://omd.lri.fr> .

I have been very involved in this project. The tasks I am in charge of comprise

- organizing the proposal and writing parts of it,
- writing a consortium agreement,
- hiring collaborators,
- organizing and managing the project web site,
- organizing periodical project meetings (three each year),
- gathering progress reports and writing syntheses,
- and presenting the project in various meetings.

A book showing the contributions of the project and a Scilab optimization toolbox are currently in preparation. We have just learned² that OMD2 (“Optimisation MultiDisciplinaire Distribuée”), a three years continuation of the OMD project oriented towards distributed implementations, was accepted by the French ANR.

¹Cf. <http://www.scilab.org> .

²on the 15th of July 2008

5.1.2 Other projects in collaboration with private companies

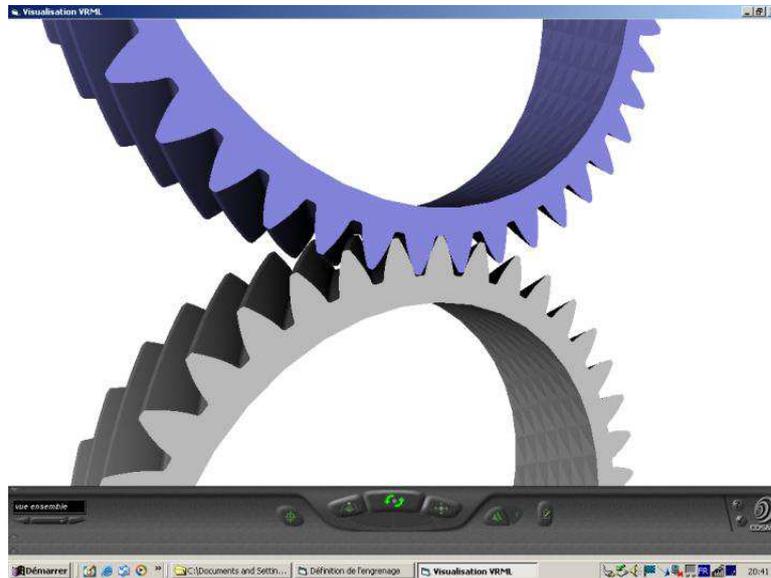


Figure 5.1: Illustration of the gears optimized in the CETIM project. The gears are analyzed with the *Filengrene* software [134].

The CETIM project. I am also currently leading a small project for the French “Centre des Etudes et Techniques de l’Industrie Mécanique” (CETIM) that aims at optimizing gears while accounting for wear. This is a complex robust design problem. The project has a grant of 50 Keuros and is carried out in partnership with the Institut National des Sciences Appliquées (INSA) de Rouen (André Meyer and Didier Lemosse), the Ecole Centrale de Lyon (Emmanuel Rigaud and Joel Perret-Liaudet) and the Ecole Nationale d’Ingénieurs de Saint-Etienne (ENISE, Joel Rech). FIG. 5.1 provides an illustration of the considered gears.

In the past, I have often been involved in joined projects with private companies. A list of these works follows:

- Pre-design of a bow with a tension release mechanism, series of student projects with a grant of 5000 euros given by Wildsteer Comp., 2008. (Cf. a drawing of the prototype in FIG. 5.2).
- Identification of instantaneous engine torque by statistical learning, contract for PSA (2004, in parntership with A. Rakotomamonjy, INSA de Rouen, 15250 euros). The results of this project have been patented ([Gualandris et al., P2007]).
- Estimation of the cargo from online acceleration measures, contract for PSA (2003, in partnership with A. Rakotomamonjy, INSA de Rouen, 15250 euros).
- Swimming monofin optimization, contract for Breier S.A., (7620 euros in 2002, 2000 euros in 2003, managed by ARMINES).
- Identification of hygro-thermal cycles for accelerated wet-ageing in composites, contract for EADS-CCR, (2002, 3811 euros, managed by INSA de Rouen).

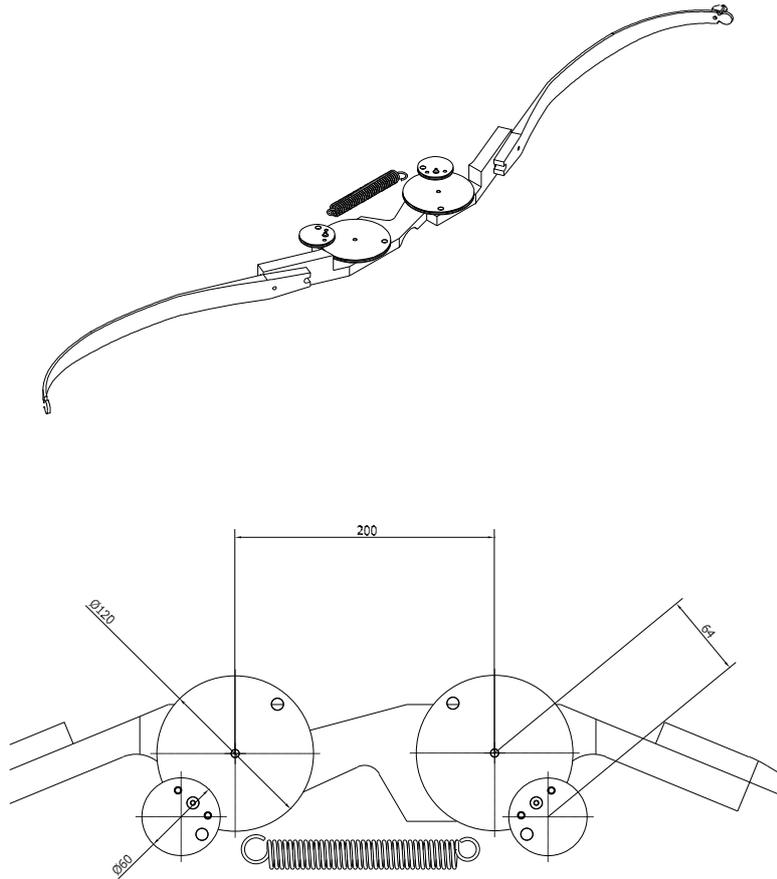


Figure 5.2: Drawing of the bow prototype that was built: an inner pulley with two eccentricities and a spring create a mechanism that releases tension in the bow string as it is pulled. This is a feature of modern compound bows.

- Identification of moisture diffusion in composites, contract for EADS-CCR, (2002, 3811 euros, managed by INSA de Rouen).
- Consulting for the development of the LAMKIT software, EADS-CCR, 2003, 7620 euros, managed by Centrale Lyon Innovation.
- Dynamic interfacing in LAMKIT, contract for EADS-CCR, 2001, 4573 euros, managed by INSA de Rouen.
- Buckling analysis in LAMKIT, contract for EADS-CCR, 2001, 4573 euros, managed by INSA de Rouen.
- Dynamic systems representation by neural networks, contract for PSA, 2000, 7620 euros, managed INSA de Rouen. The methods developed for PSA were “industrialized” in 2001, i.e., they were rewritten and became part of PSA’s design offices.
- Optimization of cable carrying structures, contract for Compagnie d’Electricité de la Seine (CES), 2000, 3050 euros, managed by INSA de Rouen.
- Simulation of the thermal choc test, contract for GDF, 38000 euros, 1996, managed by ARMINES.
- Identification of the AS5U3G alloy constitutive law, contract for Montupet, 1996, 38000 euros, managed by ARMINES.
- Shape optimization of fan disks, contract for SNECMA, 76200 euros, 1995, managed by ARMINES.

5.2 Teaching

As permanent research assistant with the CNRS, teaching is not my primary assignment. Nevertheless, I think teaching is intimately linked to researching, firstly because it is an efficient way to diffuse acquired knowledge to the rest of the society, and secondly because it helps the researcher to put his activity into perspective. Ever since I started my PhD in 1991, I have always devoted some of my time to teaching. The topics I have taught range from strength of material to global optimization and include programming and identification of constitutive laws in solid mechanics. I have taught students of levels varying from first year undergraduate to Master’s students and continuing education. More precisely, in the French educational system, they were students from DEUG, IUT, all years of engineering college, DEA (i.e., Master’s degree) and CACEMI. In total, about 450 hours were spent teaching classes. In addition, I have always tutored projects in the university and professional trainings (see section 5.2.2).

5.2.1 Classes taught

The classes I have taught are summarized in Table 5.2 where the year, the place, the topic, the academic level and the number of students attending the class are listed. The class topics are described in more details in the following syllabus. The acronyms used are explained in Table 5.1.

1Y,2Y,3Y	1 st , 2 nd , 3 rd year undergraduate engineering college
ANR	Agence Nationale de la Recherche
CACEMI	Centre d'Actualisation des Connaissances et d'Etude des Matériaux Industriels
CETIM	Centre des Etudes et Techniques de l'Industrie Mécanique
CIME	option Connaître et Imaginer les Matériaux pour Entreprendre de 3 ^{ème} année de l'ENSM-SE
CNAM	Conservatoire National des Arts et Métiers
CNRS	Centre National de la Recherche Scientifique
DEUG	Diplôme d'Etudes Universitaires Générales
DLR	Deutsches Forschungsinstitut für Luft und Raumfahrt
EADS	European Aeronautic Defence and Space company
EADS-CCR	Centre Commun de Recherches d'EADS (now EADS-Innovative Works)
EF	Eléments Finis
ENISE	Ecole Nationale d'Ingénieurs de Saint-Etienne
ENSM-SE	Ecole Nationale Supérieure des Mines de Paris
FC	Formation Continue
ICM	Ingénieur Civil des Mines
INSA	Institut National des Sciences Appliquées
IPSI	Institut pour la Promotion des Sciences pour l'Ingénieur
IUT	Institut Universitaire de Technologie
LDC	Lois de Comportement
LdV	Pôle Universitaire Léonard de Vinci
NFL	No Free Lunch
OMD	Optimisation MultiDisciplinaire (project)
PCA and PSA	Peugeot Citroen Automobiles
Prof. project	Professional project during the 3rd year of eng. college, 120 <i>h</i> .
3Y training	3rd year professional training, that concludes the engineering college (Bachelor of Science).
UJM	Université Jean Monnet
UF	University of Florida at Gainesville
UTC	Université de Technologie de Compiègne

Table 5.1: Main acronyms.

Year	University	Topic	Level	Hours	No. Students
2007-2008 and 2006-2007 and 2005-2006	ENSM-SE	Global Optim. 1 [GO1]	2Y	21	15
	UJM+ENSM-SE	Global Optim. 2 [GO2]	Master	10	10
2004-2005	ENSM-SE	Optim. Context [OC]	1Y	0,5	100
	ENSM-SE	Global Optim. 1 [GO1]	2Y	21	18
	UJM+ENSM-SE	Global Optim. 2 [GO2]	Master	10	10
2003-2004 and 2002-2003	ENSM-SE	Global Optim. 1 [GO1]	2Y	2 × 21	≈ 20
	ENSM-SE	Struct. Optim. 1 [SO1]	3Y	2 × 6	≈ 20
	ENSMP	Struct. Optim. 2 [SO2]	3Y	6	≈ 15
2001-2002	ENSM-SE	Global Optim. 1 [GO1]	2Y	21	≈ 20
	INSA Rouen	Struct. Optim. 2 [SO2]	DEA	20	≈ 20
	ENSMP	Struct. Optim. 2 [SO2]	3Y	6	≈ 15
	ENSMP	Struct. Optim. 2 [SO2]	Master	12	≈ 15
2000-2001	INSA Rouen	Struct. Optim. 2 [SO2]	DEA	20	≈ 20
	CACEMI/CNAM	Identif. Const. Laws [IdCL]	FC	5	≈ 10
	ENSMP	Struct. Optim. 2 [SO2]	Master	6	≈ 15
1999-2000	INSA Rouen	Struct. Optim. 2 [SO2]	DEA	20	≈ 20
	ENSMP	Optim. in Zset [ZsetC]	FC	8	≈ 10
	ENSMP	Identif. Const. Laws [IdCL]	3Y	6	≈ 15
1998-1999	INSA Rouen	Programming [C++]	1Y	33	≈ 20
	ENSMP	Optim. in Zset [ZsetC]	FC	8	≈ 10
	ENSMP	Identif. Const. Laws [IdCL]	3Y	6	≈ 15
	UTC	Evol. Optim. [EA]	FC	3	≈ 10
	LdV	Optim. Evolut. [EA]	Master	6	≈ 15
1997-1998	INSA Rouen	Programming [C++]	1Y	16	≈ 20
	IPSI	Identif. Const. Laws [IdCL]	FC	3	≈ 30
	LdV	Evol. Optim. [EA]	Master	6	≈ 15
1996-1997	IUT Champs/Marne	Programming [ProgC]	IUT	60	≈ 30

Table 5.2: Summary of classes taught with the year, topic, number of hours and number of students. Cf. syllabus for a description of the class material and Table 5.1 for a glossary of the abbreviations.

Syllabus of the classes taught

<i>C++</i>	C++ Programming : basic commands, pointers, introduction to classes and objects, polymorphism, inheritance, dynamic inheritance.
<i>OC</i>	Optimization put in a broader Context : uses in modelling, identification, design. Main softwares. Practical remarks.
<i>IdCL</i>	Identification of constitutive laws : introduction, identifiability (unicity, stability), non-linear least squares, Levenberg-Marquardt algorithm.
<i>GO1</i>	Global Optimization 1 : introduction to global optimization, methods classification, deterministic methods, enumerative methods, evolutionary algorithms. 9 hours practice in computer lab.
<i>GO2</i>	Global Optimization 2 : reminder of main methods, theory of stochastic perturbations (variance control, use of populations), kriging and optimization.
<i>SO1</i>	Structural Optimization 1 : introduction to structural design, reminder on optimality conditions, application to the minimization of a system total potential energy and to optimal design, mathematical programming (gradient and Gauss-Newton), introduction to the identification of models.
<i>SO2</i>	Structural Optimization 2 : introduction to structural design, optimality conditions, application to the minimization of a system total potential energy and to optimal design, mathematical programming (order 0, 1 and 2), non-linear least squares and model identification.
<i>ProgC</i>	C Programming : basic commands, compilation, loops, functions, input-output, pointers. Application to a truss simulation.
<i>ZsetC</i>	Class on the optimization module of the finite elements software Zset. Behavior laws simulation. Input files. Presentation of the implemented optimizers (Nelder-Mead, evolutionary algorithm, sequential quadratic programming). Application to the identification of elasto-visco-plastic behavior laws.

5.2.2 Students advising

Graduate students

I have co-advised six doctorate and three masters' students:

- Christian Gogu is in his 2nd year of a PhD about the use of response surfaces to identify and optimize structures. He is co-advised by Raphael Haftka from Univ. of Florida at Gainesville, Jérôme Molimard, Alain Vautrin and myself (Ecole des Mines de Saint-Etienne).
- Gustavo Silva will defend his PhD in the Fall of 2008. He is working on the identification of elastic material properties and other experimental parameters from full-field measures. This PhD is co-advised by Jérôme Molimard, Alain Vautrin and myself.
- David Ginsbourger will defend his PhD on multiple surrogates, kriging and optimization in the Fall of 2008. His work is co-advised by Laurent Carraro, Olivier Roustant and myself at the Ecole des Mines de Saint-Etienne.
- Marco Luersen has defended his PhD entitled *GBNM, un algorithme d'optimisation par recherche directe – Application à la conception de monopalmes de nage* in December 2004 at the INSA de Rouen, France. I served as unique advisor (administrative Director: Eduardo Souza de Curci).

- Laurent Grosset has defended his PhD entitled *Statistical optimization of composite structures* in September 2004. I have advised 50% of the work in collaboration with Raphael Haftka from the Univ. of Florida at Gainesville. Alain Vautrin was the French administrative Director.
- I currently co-advise Christophe Mediavilla's Master thesis with Michel Ravachol (Dassault Aviation). The thesis concerns the robust design of a supersonic business jet.
- I have co-advised 30% of Gustavo Silva's Master thesis with J. Molimard and A. Vautrin in 2004. The work concerned the identification of elastic composite properties from full-field measures.
- In collaboration with Alain Vinet from EADS-CCR, I co-advised Franck Pellé's Master thesis, defended in September 2003 and entitled *Optimisation des matériaux composites stratifiés en fluage* (optimization of composite materials for creep).

Moreover, I was a member of the following PhD juries:

- Laurence Giraud Moreau, *Optimisation des systèmes mécaniques : couplage des méthodes déterministes et évolutionnaires pour les problèmes en variables mixtes*, (Optimization of mechanical systems: coupling determinist and evolutionary methods for mixed variables problems,) November 1999, Univ. de Technologie de Troyes.
- Chung-Hae Park, *Simultaneous optimization of manufacturing and design of composite structures*, June 2003, Ecole des Mines de Saint-Étienne and Seoul National University, South Korea.
- Papa Aldemba Faye, *Couplage algorithme génétique - code éléments finis pour le dimensionnement de structures en matériaux composites*, (Genetic algorithm-finite elements coupling for designing composite structures,) March 2004, Univ. Clermont Ferrand 2 and ONERA Lille.
- Jung Riul Lee, *Application of Laser Interferometric Techniques to the Experimental Analysis of Materials and Structures - Special case of Composites*, Ecole des Mines de Saint-Étienne, March 2004.
- Marco Gigliotti, *Modelling, simulation and experimental assessment of hygrothermoelastic behaviour of composite laminated plates*, Ecole des Mines de Saint-Étienne, December 2004.
- Antoine Merval, *Application de modèles réduits à l'optimisation multiniveaux d'une structure aéronautique*, (Reduced models for the multi-level design of an airplane structure), Institut Supérieur de l'Aéronautique et de l'Espace, June 2008.

Undergraduate students

I now summarize my tutoring activity related to

- the students professional training in external companies (3rd and 2nd year undergraduate professional training),
- and the professional project, the long project and the short project, all of which are taking place in the university during the undergraduate studies.

	3Y train- ing	2Y train- ing	Prof. proj.	long proj.	short proj.	total / year (h. tutoring)
2007-2008	1	1	0	2	0	26
2006-2007	1	0	1	2	1	43
2005-2006	1	1	0	2	1	33
2004-2005	3	2	0	2	1	43
2003-2004	2	2	1	1	4	62
2002-2003	2	1	0	0	2	24
2001-2002	0	0	0	0	0	0
2000-2001	0	0	1	1	0	22
1999-2000	2	0	1	0	0	8
1998-1999	2	0	0	1	0	18
1997-1998	1	0	1	0	0	16
Total :						295

Table 5.3: Undergraduate students advising activity per year, in tutoring hours.

The projects I tutored dealt with composite materials, structural design, optimization, data mining and programming. Table 5.3 provides an estimation of my undergraduate tutoring activity in equivalent tutoring hours (or “heures équivalent TD”). At the ENSM-SE, tutoring hours are defined as follows: 20% of the actual duration of short projects; the short and long projects, which last 50 and 35 hours respectively, are associated to 10 and 7 h. tutoring; 10% of the actual length of the 120 hours long professional project, i.e., 12h. tutoring; 4 and 2h. tutoring for the 3rd and 2nd year professional trainings, respectively. According to this scale, I have tutored 295 hours in total (not accounting for graduate students advising).

5.3 Reviewing and other administrative tasks

5.3.1 Reviewing

Journals:

- Review editor at Structural and Multidisciplinary Optimization since 2006.
- Structural and Multidisciplinary Optimization, about 2 articles each year since 2001.
- Journal of Global Optimization, 1 article, 2008.
- Revue des Sciences et Technologies de l’information (Hermes publ., in French), 3 articles, 2005.
- Int. Journal of Numerical Methods in Engineering, 2 articles, 1998 and 1999.
- IEEE Transactions on Evolutionary Computation, 10 articles from 1998 to 2004.
- European Journal of Finite Elements, 4 articles from 1998 to 2000.

Conferences:

- Congrès Français de Mécanique (CFM05), Univ. de Technologie de Troyes, France, 29 August - 2 September 2005, 10 articles.

- 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII), Birmingham, England, 18-22 September 2004, 3 articles.
- Artificial Evolution 03, Marseilles, France, 27-30 October 2003, 4 articles.
- 7th Intl Conference on Parallel Problem Solving from Nature, Granada, Spain, 7 - 11 September 2002, 3 articles.
- 9th World Symposium on Biomechanics and Medicine in Swimming, Saint-Étienne, France, 21-23 June 2002, 2 articles.
- Artificial Evolution 01, Le Creusot, France, 29-31 October 2001, 3 articles.
- The Sixth International Conference on Parallel Problem Solving from Nature, Paris, 16-20 September 2000, 3 articles.
- Genetic and Evolutionary Computation Conference, GECC-99, Joint meeting of the 8th Int. Conference on Genetic Algorithms and the 4th Annual Genetic Programming Conference, Orlando, Florida, 13-17 July 1999, 3 articles.
- The 1999 Congress on Evolutionary Computation, CEC99, Mayflower Hotel, Washington DC, 6-9 July 1999, 3 articles.

Projects:

- Agence Nationale de la Recherche, France, 2007, 1 project.
- Université Numérique Ingénierie et Technologie, France, 2006, 1 project.
- Fonds de recherche sur la nature et les technologies, Quebec, Canada, 2006, 1 project.

5.3.2 Conferences and seminars organization

- Organizer of the optimization workshop in the Congrès Français de Mécanique, CFM'05, Université de Technologie de Troyes, 2005.
- Organizer of the “identification work group” at the ENSM-SE, two meetings in 2004 and 2005.
- Organizer of the “Z-Set work group” at the ENSM-SE (one day training), 2004.
- Organizer of the “optimization work group” at the ENSM-SE, 7 meetings in 2002 and 2003.
- Organizer of the seminars at the Laboratoire de Mécanique de Rouen from 1998 to 2000.

5.3.3 Other administrative tasks

- Member of the hiring committee of the Université Jean Monnet (Saint-Etienne, France) in the mechanical engineering section since 2007.
- Elected member of the scientific board of the INSA de Rouen from 1998 to 2000.

5.4 Curriculum Vitae

Le Riche Rodolphe

optimization and identification
of mechanical systems

Permanent research assistant, CNRS
D.O.B. Sept. 4th 1968
2 children

4, rue Hovelacque, 42100 St.-Étienne
France
Tel : (33)(0)477420023
e-mail : leriche@emse.fr

- 2002-2008 **Permanent research assistant, 1st class, French National Research Center (CNRS),**
Ecole des Mines de Saint-Étienne, France.
Research : leader of the French multi-disciplinary optimization project (ANR/OMD, 3 Meuros of budget); co-author of 2 patents, 17 journal articles, 14 contracts with private companies. Advisor of 5 PhD and 2 Master students.
Teaching : global optimization, structural optimization.
- 1998-2001 **Permanent research assistant, 2nd class, CNRS.**
Lab. de Mécanique de Rouen, INSA de Rouen, France.
Research : co-author of the LAMKIT software, 4 journal articles and 4 contracts. 3 month training period at Stanford Univ. with I. Kroo.
Teaching : structural optimization, constitutive law identification, C++, evolutionary algorithms.
- 1996-1997 **Post-Doctorate** Ecole des Mines de Paris (with G. Cailletaud).
Research : co-author of the Zset finite elements software, 2 journal, 3 contracts.
Teaching : C/C++, constitutive law identification in solids mechanics, evolutionary algorithms.
- 1995 Military Service at the Secrétariat Général à la Défense Nationale, Paris.
Technological survey on military and space launchers.
- 1991-1994 **Ph.D. of Aerospace Engineering**, Virginia tech, USA
Subject : evolutionary optimization of composites structures (R.T. Haftka, advisor). Co-author of 2 journal articles. 1 month training period at NASA Langley (with J. Starnes). Teaching assistant.
- 1986-1991 **Mechanical Eng. Bachelor**, Université de Technologie de Compiègne, France. Majors: materials and design.
6 month professional training with Elsnerdruck, Berlin, Germany.
- 1986 **Baccalauréat of science**, Pontoise, France.
-

Acknowledgements

Merci à Eliaz, Silvan et Valou pour leur patience quand je n'étais pas tout à fait là ainsi que mes parents, Odile et Christian, pour leur soutien constant.

Merci aussi à mes collègues sans lesquels ce travail n'aurait pas pu exister : Rafi Haftka qui sait si bien allier amitié et travail, Gilbert Touzot pour son enthousiasme, Fred Guyon pour les séances de maths-pinard, Alain Rakotomamonjy pour le boulot bonne humeur, André Meyer dont j'ai adoré partager le bureau, Xavier Bay pour sa disponibilité et son amitié, Georges Cailletaud pour les zébulonesques années, Jérôme Molimard pour ses jeux de mots, mes ex-doctorants de choc devenus amis (Laurent Grosset, Marco Luersen), Albert Boyer pour son amitié et son aide, Marc Schoenauer et Michèle Sebag pour l'inspiration qu'ils véhiculent, Rebecca Hudson pour la relecture et son humour, Agnès Lépine pour ses boujous, Bernadette Degache pour son humanité, Alain Vautrin pour sa souplesse, mes doctorants (David Ginsbourger, Gustavo Silva, Christian Gogu), la tribu 3MI (Eric Touboul, Olivier Roustant, Céline Helbert, Delphine Dupuy, Victor Picheny, Laurent Carraro), Steph Canu pour ses v'lal, Didier Lemosse, Olivier LeMaître, Abdou Saouab, Joel Bréard, Jocelyn Gaudin, Nagendra Somanath, Eric Breier, Jacques Besson, Ronald Foerch, Frédéric Feyel, Catherine Vayssade, Piotr Breitkopf, Peter Harrison, Samy Missoum.

List of Figures

1	Identification of vehicle mass, loads and torque from online measures	xxi
2	Identification of the elastic properties of a composite plate with a hole.	xxii
1.1	Example of function landscape and evolutionary optimizer population	2
1.2	Generic flowchart of an evolutionary algorithm	4
1.3	Genotypical and phenotypical spaces	5
1.4	Sphere-tunnel function	9
1.5	initialization and <i>FDC</i>	15
1.6	Function seen as a random process	17
1.7	Idealized EDA convergence in 1D	18
1.8	Selected points and marginal densities product	19
1.9	Flowchart of a segregated genetic algorithm	23
1.10	Example of dual and approximate dual functions, problem with a saddle point.	26
1.11	Dual evolutionary optimizer	28
1.12	Example of dual and approximate dual functions, problem without a saddle point.	29
1.13	Evolutionary optimization based on f_p .	30
1.14	Comparison of EA convergences for varying amounts of penalty, hoop problem	31
1.15	Restarts and convergence tests linking in GBNM	33
1.16	Carbon monofin by Breier©.	34
1.17	Swimmer and fin model	34
1.18	Optimal stiffness distribution for various swimmers	35
1.19	Modal FE analysis of a fin for identifying longitudinal thickness distribution.	36
1.20	Comparison of identified monofin thickness distributions from different formulations.	37
1.21	PI and EI surfaces of the Branin-Hoo function	40
1.22	Comparison between EI and 2-EI	42
1.23	Comparison of the q -EI achieved by constants, LHS and uniform random samples	44
2.1	Composite plate coding	47
2.2	Examples of crossovers for composite laminates	49
2.3	Generic flowchart of a DDOA algorithm.	51
2.4	Reliability of the GA, UMDA and DDOA on the extensional/flexural design problem, $n = 12$.	54
2.5	Coding of a fan disk shape with B-splines	55
2.6	Example of penalized generalized biological growth	56
2.7	Optimum fan disk shape obtained by coupling an EA with the penalized generalized biological growth	57

3.1	Parameters definitions: x are the design variables, U the random variables, θ the numerical model parameters and y its output.	60
3.2	Identification of experimental parameters	63
3.3	Displacement field u generated by a rigid body rotation	64
3.4	Distance maps of a plate with a hole and local damage	65
3.5	Bayes vs. Least-squares, truss example	67
5.1	Illustration of the gears optimized in the CETIM project.	98
5.2	Bow with tension release mechanism.	99

List of Tables

1.1	The Darwinian metaphor.	3
4.1	Number of publications per year	86
5.1	Main acronyms.	101
5.2	Summary of classes taught.	102
5.3	Undergraduate advising activity	105