OPTIMIZATION OF COMPOSITE STRUCTURES
BY ESTIMATION OF DISTRIBUTION ALGORITHMS

By

LAURENT GROSSET

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2004
I dedicate this work to my parents.
ACKNOWLEDGMENTS

I want to express my gratitude to Dr. Raphael T. Haftka and Dr. Rodolphe Le Riche for giving me the opportunity to complete this dual Ph.D. I thank them for the time and effort they spent to make this joint-degree program possible. I also thank them for their excellent scientific guidance and for their never-ending enthusiasm to explore new areas. Working with two advisors, and being part of two teams has been a great source of inspiration and an enriching human experience.

I would also like to thank Drs. Fortunier, Kim, Pardalos, Sankar, Schoenauer, and Vautrin for agreeing to be members of my Ph.D. supervisory committee and for taking the time to review this dissertation.

I am grateful for the friendship of my colleagues in the Structural and Multi-disciplinary Optimization Research Group, Amit Kale, Melih Papila, Xueyong Qu, Palani Ramu, Raluca Rosca, Jaco Schutte, Satchi Venkataraman, who made my studies in Florida a pleasurable experience, and of the MeM department in the École des Mines, Pierre Celle, Bernadette Degache, Sylvain Drapier, Marco Gigliotti, Jihe'd Jedidi, Jerôme Molimard, Joël Monmatte, Benoit Serre, Stéphane Vacher: they made life in Saint-Étienne as colorful as in Florida.

Finally, I want to thank my family for trusting and supporting me during these four years.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>ix</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>LIST OF ACRONYMS</td>
<td>xv</td>
</tr>
<tr>
<td>LIST OF SYMBOLS</td>
<td>xvi</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>xviii</td>
</tr>
<tr>
<td>Chapter 1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Optimization of composite laminates</td>
<td>3</td>
</tr>
<tr>
<td>1.2 From genetic algorithms to estimation of distribution algorithms</td>
<td>7</td>
</tr>
<tr>
<td>1.3 Goals of this research</td>
<td>13</td>
</tr>
<tr>
<td>Chapter 2 ESTIMATION OF DISTRIBUTION ALGORITHMS</td>
<td>15</td>
</tr>
<tr>
<td>2.1 Preliminary: stochastic model of a deterministic function</td>
<td>15</td>
</tr>
<tr>
<td>2.2 Estimation of distribution algorithms</td>
<td>18</td>
</tr>
<tr>
<td>2.2.1 General principle</td>
<td>18</td>
</tr>
<tr>
<td>2.2.2 Illustration</td>
<td>19</td>
</tr>
<tr>
<td>2.3 The general estimation of distribution algorithm</td>
<td>20</td>
</tr>
<tr>
<td>2.4 Selection schemes</td>
<td>22</td>
</tr>
<tr>
<td>2.5 Estimating the distribution of promising points: theoretical issues</td>
<td>26</td>
</tr>
<tr>
<td>2.5.1 The conservation law for generalization performance</td>
<td>26</td>
</tr>
<tr>
<td>2.5.2 Simple example</td>
<td>27</td>
</tr>
<tr>
<td>2.5.3 The bias versus variance compromise</td>
<td>28</td>
</tr>
<tr>
<td>2.5.4 Assessing the accuracy of an estimate</td>
<td>29</td>
</tr>
<tr>
<td>2.6 Estimating the distribution of promising points in practice</td>
<td>31</td>
</tr>
<tr>
<td>2.7 Estimation of distribution algorithms and other stochastic algorithms</td>
<td>32</td>
</tr>
</tbody>
</table>
3 THE UNIVARIATE MARGINAL DISTRIBUTION ALGORITHM . . . 35
  3.1 Algorithm .......................................................... 35
  3.2 Study of the original UMDA ........................................ 37
    3.2.1 Problem description ......................................... 38
    3.2.2 Population size and selection pressure .................... 39
    3.2.3 Dimensionality ............................................... 46
  3.3 Investigation of possible modifications to UMDA: memory and mu-
    tation ............................................................... 58
    3.3.1 Memory ....................................................... 58
    3.3.2 Mutation .................................................... 60
    3.3.3 Bound on the probabilities .................................. 63
    3.3.4 Elitism ...................................................... 64
    3.3.5 Conclusion of the parameter study ......................... 66
  3.4 Comparison on three problems .................................... 66
    3.4.1 Presentation of the algorithms ............................... 67
    3.4.2 Constrained maximization of the first natural frequency .. 67
    3.4.3 Minimize $A_{66}$ ............................................. 72
    3.4.4 Strength maximization ....................................... 74
  3.5 Conclusion .......................................................... 76

4 THE DOUBLE-DISTRIBUTION OPTIMIZATION ALGORITHM . . . 77
  4.1 Motivation .......................................................... 77
  4.2 Principles ........................................................ 80
    4.2.1 Identification of dependencies and data simplification .... 80
    4.2.2 Sampling from two distributions ............................. 82
  4.3 The double-distribution optimization algorithm ................ 87
    4.3.1 General algorithm ......................................... 87
  4.4 Application to composite optimization .......................... 88
    4.4.1 DDOA for composites ....................................... 88
    4.4.2 Issues associated with the representation of the auxiliary
          variable distribution ....................................... 91
    4.4.3 2D didactic example ....................................... 93
    4.4.4 Extensional problem ..................................... 94
    4.4.5 Extensional-flexural problem ............................... 100
    4.4.6 Strength problem ......................................... 104
    4.4.7 A comparison of diversity in UMDA and DDOA ............ 106
  4.5 Performance with optimized parameters .......................... 115
    4.5.1 Parameter study and best setting ......................... 115
  4.6 Generalization of DDOA to continuous optimization problem ..... 119
    4.6.1 Problem description ..................................... 119
    4.6.2 The algorithms .......................................... 120
    4.6.3 Results and discussion .................................. 121
    4.6.4 Improvement to the algorithm ............................. 125
4.7 Conclusion .............................................. 127

5 CONCLUSION ............................................. 129

5.1 Summary of the findings .......................... 129
  5.1.1 Incorporation of variable dependencies through physics-based auxiliary variables .......................... 130
  5.1.2 Control of the diversity .......................... 131
  5.1.3 Combination of distribution-based and directional search mechanisms ........................................ 132

5.2 Potential avenues for future research .......... 133

6 RÉSUMÉ ................................................... 134

6.1 Optimisation de stratifiés composites .......... 134
6.2 Optimisation statistique .......................... 135
6.3 Contenu de la thèse ............................... 139
6.4 Conclusions .......................................... 144

RELATED CONTENT

REFERENCES ................................................. 146

APPENDICES

A COMPOSITE LAMINATE OPTIMIZATION ........... 153

A.1 Basic mechanics of laminated plates ............. 153
  A.1.1 Stiffness matrices ................................ 153
  A.1.2 Lamination parameters .......................... 155

A.2 Mechanical properties used in this work ......... 159
  A.2.1 Poisson’s ratio .................................... 159
  A.2.2 Coefficient of thermal expansion ............... 160
  A.2.3 First natural frequency ........................ 162
  A.2.4 Strength ........................................... 162

B BANDWIDTH SELECTION FOR DDOA .............. 164

B.1 Kernel density estimation .......................... 164
B.2 Determination of the bandwidth ................. 166
  B.2.1 Effect of $\sigma$ on the accuracy .................. 166
  B.2.2 Maximum likelihood method ...................... 169

B.3 Application in the context of DDOA ............ 169
B.4 Effect of the bandwidth: empirical results .... 171

C LAMINATION PARAMETER PROPORTIONAL ACCEPTANCE: EXPERIMENTAL RESULTS ............. 174
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2–1</td>
<td>Sample points, $\mu = 20$</td>
<td>27</td>
</tr>
<tr>
<td>3–1</td>
<td>Material properties of graphite-epoxy</td>
<td>39</td>
</tr>
<tr>
<td>3–2</td>
<td>Mean best fitness at the fifth generation</td>
<td>41</td>
</tr>
<tr>
<td>3–3</td>
<td>Material properties of graphite-epoxy</td>
<td>68</td>
</tr>
<tr>
<td>3–4</td>
<td>Material properties of glass-epoxy</td>
<td>74</td>
</tr>
<tr>
<td>4–1</td>
<td>Characteristics of the three algorithms</td>
<td>116</td>
</tr>
<tr>
<td>D–1</td>
<td>Material properties of graphite-epoxy and glass-epoxy</td>
<td>178</td>
</tr>
<tr>
<td>D–2</td>
<td>Parameters associated with the genetic operators</td>
<td>189</td>
</tr>
<tr>
<td>D–3</td>
<td>Optimum designs of the minimization of the composite objective function</td>
<td>190</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>2–1</td>
<td>Stochastic model of a function</td>
<td>16</td>
</tr>
<tr>
<td>2–2</td>
<td>1D objective function</td>
<td>20</td>
</tr>
<tr>
<td>2–3</td>
<td>Convergence of the distribution $p(x)$</td>
<td>20</td>
</tr>
<tr>
<td>2–4</td>
<td>General Estimation of Distribution Algorithm</td>
<td>22</td>
</tr>
<tr>
<td>2–5</td>
<td>Selection as a noisy classification task</td>
<td>25</td>
</tr>
<tr>
<td>2–6</td>
<td>Estimating the distribution from a sample</td>
<td>27</td>
</tr>
<tr>
<td>3–1</td>
<td>Basic Univariate Marginal Distribution Algorithm</td>
<td>37</td>
</tr>
<tr>
<td>3–2</td>
<td>Fitness landscape $A_{11}$</td>
<td>39</td>
</tr>
<tr>
<td>3–3</td>
<td>Evolution of the mean best fitness</td>
<td>41</td>
</tr>
<tr>
<td>3–4</td>
<td>Reliability of the optimization for various combinations of populations</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>sizes as a function of the number of iterations</td>
<td></td>
</tr>
<tr>
<td>3–5</td>
<td>Reliability of the optimization for various combinations of populations</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>sizes as a function of the number of evaluations</td>
<td></td>
</tr>
<tr>
<td>3–6</td>
<td>Evolution of the probabilities for $\theta_1 \equiv x_1$, $A_{11}$ max problem</td>
<td>45</td>
</tr>
<tr>
<td>3–7</td>
<td>Ordering convention for a balanced symmetric laminate.</td>
<td>48</td>
</tr>
<tr>
<td>3–8</td>
<td>Fitness landscape of the first natural frequency $f_1$</td>
<td>48</td>
</tr>
<tr>
<td>3–9</td>
<td>Expected convergence time of the stochastic hill-climber for the max</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>$A_{11}$ and vibration problems</td>
<td></td>
</tr>
<tr>
<td>3–10</td>
<td>Number of analyses until the average maximum fitness reaches 98% of the optimal fitness, $\max A_{11}$ problem</td>
<td>54</td>
</tr>
</tbody>
</table>
3–11 Number of analyses until the average maximum fitness reaches 98% of the optimal fitness, vibration problem .......................... 55

3–12 Evolution of the probability distributions. .............................. 56

3–13 Comparison of the performance of SHC on the Max A_{11} problem and the vibration problem. ........................................ 57

3–14 Effect of memory on UMDA .............................................. 59

3–15 Influence of mutation on the reliability for two UMDA variants . ... 62

3–16 Effect of mutation for $p_m = 0.005$ .................................. 63

3–17 Effect of imposing a lower bound on the probabilities on the reliability. 64

3–18 Effect of an elitist strategy on reliability. .............................. 65

3–19 Mean best fitness and reliability for SHC, UMDA, and GA for the constrained maximization of the first natural frequency of a laminated plate ($n = 8$) ..................................................... 70

3–20 Mean best fitness and reliability for SHC, UMDA, and GA for the constrained maximization of the first natural frequency of a laminated plate ($n = 15$) ............................................................. 71

3–21 Evolution of the marginal distribution $p(x_1)$, UMDA, max $A_{66}$ problem. 73

3–22 Reliability and mean best fitness of UMDA and SHC for the strength problem ................................................................. 75

4–1 Selected points and univariate distribution for the constrained vibration problem .............................................................. 78

4–2 Interpretation of variable dependencies as the joint action of hidden variables ............................................................... 80

4–3 Incorporation of variable dependencies through auxiliary variables. ... 82

4–4 Probabilistic acceptance scheme: variance reduction ................................. 84

4–5 Influence of the two-distribution scheme on the variance ................................................................. 86

4–6 Flowchart of the double-distribution optimization algorithm. ................ 87
4–7 Effect of the two-distribution sampling procedure .......................... 93
4–8 Fitness landscape of the penalized $A_{22}$ for $n = 2$ ........................... 95
4–9 Mean maximum fitness and reliability for the constrained max $A_{22}$ problem, $n = 6$ (50 runs). ..................................................... 98
4–10 Mean maximum fitness and reliability for the constrained max $A_{22}$ problem, $n = 12$ (50 runs). ................................................. 99
4–11 Fitness landscape of the penalized CTE problem for $n = 2$. .......... 101
4–12 Mean maximum fitness and reliability for the constrained min CTE problem, $n = 6$. ................................................................. 102
4–13 Mean maximum fitness and reliability for the constrained min CTE problem, $n = 12$ (50 runs). ................................................. 103
4–14 Mean best fitness and reliability for the strength problem, $n = 12$ (50 runs) ................................................................. 106
4–15 Diversity in the criterion domain for UMDA and DDOA. .......... 109
4–16 Diversity in the variable domain for UMDA and DDOA. .......... 110
4–17 Distribution of the fitness of the best solution at each iteration for UMDA and DDOA. ................................................................. 111
4–18 Effect of mutation for UMDA and DDOA ................................. 113
4–19 Effect of bounds on the probability distributions $p(\theta_k)$ for UMDA and DDOA ................................................................. 114
4–20 Compared performances of the optimized GA, UMDA, and DDOA. 117
4–21 Best schemes for seven population sizes, for GA, UMDA, and DDOA. 118
4–22 Contours of the penalized fitness function for $n = 2$ ................. 120
4–23 Evolution of the best fitness for cUMDA and cDDOA, $n = 2$ (average over 50 runs). ................................................................. 122
4–24 Convergence of the primary and auxiliary distributions for cDDOA, $n = 2$ (50 runs) ................................................................. 123
D–5 Addition and deletion ........................................... 188

D–6 Sandwich structure of the optimum design ...................... 190

D–7 Pareto curve for the minimum weight and cost design of the laminated plate ......................................................... 192

D–8 Limitations of the weighting method for constructing a concave Pareto front ................................................................. 193
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDOA</td>
<td>double-distribution optimization algorithm</td>
</tr>
<tr>
<td>EA</td>
<td>evolutionary algorithm</td>
</tr>
<tr>
<td>EDA</td>
<td>estimation of distribution algorithm</td>
</tr>
<tr>
<td>ES</td>
<td>evolution strategy</td>
</tr>
<tr>
<td>GA</td>
<td>genetic algorithm</td>
</tr>
<tr>
<td>KDE</td>
<td>kernel density estimate</td>
</tr>
<tr>
<td>PBIL</td>
<td>probability-based incremental learning</td>
</tr>
<tr>
<td>UMDA</td>
<td>univariate marginal distribution algorithm</td>
</tr>
</tbody>
</table>
LIST OF SYMBOLS

Optimization problem definition

\[ x = (x_1, x_2, \ldots, x_n) \] design variables

\[ n \] number of design variables, problem dimension

\[ F \] objective function, fitness function

Estimation of distribution algorithms

\[ p^*(x) \] selection probability

\[ p(x) \] probability distribution of selected points

\[ \lambda \] population size

\[ \mu \] number of selected points

\[ \nu \] candidate pool size

\[ \tau \] selection ratio for truncation selection

\[ p_m \] mutation probability

\[ \epsilon \] bound on marginal probabilities

\[ \sigma \] bandwidth of the kernel density estimate

\[ V = (V_1, V_2, \ldots, V_m) \] auxiliary variables

\[ m \] number of auxiliary variables
**Composite laminates**

\[ \theta = (\theta_1, \theta_2, \ldots, \theta_n) \]  

laminate ply angles with respect to a reference coordinate system

\[ [\pm \theta_1 / \pm \theta_2 / \ldots / \pm \theta_n]_s \]  

balanced symmetric stacking sequence: \( \pm \theta_k \) stands for the two-ply stack \( \theta_k / -\theta_k \). The ‘s’ subscript indicates that only the top half of the laminate is given: the lower half is obtained by symmetry with respect to the mid-plane.

\( V_1^*, V_3^*, W_1^*, W_3^* \)  

lamination parameters: contribution of the geometry to the laminate stiffness

**Evolutionary computation terminology**

- **fitness**  
  measure of the goodness of a candidate solution = objective function

- **population**  
  set of candidate solutions to the problem

- **generation**  
  \( \approx \) iteration

- **chromosome**  
  candidate solution

- **gene**  
  design variable (in this work)
Abstract of Dissertation Presented to the Graduate School of the University of Florida in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

OPTIMIZATION OF COMPOSITE STRUCTURES
BY ESTIMATION OF DISTRIBUTION ALGORITHMS

By
Laurent Grosset

December 2004

Chair: Raphael T. Haftka
Major Department: Department of Mechanical and Aerospace Engineering

The design of high performance composite laminates, such as those used in aerospace structures, leads to complex combinatorial optimization problems that cannot be addressed by conventional methods. These problems are typically solved by stochastic algorithms, such as evolutionary algorithms.

This dissertation proposes a new evolutionary algorithm for composite laminate optimization, named Double-Distribution Optimization Algorithm (DDOA). DDOA belongs to the family of estimation of distributions algorithms (EDA) that build a statistical model of promising regions of the design space based on sets of good points, and use it to guide the search. A generic framework for introducing statistical variable dependencies by making use of the physics of the problem is proposed. The algorithm uses two distributions simultaneously: the marginal distributions of the design variables, complemented by the distribution of auxiliary variables. The combination of
the two generates complex distributions at a low computational cost.

The dissertation demonstrates the efficiency of DDOA for several laminate optimization problems where the design variables are the fiber angles and the auxiliary variables are the lamination parameters. The results show that its reliability in finding the optima is greater than that of a simple EDA and of a standard genetic algorithm, and that its advantage increases with the problem dimension. A continuous version of the algorithm is presented and applied to a constrained quadratic problem. Finally, a modification of the algorithm incorporating probabilistic and directional search mechanisms is proposed. The algorithm exhibits a faster convergence to the optimum and opens the way for a unified framework for stochastic and directional optimization.
CHAPTER 1
INTRODUCTION

Today, composite materials are essential in the development of high performance structures. In particular, they have found usage in aerospace, automotive, marine, civil, and sport equipment applications where their high stiffness-to-weight or strength-to-weight, as well as their amenability to tailoring are greatly appreciated. However they are costly and careful design is critical in order to make a rational use of these materials: the freedom that they give to the designer comes at the price of a more complex design process because the boundary between materials and structures is blurred. As a result, off-the-shelf optimization methods (see Pardalos and Resende, 2002, for a survey of optimization methods) are either computationally too expensive or not applicable to many composite design problems. The challenge of optimizing composite structures has been the object of intensive research for many years, and has resulted in a number of effective tools (Venkataraman and Haftka, 1999).

One of the most successful developments of the last decade in the field of laminate optimization has been the introduction of genetic algorithms (GA), which are search methods inspired by the interpretation of natural evolution as an optimization process. One of the strengths of GAs is their ability to handle non-linear discrete problems, which often arise in composite laminate optimization. One may argue that another appeal of these methods is their ease of implementation and use. However, while the method itself is simple, effective application to practical cases often requires
expertise from the user. Typically, a number of parameters have to be adjusted by hand, based on the user’s knowledge of the problem and experience with the algorithm. In addition, GAs typically require a large number of function evaluations, which can be prohibitive when a single evaluation takes minutes or hours. More recently, several attempts have been made to reformulate genetic algorithms in a statistically more general form, giving rise to a family of optimization methods called estimation of distribution algorithms (EDA). The main benefit of abandoning the biological analogy is the increased control over the search it provides. Unlike GAs, in which promising designs are created by combining features from good individuals at random, EDAs attempt to explicitly identify the features that make these individuals good, and to use this knowledge to create new individuals that are likely to have a high fitness.

However there are only few examples of applications of EDA methods to engineering problems, and to our knowledge, these advances have yet not reached the field of laminate optimization. The main objective of the present work is to develop efficient estimation of distribution algorithms for composite laminate optimization. Our approach is dictated by one of the conclusions of the No Free Lunch theorem (Wolpert and Macready, 1995), which expresses the fact that if an algorithm is more efficient than some other algorithm on one class of problems, the price to pay is lower performance on other classes of problems. The authors formulate the task of developing a new algorithm as follows: “The only important question is, “How do I find good solutions for my given cost function $f$?” The proper answer to this question is to start with the given $f$, determine certain salient features of it, and then construct
a search algorithm, \( a \), specifically tailored to match those features.” Consequently, instead of creating a black box algorithm that works well for most problems, we use our knowledge of the particular class of problems under consideration to improve performance. We will show that by incorporating information about the physics of the problem via carefully chosen auxiliary variables, one can significantly improve the efficiency of the search.

1.1 Optimization of composite laminates

The mechanical properties of composite materials can be tailored to the problem at hand. This constitutes an advantage but it makes the design of the structure more complex because it involves not only the choice of the materials and of the geometry, but also the internal arrangement of the material.

Laminated composites are made up of a stack of plies composed of stiff fibers oriented along a given direction embedded in matrix, such as polymer resin. The macroscopic properties of the laminate depend on the type and the geometry of the fibers (carbon, glass, Kevlar, etc), on the matrix properties, on the fiber volume fraction, on the number of plies and their thickness, and on the orientation of the fibers. For unidirectional composites, the strength and the stiffness are considerably higher in the direction of the fibers than in the transverse direction. The purpose of optimization is to determine the number of plies, their thickness as well as their material and orientation so as to extremize certain criterion, for instance the displacement at one point, the weight, the cost, the first natural frequency, etc., subject to constraints on strength or manufacturing. This is known as stacking-sequence optimization.
In general, the variables involved in laminated composite design are mixed, i.e. defined over discrete and continuous spaces: some are discrete by definition (the material has to be chosen from a catalog), some become discrete due to manufacturing constraints. In particular, in many applications, the orientation of the plies is not arbitrary but can take only a limited set of discrete values, for example $0^\circ$, $\pm 45^\circ$, and $90^\circ$. Consequently many composite optimization problems are combinatorial problems, notorious for being difficult to solve by traditional (gradient-based) methods. In addition, these problems, when expressed in terms of the ply angles as design variables, have been shown by Pedersen (1987) to possess several local optima. To summarize, composite laminate optimization problems are in general multi-modal discrete optimization problems.

Over the years, extensive research has been devoted to solving these problems. The high computational cost of available discrete optimization algorithms led early researchers to relax the discreteness constraint in order to apply readily available methods to stacking-sequence optimization problems. In a design for minimum weight under buckling constraints, Schmit and Farshi (1977) forced discreteness of the orientations by using the total thickness of each of some prespecified angles as design variables. These were treated as continuous during the optimization and the final result was rounded to the closest multiple of the ply thickness available from the manufacturer. The method was not capable of optimizing the stacking sequence, furthermore, the rounding procedure did not guarantee optimality. The difficulty of dealing with discrete ply angles was addressed by Shin et al. (1990) who used a gradually increasing penalty approach which forced a continuous search algorithm to
converge to discrete values for the ply orientations. The main difficulty was to choose an appropriate adaptation strategy for the penalty parameter in order to prevent both premature convergence to a local optimum and convergence to non-discrete solutions.

The second undesirable characteristics of stacking-sequence optimization problems, namely the non-convexity was removed by Haftka and Walsh (1992) for the buckling load maximization problem. They transformed the non-linear optimization problem into a linear programming problem by formulating it in terms of “ply-orientation-identity” design variables. The resulting problem was then amenable to very efficient linear programming algorithms. However, the method is limited to stiffness design and cannot address strength constraints. More recently, Foldager et al. (1998) used the fact that many laminate optimization problems are convex when expressed in terms of higher-level mechanical quantities called lamination parameters. They performed a continuous optimization using ply-angles and lamination parameters simultaneously with the aim of forcing convexity. They used a sequential procedure, where ever improving starting points were chosen based on lamination parameter sensitivity information.

In the early 1990’s, stochastic search methods began to be applied to stacking-sequence optimization because they exhibit several desirable attributes: firstly, they do not require continuity or gradient information and can therefore be applied to integer or discrete problems. Secondly, they are inherently global and are consequently less likely to get trapped in local optima. Lombardi et al. (1992) solved the bucking load maximization by using simulated annealing. The algorithm demonstrated high reliability in finding near-optimal designs. Callahan and Weeks (1992) first applied
a genetic algorithm to solve a stacking-sequence problem by using the ply orientations as discrete design variables. Le Riche and Haftka (1993) developed a genetic algorithm that took advantage of the physics of the problem by devising genetic operators, such as the permutation operator, specifically targeted at modifying certain mechanical quantities, such as bending stiffness while leaving other quantities constant. They were able to improve efficiency because the number of unsuccessful mutations could be reduced. An alternate strategy was proposed by Autio (2000) to incorporate physics-based knowledge into a the optimization: first a continuous optimization is applied, with the lamination parameters as design variables, then a genetic algorithm is used to determine the stacking-sequence that yields the optimal lamination parameters. Other researchers have reported successful applications of stochastic optimization approaches to laminate optimization problems. For instance Zabinsky (1998) advocated the use of simple random search algorithms called Improving Hit-and-Run (IHR) in combination with “black box” numerical simulation tools because they provide both efficient and robust methods.

During the last decade, genetic algorithms have become a standard tool for stacking-sequence optimization (e.g. Punch et al., 1995; McMahon et al., 1998; Park et al., 2003). At the same time, the field of genetic optimization has undergone major mutations. Genetic algorithms, and more generally evolutionary algorithms have been thoroughly studied and improved. That research has given rise to a new type of genetic algorithms based on statistical methods. The next section explains how the transition from standard GAs to search distribution evolutionary algorithms can potentially be beneficial to stacking sequence optimization.
1.2 From genetic algorithms to estimation of distribution algorithms

Genetic algorithms (GA) (Holland, 1975; Goldberg, 1989) are inspired from the biological metaphor of the Darwinian theory of evolution, which states that a species is capable of adapting to a changing environment because those individuals of the population that are best adapted are more likely to have descendants and therefore to pass on their good traits to the next generation. The underlying assumptions are that natural selection produces individuals that are optimal with respect to their environment and that one can duplicate the process on an artificial optimization problem with a computer.

Genetic algorithms start by initializing a population of individuals at random. Each individual represents a possible configuration of the structure to optimize, coded into a string of finite length, called chromosome. Then “evolution” is simulated by successively applying selection of parents, recombination, mutation and replacement of the parents, creating a sequence of populations of increasingly “fit” individuals (individuals with high objective function evaluation). Recombination, or crossover, is an operator that swaps genetic material between two or more parent individuals to create children. Mutation introduces random perturbation in an individual, for instance by changing the value of one gene.

GAs are part of a more general class of stochastic algorithms called evolutionary algorithms, which also includes evolutionary programming (EP) (Fogel et al., 1966) and evolution strategies (ES) (Rechenberg, 1973). All three algorithms are based on the Darwinian evolution principle. Although initially these methods had strong
differences, nowadays, the frontiers have blurred, and they differ only in implemen-
tation details. Mostly, GAs typically use binary representation and use crossover as
the main operator and mutation as a background operator, EP uses representations
tailored to the problem and does not use crossover, and ES’s work mainly with con-
tinuous representation. While EP and ES use both crossover and mutation, they rely
heavily on mutation, which is usually adaptive. For a more detailed introduction to
evolutionary algorithms, the reader is referred to Spears et al. (1993) or Bäck (1996).
Although genetic algorithms usually designate evolutionary algorithms that use bi-
nary representation, we will use the expression in a broader sense to refer to discrete
evolutionary algorithms, even when they work with non-binary variables and make
ample use of mutation.

Since their inception, genetic algorithms have been the subject of intensive inves-
tigation to understand how they work and the types of problems on which they can
be expected to perform well (relative to other algorithms). Holland (1975) introduced
the concept of schemata, or partially defined strings to analyze the performance of
GAs. This led to the Building Block Hypothesis, which states that “GA works well
when instances of low-order, short schemas (‘building blocks’) that confer high fit-
ness can be recombined to form instances of schemas that confer even higher fitness”
(Mitchell et al., 1994). In other words, the efficiency increases when genes that work
together can be detected by partially sampling the design space and their association
has a high probability of being preserved during GA processing. A reflection on ways
to detect and preserve linkage between genes resulted in the emergence of a new class
of GAs. Initial work relied on reordering operator to group these genes, so that they
do not get parted by crossover,\(^1\) but Goldberg and Bridges (1990) showed that these operators were too slow to be effective. The messy genetic algorithm (mGA, Goldberg et al., 1989) was an attempt to “learn” the structure of the problem by using a flexible coding that enables the algorithm itself to evolve the adjacency relation between genes so as to keep strongly correlated variables together. By minimizing the defining length of these building blocks, the chance of crossover breaking it would be minimized. This line of research has led to the conclusion that the power of natural evolution was bounded and that linkage could only be efficiently learned by explicit statistical methods (Harik, 1999).

At the other end of the spectrum, the increasing complexity of algorithms provoked skepticism among certain researchers, such as Baluja, who claimed that one could dispense with the population (Baluja, 1994; Baluja and Caruana, 1995). Based on the hypothesis that the role of the population is to keep memory of good regions of the design space, he proposed to replace it with an explicit probability vector that summarizes information collected from all the points visited so far during the search. That vector is used to generate new individuals with high probability of having high fitness function evaluations. The new observations are in turn used to update the model by moving the probabilities toward the best individual. The resulting algorithm is called PBIL, for probability-based incremental learning. Extensive comparisons with a simple GA, which has a higher level of sophistication revealed

\(^{1}\) In standard crossover operators, genes that lie close to each other are less likely to be separated than genes that are situated far apart on the chromosome.
that PBIL outperforms GA on many standard test problems (see also Baluja (1993)).

Finally, one last endeavor contributed to the emergence of estimation of distribution algorithms: the attempt to study GA’s behavior by Markov chain analysis to obtain convergence time estimates (Nix and Vose, 1991). These methods see evolutionary algorithms as dynamical systems and try to determine their asymptotic behavior (stability, convergence time, etc). They involve modeling the population in terms of transition probabilities. Naturally, researchers of that field realized that behaviors similar to GA’s could be achieved by replacing populations by probability distributions. For example, Mühlenbein and Mahnig (2000) showed that the traditional crossover operator can be replaced by a probabilistic model that represents the distribution of good designs under certain conditions.

All three approaches converged toward the emergence of estimation of distribution algorithms. These algorithms use a probabilistic model of good individuals encountered during the optimization to guide the search toward promising regions of the design space. Most of the research conducted in the field of EDAs is devoted to the construction of good probabilistic models.

Mühlenbein and Mahnig (2000) found that without selection, the gene frequencies of a population converged to a state where genes become decorrelated from each other. Based on this conclusion, they created an algorithm that explicitly stores the univariate allele frequencies in the form of a probability vector. The algorithm, called univariate marginal distribution algorithm (UMDA) is very similar to the PBIL algorithm. Harik et al. (1999) developed a similar probabilistic algorithm named compact genetic algorithm (cGA), which mostly differs in its probability update rule.
All of these algorithms used simple probabilistic models that do not account for variable interactions. In other words, they assume that the variables have no combined action (independent variables) and monitor only univariate marginal frequencies. In many practical situations, however, variables do influence one another. For instance, in the knapsack problem, the size of the items already chosen affects the choice of the remaining items, or in stacking sequence design for maximum buckling load, some groups of plies can determine the buckling mode, and alter the most desirable angle for other plies. Therefore it is important to account for interaction between variables. A practical way of modeling variable interaction in the probabilistic sense is to use Bayesian networks. Bayesian networks are made up of a structure (directed acyclic graph), which indicates the variables that are interdependent, and numbers (conditional probabilities), which express the strength of the relationship. Although in theory it is possible to model any order of interaction between variables, the number of model parameters (the conditional probabilities associated with each link of the graph) to estimate from the data, and hence the size of the sample required to obtain an accurate graph increases very rapidly with the order of the interactions considered. In addition, having a high order model may be harmful, generating only duplicates of the population being modeled: this is especially detrimental when the regions already visited do not contain the optimum, because the model will prevent

---

2 Task of selecting items to be packed into a knapsack. The items are chosen from a large number of objects with different volumes. One can choose as many items of whatever volume to go into the knapsack as one likes, provided that the total volume of the knapsack is filled.
the exploration of new areas, as will be seen in Section 4.6.

The simplest type of interactions are pairwise interactions, which relate two variables (genes). Several approaches have been proposed to incorporate pairwise interactions into the probabilistic model, depending on what measure of coupling is used, and how the model is constructed. Even though in general an arbitrary joint distribution cannot be expressed in terms of pairwise conditional probabilities, the first order interaction model can be fitted to the distribution in the sense of some distance. De Bonet et al. (1997) used a chain model to represent the joint distribution of promising individuals. Their algorithm, called MIMIC (for mutual information maximizing input clustering) looks for the gene permutation and the conditional probabilities that minimize the Kullback-Lieber distance with respect to the true distribution of the selected individuals. They applied MIMIC algorithm to solve several deceptive problems (problems where univariate frequencies alone drive the optimization away from the optimum) and observed considerable gain over GA and PBIL. Baluja and Davies (1997) extended MIMIC algorithm by using dependency-tree instead of chain model, thus offering more flexibility in the representation of joint distributions. Even higher order interactions can be represented by using Bayesian Networks. Pelikan et al. (1999) developed an algorithm, called Bayesian Optimization Algorithm (BOA) that can model any order of interaction between variables. Experimental results on several test problems indicate that BOA benefits from learning second order interactions on functions involving strong variable interactions, while simple GAs or zeroth order probabilistic algorithms are more effective on problems that do not exhibit strong dependencies.
To date, there have been only few applications of EDAs to engineering problems. The only application that we found was a comparison of the relative performances of GA and PBIL for two combinatorial problems: a turbine balancing problem and a stock cutting problem (Carter, 1997). Carter found that GA performed significantly better than PBIL on both problems. One possible explanation for the better performance of GA on that problem may be that it effectively uses a swap operator, which exchanges two genes at random, which cannot be obtained with the PBIL algorithm.

1.3 Goals of this research

The present work aims at developing efficient strategies for stacking sequence optimization through the implementation and improvement of estimation of distribution algorithms. We start by implementing simple EDAs to solve laminate optimization problems, with only minor changes to the original UMDA algorithm. Then modifications are proposed to improve the performance. In particular, the main emphasis of this work is the incorporation of physics-based knowledge to the algorithm to improve the accuracy of the statistical model of promising regions, and hence the efficiency of the search. This dissertation is organized as follows:

- in Chapter 2, a general introduction to EDAs is provided. General principles are presented, along with the inherent difficulties associated with the estimation of distributions;
- Chapter 3 presents a simple EDA implementation and its application to composite laminate optimization. The algorithm’s performance is investigated, and improvements are proposed.
• in Chapter 4, physics-based knowledge is used to improve the performance of the algorithm introduced in Chapter 3. This is achieved by monitoring the probability distributions of good individuals both in the ply angle space and in the lamination parameter space to obtain complementary information about promising regions. The goal is to model variable interactions with two simple statistical distributions.

• Chapter 5 provides a summary of the dissertation and identifies future areas of research.

• Chapter 6 is a summary of this work in French, in compliance with the joint degree agreement between the University of Florida and the École des Mines de Saint-Étienne.
CHAPTER 2
ESTIMATION OF DISTRIBUTION ALGORITHMS

In this chapter, a general introduction to estimation of distribution algorithms (EDAs) is provided. Estimation of distribution algorithms search the design space by iteratively estimating the distribution of promising points and using it to guide the search in subsequent generations. One goal of this chapter is to approach EDAs from a statistical standpoint in order to identify difficulties and outline potential avenues to improve their efficiency. After introducing a conceptual EDA dealing with probability distributions only, the actual algorithm is obtained by estimating these distributions from samples of points.

2.1 Preliminary: stochastic model of a deterministic function

Let $F(x)$ be a deterministic function to be maximized over a domain $D$. The goal of global optimization is to find the set of maxima $O = \{x \in D : \forall y \in D, F(y) \leq F(x)\}$ (see Horst and Pardalos, 1995, for a survey of global optimization methods). When prior information about the form of $F$ is available (for instance if we know that $F$ is linear, or convex over $D$), then its value at some points of the design space can be extrapolated to other points of $D$, and information about the location of the optimum can be extracted. However, when no prior information about the form of the objective function is available, assumptions need to be made. A convenient way to express uncertainty about the true form of $F$ is to define a prior distribution
over a set of functions,\(^1\) and update that distribution on the basis of incoming observations, thereby reducing the uncertainty about \(F\). Even if \(F\) is deterministic, it will be represented as a probability distribution reflecting our lack of knowledge, as illustrated conceptually in Figure 2–1.

![Figure 2-1](image_url)

(a) Functions compatible with the observations.

(b) Contours of \(p(F|x)\). The combination of a prior probability about the form of \(F(x)\) and the observations yields a distribution over \(D\).

Figure 2–1: Representation of our uncertainty about the objective function as a probability distribution \(p(F|x)\).

This representation of uncertainties about the objective function is explicitly used in global optimization algorithms called stochastic optimization (SO) algorithms\(^2\) (see for example Betrò (1990); Schoen (1990), and Törn and Zilinskas (1987, Chap. 6) for a general introduction). Given a stochastic model of \(F\), SO define heuristics to choose the next point to be evaluated so as to extremize some criterion. Two common

\(\text{---}\)

\(^1\) Often, this prior distribution is only implicit, in the form of a bias toward simple, smooth functions.

\(^2\) Sometimes, these algorithms are referred to as Bayesian Algorithms, because they use Bayes’ rule to update probabilities when new observations are made.
criteria are:

1. maximize the probability that $F$ exceeds some value $F_0$;
2. maximize the expected value of $F$.

Estimation of distribution algorithms (EDAs) derive from this stochastic representation of the objective function. They focus on the set of “good points” $S$ (for example the set of points whose fitness exceeds $F_0$). The stochastic model $p(F|x)$, constructed from a set of observations $x_i \in S$, $i = 1, \ldots, N$, combined with the definition of the set of promising points $S$ determines the probability of $F$ being in $S$, denoted by $p(F \in S|x)$. The probability distribution $p(x|F \in S)$ is then obtained by Bayes’ rule:

$$p(x|F \in S) = \frac{p(F \in S|x)p(x)}{p(F \in S)}, \quad (2.1)$$

where $p(F \in S)$ is a normalizing constant, and the prior probability $p(x)$ reflects a potential bias toward certain regions of the space. When no region is preferred a priori, the probability density of promising regions $p(x|F \in S)$ is equal to the probability that the value of $F$ belongs to $S$.

Our uncertainty about the form of $F$ gives rise to the probabilistic model of promising regions, which dictates a probabilistic search strategy implemented in estimation of distribution algorithms. The idea of EDAs is to directly model the set of promising points $S$ (unlike SO, which models the function $F$) and sample from it to obtain more good points, and eventually find the optimum. The general principles of EDAs will be provided in the next sections.
2.2 Estimation of distribution algorithms

2.2.1 General principle

Let $F(x)$ be a (fitness) function to be maximized over a domain $D$. Let $O$ designate the set of optima $O = \{ x \in D : F(x) = F^\text{opt} \}$, where $F^\text{opt}$ is the maximum if $F$ over $D$. The principle of estimation of distribution algorithms is to construct a sequence of distributions $\{ p_t(x) \}_{t \in \mathbb{N}}$ that converges toward the uniform distribution over the set of global optima $O$.

The sequence of distributions is obtained as follows:

1. $p^0(x)$ is uniform over $D$
2. $p^{t+1}(x)$ is the distribution over $D$ obtained after application of a fitness-based selection operator to all the points of the domain. The selection operator is defined by its probability $p^s(x) \in [0, 1]$, which is the probability of accepting a point $x$ in an acceptance/rejection process. Using Bayes’ rule, the recurrence equation reads

$$ p^{t+1}(x) = \frac{p^t(x)p^s(x)}{\bar{p}^s} \quad (2.2) $$

where $\bar{p}^s = \int_D p^t(x)p^s(x)dx$ is a normalizing constant.

For the sequence to converge to the uniform distribution over $O$, two conditions are required:

1. $F(x_i) = F(x_j) \Rightarrow p^s(x_i) = p^s(x_j)$
   (the selection probabilities of two points having the same fitness function evaluation are the same),
2. let $D^t = \{ x \in D : p^t(x) > 0 \}$, let $R^t_F = \{ F(x) : x \in D^t \}$,
\[ \forall t > 0, \exists F^t_0 \in \mathcal{R}_F^t \text{ such that} \]
\[ F(x_i) \leq F^t_0 \quad \Rightarrow p^*(x_i) < p^*(x_j) \quad \text{for} \ (x_i, x_j) \in (\mathcal{D})^2 \]
\[ F(x_j) > F^t_0 \]
(\text{the probability of low-fitness points is smaller than that of high-fitness points})

It has been shown (Berny, 1999, for a discrete case) that the sequence of probability distributions \( p^t(x) \) generated by the successive applications of fitness-based selection to the whole space is identical to the sequence of distributions that would be obtained by performing a gradient-based maximization of the expectation of the fitness

\[ E_p[F] = \int_{\mathcal{P}} F(x)p(x)dx \]

with respect to the distribution \( p(x) \) in the space of the probabilities \( p \).

### 2.2.2 Illustration

The evolution of the distribution of promising points is illustrated for the simple 1D non-negative function:

\[ F(x) = 50 - 60x^2 + 50x^5 + 20x^{10} + 20 \sin(40x/\pi) . \quad (2.3) \]

The function has three local optima on \([0, 1]\), at \( x = 0.11, x = 0.61, \) and \( x = 1.00, \) as shown in Figure 2–2.

Given a fitness function and a selection operator, it is possible to explicitly determine the distribution of promising points at any iteration \( t \). For instance, when fitness proportional selection is used, the selection probability is given by

\[ p^*(x) \equiv p(S|x) = \alpha f(x) , \]
where \( \alpha \) can be chosen arbitrarily. The probability distribution evolves according to

\[
p^{t+1}(x) = p(x|S)
\]

\[
= \frac{p(S|x)p(x)}{\int_0^1 p(S|x)p(x)dx}
\]

\[
= \frac{f(x)p^t(x)}{\int_0^1 f(x)p^t(x)dx}.
\]

(2.4)

The distribution at times \( t = 0, t = 5, t = 10, \) and \( t = 20 \) is shown in Figure 2–3. Starting from a uniform distribution over \([0, 1]\), \( p^t(x) \) gradually focuses on high-fitness regions, and degenerates into a Dirac function centered about the (here unique) global optimum.

![Figure 2–2: The 1D objective function has three local maxima at \( x = 0.11, x = 0.61, \) and \( x = 1.00, \) but a single global maximum at \( x = 0.11. \)](image)

![Figure 2–3: Convergence of the distribution \( p(x) \) toward the global optimum \( x = 0.11. \)](image)

2.3 The general estimation of distribution algorithm

Estimation of distribution algorithms construct a sequence of probability distributions \( \{p^t(x)\}_{t \in \mathbb{N}} \) that converges to a uniform distribution over the set of global optima. However, explicitly calculating the distributions using formulas such as Eq. (2.2) is not reasonable, as it implies computing the fitness function everywhere, which
renders the whole optimization useless.

In practice, the distributions $p^{t+1}(x)$ are not calculated explicitly, but approximated from a finite sample of size $\mu$ (similar to the “parent population” in the context of evolutionary computation), obtained by fitness-based selection among a set of $\lambda$ points (the “child population” in EAs) generated from $p^t(x)$.

Since the selection probability distribution $p^s(x)$ is simulated from a finite sample, so that the second condition of Section 2.2.1 may not be satisfied, causing selection error, and the distribution $p^t(x)$ of promising points at time $t$ is estimated based on a finite number of points, which generates estimation error (this will be addressed in detail in the next section), the practical algorithm can only achieve an approximate distribution $\hat{p}^t(x)$, which is not identical to the theoretical distribution $p^t(x)$ in general. The actual trajectory $\{\hat{p}^t(x)\}_{t \in \mathbb{N}}$ in the space of probabilities will be different from the theoretical trajectory. Depending on the quality of the approximations employed, it may or may not converge to the optimum. The “hat” (as well as the subscript $t$) will often be dropped in the remainder of this work for simplicity.

The general flowchart of EDAs is presented in Figure 2–4. The algorithm is comprised of three main steps, besides the initialization, where the distribution $p^0(x)$ is chosen (when not prior knowledge about the location of the optima is available, a uniform distribution over $\mathcal{D}$ is used). The first step consists in generating a population of $\lambda$ points by sampling from $p^t(x)$. Then, fitness-based selection is applied to obtain $\mu$ “good” points. The new distribution $p^{t+1}(x)$ is finally estimated from the selected population. The values of the population size $\lambda$ and the selected population size $\mu$ determine the accuracy of the approximations $p^s(x)$ and $p^{t+1}(x)$. The procedure is
repeated until a stopping criterion (usually a fixed number of function evaluations) is met.

The performance of estimation of distribution algorithms is determined by two factors:

- the choice of the selection scheme (probability distribution $p^s(x)$), which determines the bias toward fitter points of the population;
- the choice of the probability distribution model, and method adopted to estimate the distribution of selected points.

The next sections are devoted to these two critical components of the algorithm.

2.4 Selection schemes

In the theoretical EDA presented in Section 2.2.1, the transition from the distribution $p^t(x)$ of promising points at time $t$ to the distribution at time $t + 1$ is
accomplished by applying selection to the whole search space $D$. In the actual algorithm, the goal of selection is to generate $\mu$ “good” points that will be used to estimate the distribution of promising points.

A selection scheme is characterized by $\lambda$, the number of points sampled from $p'(x)$, and the selection probability $p^*(x_i)$, $i = 1, \ldots, \lambda$. The value of $\lambda$ determines the departure from the theoretical selection probability (the first condition of Section 2.3 requiring that the selection probability of higher-fitness points be higher than that of lower-fitness points may not be satisfied): larger values of $\lambda$ will lower the selection error. When $\lambda$ is too small, high-fitness areas may be ignored, and the algorithm may fail to locate the global optima, as will be observed in Section 4.6.

Selection determines the bias toward fitter points of the population. Let $P = \{a_1, a_2, \ldots, a_\lambda\}$ designate the population of individuals sorted in decreasing order of their fitness ($F(a_{i+1}) \leq F(a_i)$). Many different selection schemes can be devised: fitness proportional selection, tournament selection ($k$ points are drawn from the population, and the point having the highest fitness evaluation is kept), Bolzmann selection ($p^*(x) \propto e^{\beta F(x)}$), etc. In this work, two schemes will be used:

**Truncation selection:** given an initial population of $\lambda$ individuals, the $\mu = [\tau \lambda]$, $\tau \in [0, 1]$, individuals with the highest fitness evaluation are chosen. The selection probability distribution (without replacement) is:

$$p^*(a_i) = \begin{cases} \frac{1}{\mu} & \text{if } 1 \leq i \leq \mu \\ 0 & \text{otherwise} \end{cases}$$
**Linear ranking selection:** the probability of selecting an element $a_i$ of the population is proportional to its rank $i$ in the population. The selection probability distribution (with replacement) is:

$$p_s(a_i) = \frac{2(\lambda - i + 1)}{\lambda(\lambda + 1)}$$

The choice of a particular selection scheme determines the “selective pressure” (also called selection intensity), or bias toward fitter points of the distribution: strong selective pressure means that only the very best points will appear in the selected population. Several measures of the selective pressure have been proposed: Goldberg and Deb (1991) introduced the takeover time, or number of generations until the best individual fills up the whole population under the sole action of selection (no mutation and no crossover), Mühlenbein (1998) borrowed the selection intensity from the field of breeding, Blickle and Thiele (1996) used the concept of loss of diversity $d$, which is defined as the proportion of individuals that disappear in the process of selection. The loss of diversity of various schemes was quantified by Wieczorek and Czech (2002): they obtained values of $d = 0.711$ for truncation selection of ratio $\tau = 0.3$, and $d \leq 0.432$ for linear ranking selection (the maximum loss of diversity is obtained for an infinite population; smaller populations achieve a lower selective pressure).

The optimal selection scheme is problem-dependent: it is the result of a trade-off between fast convergence to high-fitness regions, and the risk of premature convergence to local optima when finite populations are used. Indeed, selection is a noisy classification task: considering two competing candidate points $x_1$ and $x_2$, the goal
is to retain the one which is more informative about promising regions. Practically, given a partition\(^3\) of \(x = (x_I; x_{II})\), such that \(F(x) = F_I(x_I) + F_{II}(x_{II})\) (such partitions are sometimes called “schema”, “forma” (Radcliffe, 1992) or “hyperplane” in the EA literature), and one wants to identify good values of \(x_I\) based on the observed response \(F\), the decision will be influenced by the value of the partition \(x_{II}\). The observed fitness \(F(x^*_I)\) of the optimal value \(x^*_I\) of the partition \(x_I\) is in fact a random variable \(F(x^*_I) = F_I(x^*_I) + F_{II}(x_{II})\), where \(x_{II}\) can take arbitrary values. This gives rise to a noisy decision task, as shown in Figure 2–5: the graph schematically shows the distribution \(p(F|\text{good})\) of the observed fitness of the optimal value \(F(x^*_I)\), and the distribution \(p(F|\text{bad})\) of some non-optimal competing value.

![Graph showing distributions](image)

**Figure 2–5**: Selection is a noisy classification task, as the influence of other variables affects the fitness of a subregion.

Clearly, the decision to choose one candidate over the other will be dictated by

---

\(^3\) For instance, \(x_I = (x_1, x_2, x_3)\) and \(x_{II} = (x_4 \ldots, x_n)\).
2.5 Estimating the distribution of promising points: theoretical issues

2.5.1 The conservation law for generalization performance

The estimation of a distribution is a common task in machine learning, in particular in pattern classification, and a host of techniques have been developed to address it. A number of theoretical, as well as practical issues associated with this task are well-known, and discussed at length in the literature. One classical problem consists in choosing a good model to approximate a distribution from a sample. Suppose we want to estimate a distribution \( p(x), x \in D \), on the basis of \( \mu \) sample points \( x_i, i = 1, \ldots, \mu \) distributed according to \( p(x) \). A fundamental result is that, in the absence of any problem-dependent information, there is no single best approximate distribution \( \hat{p}(x) \) that describes the sample distribution. This results from the No Free Lunch (NFL) Theorems (also called Conservation Law for Generalization Performance) that prove that generalization from a finite sample of points cannot be accomplished unless additional information about the distribution is available.
(Wolpert, 1992; Schaffer, 1994). It will be important to be aware of these theoretical limitations when constructing approximations to the distribution of promising points (and of the implicit assumptions made to overcome the NFL).

### 2.5.2 Simple example

To illustrate the difficulty of estimating a distribution from a sample of points, let us consider the sample of \( \mu = 20 \) points given in Table 2–1. These points \( x_i \in [0, 1], i = 1, \ldots, \mu \) were generated according to an unknown distribution \( p(x) \), which we seek to estimate.

**Table 2–1: Sample points, \( \mu = 20 \)**

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| x_1 | x_2 | x_3 | x_4 | x_5 | x_6 | x_7 | x_8 | x_9 | x_{10} | x_{11} | x_{12} | x_{13} | x_{14} | x_{15} | x_{16} | x_{17} | x_{18} | x_{19} | x_{20} |
| 0.81 | 0.20 | 0.25 | 0.85 | 0.79 | 0.34 | 0.31 | 0.77 | 0.24 | 0.34 | 0.51 | 0.38 | 0.66 | 0.22 | 0.79 | 0.48 | 0.38 | 0.70 | 0.55 | 0.63 |

Figure 2–6 shows the sample points, and three different possible estimated distributions \( \hat{p}(x) \). Although the three distributions are very different from one another,
all three are equally valid descriptions of the sample:

- the distribution $p_1(x)$ has two main modes, and several local maxima;
- the distribution $p_2(x)$ has only two modes;
- the distribution $p_3(x)$ considers that all the points were generated from a unimodal distribution, and that the apparent local density maxima result from variability in the sample.

To discriminate between all the possible approximations to the true distribution $p(x)$, one has to make assumptions about its structure, or incorporate prior information about the target function.

### 2.5.3 The bias versus variance compromise

Let $p(x)$ be the target distribution to be estimated, $\hat{p}(x|D)$ the approximation to $p(x)$ obtained based on the data $D = \{x_1, \ldots, x_\mu\}$. Since all the sample points are instances of $p(x)$, there will be random variations in $D$, and $\hat{p}(x|D)$ will reflect these variations. To obtain a meaningful estimate of the error incurred when using a particular model, one needs to average over all possible samples. The usual measure of the error made by the approximation at point $x$ is the root-mean-square prediction error:

$$e_{\text{rms}} = \sqrt{E_D\left[ (\hat{p}(x|D) - p(x))^2 \right]} ,$$  \hspace{1cm} (2.5)

where $E_D$ designates the expected value with respect to sample variability.

This total error can be broken down into two components as follows:

$$e_{\text{rms}}^2 = E_D [\hat{p}(x|D) - p(x)]^2 + E_D \left[ (\hat{p}(x|D) - E_D[\hat{p}(x|D)])^2 \right] .$$  \hspace{1cm} (2.6)
The first component, the expected value of the difference between the approximation and the true value is called *bias*; the second component, which describes the magnitude of the variations about the approximation’s expected value is called *variance*. Both terms contribute to the total error. Bias is typically high when too simple a model is used to approximate the distribution, it is also called *modeling error*. Variance results from variability in the sample points. It is also referred to as *learning error* and can be reduced by increasing the sample size. The difficulty of estimating a distribution from sample points lies in the trade-off between bias and variance, as sophisticated models will be more flexible, and have a low bias, but be extremely sensitive to sample variability. In contrast, the variance achieved by a simple model will be lower, but at the cost of a larger bias.

### 2.5.4 Assessing the accuracy of an estimate

Even though theoretically there exists no absolute criterion for deciding which of two approximations best describes a set of sample points when no a priori information about the target distribution $p(x)$ is available (NFL theorem), in practice, heuristic methods have been proposed to choose between competing models, and to estimate the model parameters. Those methods implicitly make regularity assumptions about the distribution (Schaffer, 1993): in fact, the idea of finding an optimum over a domain without exhaustive enumeration rests on the assumption that the fitness function possesses some regularity attributes that allow us to extrapolate from previous observations, hence we restricts our attention to particular classes of functions, as pointed out by Rao et al. (1995).
Given a set $\mathcal{D} = \{x_1, x_2, \ldots, x_\mu\}$ of $\mu$ independent identically distributed data points and a set of competing models $\mathcal{M} = \{m_1, m_2, \ldots, m_k\}$, three classes of methods can be used to choose the model $\hat{m}$ that best approximates the data (Duda et al., 2000, Chapter 9):

**Maximum likelihood (ML) methods** choose the model which maximizes the likelihood of the data:

$$\hat{m} = \max_{i=1,\ldots,k} p(\mathcal{D} | m_i)$$

$$= \max_{i=1,\ldots,k} \prod_{j=1}^\mu p(x_j | m_i) .$$

A major disadvantage of these methods is their tendency to overfit the data. A variant of ML methods is the maximum a posteriori (MAP) method, where some models can be favored over others by assigning them a higher a priori probability $p(m_i)$. The a posteriori probability of a model after the data has been observed is obtained by

$$p(m_i | \mathcal{D}) = \frac{p(\mathcal{D} | m_i)p(m_i)}{p(\mathcal{D})} ,$$

where $p(\mathcal{D})$ is a normalizing factor that is usually ignored. The model that maximizes the a posteriori probability is retained.

**Minimum description length methods (MLD)** recast the bias-variance compromise into a more general problem of finding the most compact representation of a data set. The information theory concept of Kolmogorov complexity $K$ (the minimum number of bits required to represent data on a computer) is used to measure the total complexity of the model $m_i$ and the description of the data
set in $m_i$:

$$K(m_i, \mathcal{D}) = K(m_i) + K(\mathcal{D} \text{ using } m_i).$$

The model that achieves the best trade-off (minimum $K(m_i, \mathcal{D})$) between the model complexity $K(m_i)$ and the model’s ability to efficiently represent the data $K(\mathcal{D} \text{ using } m_i)$ is retained.

**Cross-validation methods (CV)** are classical heuristic strategies used in statistics to estimate the generalization accuracy of a model. They consist in splitting the data $\mathcal{D}$ into $k$ sets $\mathcal{D}_i$ of equal size $\mu/k$. Then $k$ approximations are constructed by leaving out one set $\mathcal{D}_i$ and estimating $p(x)$ based on $k - 1$ remaining sets. Each time, a different error $\epsilon_i$ is obtained, and the generalization error is estimated as the mean of all the $\epsilon_i$’s. A common variant of cross-validation methods is the leave-one-out method, which is obtained for $k = \mu$.

In this work, the maximum likelihood method and the cross-validation method will be used (ML is invoked implicitly in Section 3 to estimate the marginal probabilities, and CV is used in Section 4 to choose the window size of the kernel density estimate).

### 2.6 Estimating the distribution of promising points in practice

If there is no a priori criterion for choosing the best model, how do we go about constructing an approximate distribution $\hat{p}(x)$? Based on the above considerations, our approach will be dictated by two general rules:

**Simpler is better (Occam’s razor):** Even though there is no problem-independent
theoretical reason for preferring simpler models over sophisticated ones, a principle often invoked in machine learning is Occam’s razor (overfitting avoidance), which recommends that, for a given total error, models involving fewer parameters be favored. Friedman (1997) showed for classification tasks that the bias error of simple models can be compensated by low variance error, thereby making them more accurate than more sophisticated models (EDAs can be viewed as classifiers that discriminate between good and bad regions of the search space).

**Incorporate as much information about the problem as possible:** Another way of reducing error consists in incorporating as much information about the target distribution as possible.

### 2.7 Estimation of distribution algorithms and other stochastic algorithms

Estimation of distribution algorithms can be viewed as a generalization of evolutionary algorithms (Bäck, 1996), and as such, they bear many similarities with genetic algorithms (Goldberg, 1989) and evolution strategies (Rechenberg, 1973). They can also be likened to a more recent family of stochastic optimization algorithms, called Markov chain Monte Carlo (Gelfand and Smith, 1990).

**Evolution strategies (ES)** search for the optimum by (1) selecting \( \mu \) good points (the parents) out of a pool of \( \lambda \) candidate points; (2) creating new points (the children) by applying perturbations (usually Gaussian mutation) to the selected points. If the population of selected points is considered as a whole, the process of choosing a parent and applying a Gaussian mutation is equivalent to sampling from the distribution of parents, estimated by a varying window size kernel.
method (see Appendix B). The specificity of ES lies in the strategy implemented to select the local bandwidth (i.e. the mutation standard deviation) by an auto-adaptive scheme, i.e. the goodness of a particular value of the bandwidth is judged based on the the fitness of the points that are generated using that value.

**Genetic algorithms (GA)** proceed by (1) selecting $\mu$ good points (the parents) out of a pool of $\lambda$ candidate points; (2) creating new points (the children) mainly by applying recombination operators (crossover) to the selected points. Mutation is applied to a small portion of the children. Viewed from a statistical standpoint, the specific crossover operator implemented defines an implicit statistical model for the estimation of $p(x)$ by determining the marginal frequencies considered for the creation of new points. For example, using a uniform crossover to create children amounts to sampling from univariate marginal frequencies of the parents.

**Markov Chain Monte Carlo (MCMC)** algorithms simulate sampling from a probability distribution $p(x)$ by constructing a Markov chain $(x_t), t \in \mathbb{N}$ whose invariant distribution is $p(x)$. MCMC use a two-step procedure to generate new points: a set of candidate points is created based on the current state $x_t$ and a proposal distribution $q(x)$; then the new points are accepted or rejected according to an acceptance probability $A(x_{\text{new}}|x_{\text{current}})$. The parallel between conventional EDAs and MCMC has been drawn (giving rise to hybrid algorithms called eMCMC, for evolutionary MCMC (Drugan and Thierens, 2003)).
EDAs can be viewed as non-homogeneous MCMCs whose states are populations of $\mu$ points, with proposal distribution at time $t$ equal to the distribution of promising points $p'(x)$, and acceptance probability commonly set to 1, even though other replacement strategies are possible. The idea is that after a sufficient number of iterations, MCMC will sample uniformly from the set of global optima.
CHAPTER 3
THE UNIVARIATE MARGINAL DISTRIBUTION ALGORITHM

In this chapter, a simple estimation of distribution algorithm, called Univariate Marginal Distribution Algorithm, is presented. It is based on the assumption that the design variables are statistically independent. The basic algorithm is tested on laminate optimization problems, and the influence of several parameters is investigated. The distribution-based search mechanism is compared to a point-perturbation-based search mechanism. Then improvements to the algorithm are proposed, and the resulting algorithm is applied to more realistic laminate optimization problems.

3.1 Algorithm

Let $F(x)$ be a function over a domain $D$. Our objective is to find the optima $x^*$. As introduced in Chapter 2, estimation of distribution algorithms search the design space by estimating the distribution of promising regions. The distribution of selected points at a given time of the optimization depends on the form of the objective function, and on the selection procedure. In particular, it has been shown (Mühlenbein et al., 1999) that when the objective function is a sum of contributions from single variables,\(^1\) and Bolzmann selection is used ($p^*(x) \propto \exp(\beta F(x))$, where $\beta$ is a parameter that determines the selective pressure) the distribution of selected

---

\(^1\) Such functions are commonly referred to as additively decomposable functions (ADF).
points takes the simple form

\[ p(x_1, x_2, \ldots, x_n) = \prod_{k=1}^{n} p(x_k), \]  

(3.1)

where \( p(x_k) \) is the univariate marginal distribution of variable \( x_k \). In other words, the variables are statistically independent, and the complete distribution can be obtained from the univariate marginal distributions only. This greatly simplifies the probability estimation process. The algorithm resulting from the variable independence assumption, called Univariate Marginal Distribution Algorithm (Mühlenbein and Mahnig, 2000), is presented in Figure 3–1. The algorithm is identical to the general EDA, except that the general form of \( p(x) \) is replaced by the univariate model, thus considerably decreasing the number of parameters to be estimated, which simplifies the algorithm, reduces the computational overhead required to estimate the distribution, and decreases the uncertainty on the parameters due to variability in the selected points. In this first variant of UMDA, no random exploration component such as mutation is implemented, because our goal is to observed the pure statistical search mechanisms. The potential benefits of such operators will be investigated in Section 3.3.

In this work, we deal with discrete variables \( x_k \) (the fiber orientation of each ply, which can take \( c \) values \( a_i, i = 1, \ldots, c \)), hence the probability functions \( p(x_k) \) are discrete probability functions. To specify \( p(x_k) \), we need to know the probability of each value \( a_i \), \( p_{ki} = p(x_k = a_i) \). Since the distribution of each variable in the selected points follows a multinomial distribution of probability \( p(x_k) \), the maximum likelihood values of \( p_{ki} \) are simply given by the frequencies \( f_{ki} \) of each value \( a_i \) in the
3.2 Study of the original UMDA

UMDA, as any EDA, theoretically works with probability distributions, in practice the distributions are estimated from finite samples of $\mu$ selected points. The value of $\mu$ determines the accuracy of the estimated $p_{ki}$: the larger the population size, the lower the standard error on these estimated model parameters. The choice of

By incorporating information about the form of the distribution, one can obtain a more accurate estimate of $p(x)$. However, there is a price to pay for the simplification achieved: if the variable independence assumption is not satisfied, the estimated distribution will be erroneous, and the resulting algorithm may fail to locate the optima. In the following sections, we shall apply UMDA to ADF, and study their behavior on more complex functions.
the population size will therefore will be driven by two considerations: the accuracy
of \( p(x_k) \) and the computational cost one is willing to pay for it. The other factor
affecting UMDA’s behavior is the selective pressure. The present section investigates
the influence of these two factors on the algorithm’s performance.

3.2.1 Problem description

We first studied the convergence properties of UMDA for a problem that satisfies
the variable independence assumption. The goal is to understand the strengths and
limitations of UMDA (and more generally of EDA) in an ideal setting.

We considered the problem of maximizing the longitudinal in-plane stiffness \( A_{11} \)
of a balanced symmetric graphite-epoxy laminate\(^2\) \( [\pm x_1 / \pm x_2 / \ldots / \pm x_n]_s \):

\[
\text{maximize } A_{11} = h \left[ U_1 + \sum_{k=1}^{n} (U_2 \cos 2x_k + U_3 \cos 4x_k) \right], \quad (3.2)
\]

where the total laminate thickness was \( h = 0.2 \) in, and \( U_1, U_2, \) and \( U_3 \) are mate-
rial invariants (see Appendix A for more details about the mechanics of composite
laminates). The material properties of graphite-epoxy are given in Table 3–1. The
design variables are the fiber angles \( x_k, k = 1, \ldots, n \), where \( x_k \) designates the angle
of the \( k^{\text{th}} \) ply in the laminate with respect to a reference coordinate system. For
this problem, the fiber angles were to be chosen from \( \{0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ\} \),
and the number of \( \pm \theta \)-stacks in a half-laminate was set to \( n = 10 \) (40 plies in total).

\(^2\) A laminate is symmetric if the stacking sequence is symmetric with respect to
the mid-plane, and balanced if, for each \( \theta \)-degree ply, there is a \(-\theta\)-degree ply in the
laminate.
The objective function had only one maximum, at $[0_{20}]_s$ (all the fibers are aligned with the longitudinal axis). The longitudinal in-plane stiffness of that laminate was $A_{11} = 4.38 \times 10^6$ lb/in. It is apparent from Equation (3.2) that the objective function belongs to the class of ADFs, hence it makes sense to use the univariate model given in Equation (3.1) to approximate the distribution of $A_{11}$-based selected points.

![Fitness landscape](image)

**Figure 3–2:** Fitness landscape $A_{11}$ for $n = 2$ ($x_1$ and $x_2$ are in degrees).

### 3.2.2 Population size and selection pressure

In the basic form of UMDA, only two factors govern the algorithm’s behavior: the selection scheme, which determines $p_s(x)$ (the selection scheme encompasses the
selection method and the selective pressure) and the number of points \( \mu \) used to estimate the distribution of promising points \( p^f(x) \). The purpose of this section is to investigate their respective influence on the algorithm’s performance. UMDA was applied to the \( A_{11} \) maximization problem. Five different selected population sizes \( \mu \in \{10, 20, 50, 100, 200\} \) were tried, and truncation selection of ratios \( \tau \in \{0.1, 0.3, 0.5\} \) (cf. Section 2.4) were used\(^3\).

As any global optimization algorithm, an ideal estimation of distribution algorithm should possess two (often conflicting) attributes, as emphasized by Bäck (1996, Ch. 4) in the context of evolutionary algorithms: high convergence velocity and high convergence reliability. In this study, the optimization reliability \( R \) was estimated as the proportion of runs that find the optimum in a fixed number of evaluations. The mean best fitness at each iteration was used as a measure of the convergence velocity. Both criteria were estimated based on 50 independent runs, so that the standard error of \( R \) was \( \sigma_e = \sqrt{\frac{R(1-R)}{\sqrt{50}}} \) (\( R \) follows a binomial distribution), which is maximum at 0.07 for \( R = 0.5 \).

Figure 3–3 shows the evolution of the mean best fitness during the optimization for the 15 combinations of population size and selection ratio (the mean best fitness at the fifth generation is given in Table 3–2 to help the interpretation). The graph clearly

\(^3\) This choice of \( \mu \) as an independent parameter (\( \lambda \) as dependent parameter, determined by the value of \( \tau \)) is unusual: in evolutionary computation, the usual approach is to study the influence of the population size \( \lambda \). The rationale for this choice is to distinguish factors that influence the nominal value of the search distribution (the bias) and factors that affect the accuracy of the estimate (the variance).
shows three families of curves, corresponding to the selection ratio $\tau$: the algorithm converges faster to high function evaluation regions for strong selection. This is not surprising, as all the variables have the same weight in the fitness function, therefore high-fitness points are always more informative about good regions than lower-fitness points. Within a family of curves (identical selection ratio), the convergence velocity increases monotonically with the sample size $\mu$, as large samples yield more accurate estimates of the distribution of good points $p(x)$.

![Graph showing three families of curves](image)

Figure 3–3: Evolution of the mean best fitness as a function of the number of iterations. The convergence velocity increases when selection becomes stronger.

Table 3–2: Mean best fitness at the fifth generation

<table>
<thead>
<tr>
<th>$\mu$</th>
<th>$\tau$</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td>4.33</td>
<td>4.14</td>
<td>4.05</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>4.36</td>
<td>4.28</td>
<td>4.07</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td>4.38</td>
<td>4.33</td>
<td>4.19</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>4.38</td>
<td>4.36</td>
<td>4.24</td>
</tr>
<tr>
<td>200</td>
<td>N/A</td>
<td>4.37</td>
<td>4.28</td>
<td></td>
</tr>
</tbody>
</table>
The optimization reliability depends both on the selection ratio $\tau$ and on the selected population size $\mu$, as shown in Figure 3–4. For a given value of $\tau$, the accuracy of the distribution $p(x)$ is directly related to the sample size $\mu$: the larger the value of $\mu$, the smaller the variance of the marginal frequencies $f_{ki}$ of the selected points. This effect is even more significant for large values of $\tau$ (weak selection). When weak selection is used, the expected value of the proportion of points that contain the optimal value $x_k^*$ is barely higher than that of non optimal values. To guarantee that the estimated value of the proportion is higher, one has to reduce the variance, hence increase the selected population size.

![Figure 3–4: Reliability of the optimization for various combinations of populations sizes $\mu$ and $\lambda$, as a function of the number of iterations. The final reliability increases with the selected population size $\mu$.](image)

While the evolution of the reliability as a function of the number of iterations allows us to understand how the algorithm parameters affect the accuracy of the distribution $p(x)$, hence the search mechanism, the number of function evaluations
is a more meaningful measure of the cost associated with an optimization. Figure 3-5 shows the reliability of the 15 algorithm variants against the number of analyses. Using that new scale, a new hierarchy between the tested schemes appears: for small

![Figure 3-5: Reliability of the optimization for various combinations of populations sizes \( \mu \) and \( \lambda \), as a function of the number of function evaluations.](image)

selected population sizes (\( \mu = 10 \) and \( \mu = 20 \)), the best reliability is obtained for strong selection\(^4\), while for large selected populations (\( \mu = 100 \) and \( \mu = 200 \)), the reverse order is observed. This can be explained by the fact that the variance of the proportion of good points in the selected population is large for small numbers of

\(^4\) Note that the classical result that high selection rates require large populations (Thierens and Goldberg, 1993) consider \( \lambda \) as the independent parameter, consequently increasing the selective pressure results in larger sampling errors in the selected points and in a poorer performance. When the number of selected points \( \mu \) is the independent parameter, the selection pressure can be varied without affecting the sampling error, explaining the different conclusions presented here.
selected points, consequently a selective scheme has to be used to help discriminate. In contrast, when large numbers of selected points are used, the variance is sufficiently small to allow a good discrimination, so that the number of evaluations ($\lambda$) at each iteration determines the efficiency. This result shows the two ends of the spectrum in EDAs: small populations lead to statistical errors, whereas algorithms using large populations suffer from high computational costs.

This can be easily understood by examining the evolution of the probabilities with time. Figure 3-6 shows the evolution of the probability distribution $p(\theta_1)$, starting from a uniform distribution, during the first ten iterations. The combined effects of selected population size $\mu$ and selection ratio $\tau$ are clearly visible on the graphs: for the same selection intensity, the distribution converges more smoothly toward the optimum value $\theta_1 = 0$ (Figures (a) and (b)) with increasing $\mu$. With a small selected population of 10 points, the distribution loses variable values: for instance variable values 60 degrees and 90 degrees vanish between the first two iterations. In this case, the loss of values speeds up the convergence by favoring optimal values, however, premature convergence of the probabilities can lead to the loss of optimal values, which prevents the algorithm from finding the optimum, since no diversity injection mechanism (such as mutation or a lower bound on the marginal frequencies) is provided. A larger selected population enables UMDA to obtain more accurate estimates of all the probabilities, resulting in a more reliable optimization scheme. For a given

\footnote{Since all variables are interchangeable, the evolution of other variables would exhibit similar trends.}
Figure 3–6: Evolution of the probabilities for the $A_{11}$ max problem.
selected population size $\mu = 100$, Figures (b) and (c) show the faster convergence of the distribution when stronger selection is used: $p(\theta_1)$ converges in only 6 iterations for $\tau = 0.1$, while it has not fully converged after 10 iterations for $\tau = 0.5$.

To summarize, the selection ratio determines the convergence speed of the algorithm from one generation to the next. For identical selection ratios, large selected populations provide more accurate statistics about good alleles, thereby preventing premature convergence. However, if a large selection pressure associated with large selected populations constitutes the best configuration for fast convergence in terms of iterations, the increased number of analyses required to refine the evaluation of probabilities may result in an overall scheme which is less efficient than algorithms based on smaller populations. This problem comes down to the issue of the allocation of resources between exploration and exploitation: using a smaller selected population results in inaccurate parameter estimates, and can potentially drive the search toward poor individuals, but since fewer analyses are performed at each generation, the loss in accuracy is compensated by the increased number of iterations permitted, given a fixed budget.

### 3.2.3 Dimensionality

An important characteristic of optimization algorithm is the impact of an increase of the number of variables on the time to convergence, which is called the algorithm time complexity. In this section, we compare UMDA’s time complexity to that of a random search algorithm called Stochastic Hill-Climber (SHC). SHC is one of the simplest stochastic algorithms, and has been proposed as a baseline method.
for evaluating evolutionary algorithms’ performance (Juels and Wattenberg, 1995).

The algorithms are applied to two problems: the maximization of the in-plane longitudinal stiffness introduced in Section 3.2.1, and the maximization of the first natural frequency of a simply supported rectangular graphite-epoxy laminated plate.

**Problem description: frequency maximization**

The first natural frequency of a simply supported rectangular plate is proportional to the square root of the following expression:

\[ f_1 = \frac{D_{11}}{L^4} + \frac{2(D_{12} + 2D_{66})}{L^2W^2} + \frac{D_{22}}{W^4} \]  

(3.3)

where \( L = 20 \) in and \( W = 15 \) in are the length and width of the plate, the \( D_{ij} \) terms are the coefficients of the bending stiffness matrix (cf. Appendix A):

\[
D_{11} = U_1 \frac{h^3}{12} + \frac{4}{3} U_2 \sum_{k=1}^{n} t_k (3z_k^2 - t_k^2) \cos 2x_k + \frac{4}{3} U_3 \sum_{k=1}^{n} t_k (3z_k^2 - t_k^2) \cos 4x_k
\]  

(3.4)

\[
D_{22} = U_1 \frac{h^3}{12} - \frac{4}{3} U_2 \sum_{k=1}^{n} t_k (3z_k^2 - t_k^2) \cos 2x_k + \frac{4}{3} U_3 \sum_{k=1}^{n} t_k (3z_k^2 - t_k^2) \cos 4x_k
\]  

(3.5)

\[
D_{66} = U_5 \frac{h^3}{12} - \frac{4}{3} U_3 \sum_{k=1}^{n} t_k (3z_k^2 - t_k^2) \cos 4x_k
\]  

(3.6)

\[
D_{12} = U_4 \frac{h^3}{12} - \frac{4}{3} U_3 \sum_{k=1}^{n} t_k (3z_k^2 - t_k^2) \cos 4x_k
\]  

(3.7)

where the \( U \) terms are the material constants given in Table 3–1, the ply thickness \( t_k \) was fixed at 0.005 in, and \( z_k \) refers to the position of the \( k^{th} \) ply in the laminate, as illustrated in Figure 3–7.

Like the \( A_{11} \) maximization problem, this problem is an ADF, therefore UMDA theoretically converges to the optimum \( x_i^* = 60^\circ, i = 1, n \). However, contrary to the
first problem, the z’s in the $D_{ij}$’s give a hierarchical structure to the problem because plies located in outer layers have more weight than those located in inner layers, as can be seen in Figure 3–8 (case $n = 2$). As a result, the distributions corresponding to the outermost plies are expected to converge faster than those corresponding to inner plies. The functions $A_{11}$ and $f_1$ are unimodal, so that SHC will also yield $x^*$ for the two problems.

Figure 3–8: Fitness landscape of the first natural frequency $f_1$ for $n = 2$. The variable $x_1$ is the orientation of the outer plies, and $x_2$ is the orientation of the core plies.
A competitor for UMDA: stochastic hill-climber

SHC searches the space by choosing an initial point \( \mathbf{x} = (x_1, x_2, \ldots, x_n) \) at random and applying random perturbations to it: the algorithm changes the value of one variable chosen at random to an adjacent value and accepts the new point only if it improves the fitness function. The procedure is stopped when a fixed number of function evaluations has been performed.

Theoretical analysis of SHC

The following analysis considers a stochastic hill climber (SHC) operating on a unimodal function. If the SHC is at a point where \( k \) out of the \( n \) variables are correctly set, the expected time before one of the non-optimal variable is perturbed is \( n/(n-k) \). The random perturbation can then take the variable closer to the optimum or not, with probabilities \( 1/2 \) (neglecting distortions due to limits on the values). The expected time for one beneficial step is then \( 2n/(n-k) \). Let \( d_i \) denote the average distance between the \( i \)-th variables of a random point and the optimum,

\[
d_i = \frac{1}{c} \sum_{j=1}^{c} |x_i^j - x_i^*| ,
\]

(3.8)

where \( x_i^j \) is the \( j \)-th possible value of the \( i \)-th variable. In the cases presented here, the value of all the variables at the optimum is the same (\( x_i^* = 0^\circ \) for max \( A_{11} \) and \( x_i^* = 45^\circ \) for the vibration problem), therefore the average distance to the optimum is the same for all variables \( i \), \( d_i = d \) (but it varies with the problem). For each variable that is not correctly set, an average of \( d \) steps in the right direction is needed to reach the optimum. By summing up the expected times of each beneficial step, one obtains the expected time to locate the optimum from a random starting point that has \( k \)
optimal variables

\[ T_k = \sum_{i=k}^{n-1} \frac{2dn}{n-i} \cdot \tag{3.9} \]

\( T_k \) can now be averaged over all random starting points, which yields the expected convergence time of an SHC on a unimodal function,

\[ T_{\text{SHC}} = \frac{1}{2n} \sum_{k=0}^{n} C_n^k T_k = \frac{nd}{2^{n-1}} \sum_{k=0}^{n} C_n^k \sum_{i=k}^{n-1} \frac{1}{n-i} \cdot \tag{3.10} \]

It can be shown Garnier et al. (1999) that Equation (3.10) yields a convergence time of order \( \mathcal{O}(n \ln n) \) for large \( n \). Estimated (from Eq. (3.10)) and measured convergence times (average over 50 independent runs) are compared for both the max \( A_{11} \) and the vibration problems in Figure 3–9. The qualitative prediction of the dependency of the number of function evaluations in terms of the dimension \( n \) is correct. Because \( d \) is smaller in the vibration problem than in the max \( A_{11} \) problem, convergence is faster in the former case, which is also correctly predicted.

![Figure 3–9: Expected convergence time of the stochastic hill-climber for the max \( A_{11} \) and vibration problems. SHC finds the optimum in \( \mathcal{O}(n \ln n) \) evaluations.](image)
Theoretical analysis of UMDA

A univariate marginal distribution algorithm is now considered. In Mühlenbein et al. (1999), the behavior of a UMDA with truncation selection was studied for two test functions, *Onemax* and *Int*, that have common features with the max $A_{11}$ and the vibration problems. In the *Onemax* problem, the number of one’s in a binary string is maximized. As in max $A_{11}$, the function is separable, and each variable has the same contribution to the objective function. The *Int* function,

$$ Int = \sum_{i=1}^{n} 2^{i-1}x_i, $$

(3.11)
is also maximized on binary strings. As in the vibration problem, the function is separable and there is a gradual influence of the variables on the function. In the vibration problem however, the difference in sensitivity of the objective function to each variable is lower than in *Int*. If the population size, $\mu$, is larger than a critical value $\mu^*$, the authors show that the expected number of generations to convergence in distribution, denoted by $N_g$, and defined by $p^{N_g}(x^*) = 1$, is

$$ N_g \approx \mathcal{O}(\sqrt{n}) \quad \text{for Onemax} $$

(3.12)

$$ N_g \approx \mathcal{O}(n) \quad \text{for Int} . $$

(3.13)

The larger number of generations seen on *Int* is due to the different variable weights in the objective function. For a truncation rate $\tau = 0.5$, the selection in the first generation is exclusively based on $x_n$, in the second generation, it is based on $x_{n-1}$, etc. The discovery of the optimum is sequential in variable values, whereas some level of parallelism can be achieved on less hierarchical objective functions.
The expected number of objective function evaluations to convergence is

\[ N_f = N_g \mu^* . \]  

(3.14)

No analytical expression for \( m^* \) was given in Mühlenbein et al. (1999). An approximation to \( \mu^* \), \( \hat{\mu}^* \) is now proposed based on the initial random population sampling, and neglecting variable values lost during selection. The probability that a given variable value is not represented in the population is \( ((c - 1)/c)^\mu \). The probability that the values making up the optimum, \( x^* \), have at least a sample in the initial population is

\[ P_{\text{pop}} = \left( 1 - \left( \frac{c - 1}{c} \right)^\mu \right)^n . \]  

(3.15)

For a given \( P_{\text{pop}} \) (typically close to 1), the critical population size is estimated from Equation (3.15),

\[ \hat{\mu}^* = \frac{\ln(n/(1 - P_{\text{pop}}))}{\ln(c/(c - 1))} \approx \mathcal{O}(\ln(n)) . \]  

(3.16)

From Equations (3.12) to (3.16), the order of magnitude of the number of evaluations to convergence is

\[ N_f \approx \mathcal{O}(\sqrt{n} \ln(n)) \quad \text{for Onemax and} \]  

(3.17)

\[ N_f \approx \mathcal{O}(n \ln(n)) \quad \text{for Int} . \]  

(3.18)

Experimental results

The algorithms were applied to the problems for five different numbers of variables \( n = 12, 20, 50, 100, \) and 200. The selection ratio \( \tau \) of the UMDA was kept constant at \( \tau = 0.3 \) (as recommended in Mühlenbein and Mahnig (1999)). Several
population sizes $\lambda$ (50, 100, 500 and 1000) were tried in order to obtain an efficient scheme and allow a fair comparison with SHC.

Two criteria were used to compare the performance of the algorithms: the number of analyses required to reach 80% reliability (defined as the probability of finding the optimum, estimated over 50 independent runs), and the number of analyses needed until the average best fitness reaches 98% of the optimal fitness.

Figure 3–10 presents the number of evaluations to 98% of the maximum fitness against the number of variables for SHC and four different population sizes of UMDA. Clearly, SHC converges faster than UMDA for all the numbers of variables investigated. The evolution of the cost of SHC is close to linear, which confirms the results obtained in the previous section and reported in Pelikan et al. (2000) for the Onemax problem. For UMDA with a given population size, the number $N_{98}$ of evaluations needed to reach 98% of the optimal fitness increases sub-linearly: fitting a model $N_{98} = \alpha n^\beta \ln n$ to the experimental data yielded $N_{98} = 361 n^{0.37} \ln n$ for $\lambda = 500$ ($R_a^2 = 0.99$) and $N_{98} = 650 n^{0.38} \ln n$ for $\lambda = 1000$ ($R_a^2 = 0.99$), confirming the validity of the estimate $\sqrt{n} \ln n$ as. Larger populations are more expensive, but smaller population may fail to converge for large $n$. This is the case when a population of 50 individuals is used to solve the Max $A_{11}$ problem and $n \geq 50$: the average maximum fitness never reaches 98% of the maximum fitness. This can be explained by the fact that when smaller populations are used, the chance of losing particular values of the variables is higher, which prevents the algorithm from finding the optimum, as was discussed in Section 3.2.2.

The effect of the loss of variable values for small populations is visible in the
reliability: for each problem size $n$, there exists a minimum population size below which 80\% reliability is never reached because of premature convergence of the distributions. This minimum population size was $m^* = 100$ for $n = 12$, $m^* = 500$ for $n = 20, 50$ and $100$, $m^* = 1000$ for $n = 200$.

The algorithms were then applied to the vibration problem for the same five problem sizes. The number of evaluations necessary for the average maximum fitness function to reach 98\% of the optimal fitness is shown in Figure 3–11. The results are similar to those obtained for the Max $A_{11}$ problem. The cost of SHC is still close to linear in the number of variables. However, the number of evaluations needed by the two algorithms to reach 98\% of the maximum fitness is smaller than on the Max $A_{11}$ problem. For instance, for $n = 12$, SHC needs 64 evaluations on the vibration
problem, against 160 on the Max $A_{11}$ problem. In the case $n = 200$, it requires 1,402 analyses for the vibration problem, against 2,923 for the Max $A_{11}$ problem. Similarly, the number of analyses needed by UMDA with a population of 1000 individuals decreases from 4,175 analyses to 2028 analyses for $n = 12$ and from 26,322 analyses to 17,531 analyses for $n = 200$. The faster convergence toward high fitness regions for the second problem can be explained by the fact that a large part of the response is governed by outermost plies, so that most of the fitness improvement can be achieved by determining the value of these influential plies. In addition, the fact that the optimum angle ($60^\circ$) is close to the center of the domain helps SHC by reducing the average number of steps it has to take.

Figure 3–11: Number of analyses until the average maximum fitness reaches 98% of the optimal fitness, vibration problem. When the influence of some variables on the fitness function is stronger than some others’, the effect of dimensionality on UMDA’s effectiveness is reinforced.

If the hierarchical structure of the problem allows a rapid convergence to high fitness regions, it also causes numerical difficulties for UMDA. The convergence of the
probability distribution for the innermost and outermost plies for the case \( n = 100, m = 500 \) are presented in Figure 3–12. On this hierarchical problem, it appears very clearly that the algorithm proceeds from the outside to the inside, starting with the more influential variables, and determining the inner variables only at the very end. This mechanism is responsible for the loss of variable values for the less influential inner plies: in the early stages of the search, the selection of good points is dominated by the outermost variables: the mean of the fitness of dominated variables is only slightly greater for the optimum value than for non-optimum values, and its variance is large, as was remarked in Section 2.4. As a result, points which contain the optimum value of the outermost plies but not of the innermost plies get selected, potentially leading to the disappearance of these values in the distribution if too small a population is used. We observed that larger populations have to be used in order to prevent the loss of values. The minimum population sizes for this problem were \( m^* = 500 \) for \( n = 12 \) and \( n = 20 \), \( m^* = 1000 \) for \( n = 50 \). In the cases \( n = 100 \) and \( n = 200 \), the algorithm did not reach 80% reliability for the population sizes tested.
within the maximum number of 40,000 analyses used in this work.

SHC, however, is unaffected by the problem’s hierarchical structure because it merely compares two neighbor points. Figure 3–13 shows the two performance measures on the two problems. Both the number of evaluations to 98% of the maximum fitness and the number of evaluations until 80% reliability is achieved are lower for the vibration problem than for the Max $A_{11}$ problem: the location of the optimum in a center area of the design space makes the vibration problem easier for SHC.

![Graph comparing performance of SHC on Max $A_{11}$ and vibration problems.](image)

Figure 3–13: Comparison of the performance of SHC on the Max $A_{11}$ problem and the vibration problem.

This study reveals that the potential benefits of a probabilistic optimization algorithm do not lie in their performance in the face of increasing dimensionality: if it is true that for a given population size, UMDA would asymptotically outperform SHC, which needs close to linearly increasing numbers of random trials to find the optimum. To maintain a sufficient accuracy of the statistical model, large populations have to be used in UMDA, thereby neutralizing the advantage of that model. Like
other EAs, EDAs are costly algorithms. Their advantage is observed on problems that cannot be handled by simpler algorithms, such as multimodal problems, as will be demonstrated in Section 3.3.

### 3.3 Investigation of possible modifications to UMDA: memory and mutation

We have seen in Section 3.2.2 that UMDA variants that had a small population size and a large selection pressure exhibited the fastest convergence, but that they also displayed a poor reliability because when a value disappeared from the distribution at a given iteration, no recovery mechanism allowed it to be reintroduced at a later generation. When the lack of diversity injection mechanism is associated with a very selective scheme that converges rapidly to a small number of values, the algorithm becomes strongly exploitative, and premature convergence is very likely.

The reliability of these variants may be improved by three different strategies: slowing down the convergence of the probabilities using memory, providing an allele recovery mechanism in the form of mutation, or specifically not allowing probabilities to vanish. This section investigates the effect of memory, mutation, and of imposing lower bounds on the probabilities on the performance. The tests were performed on the 10-variable $A_{11}$ maximization problem introduced in Section 3.2.1.

#### 3.3.1 Memory

A simple way of preventing premature loss of diversity due to sampling errors is to use a conservative updating method that makes it possible to adjust the influence of incoming observations, as proposed by Baluja (1994). The new probability distribution $p_{t+1}^{t+1}(x_k)$ is obtained as a linear combination of the old distribution $p_t^t(x_k)$ and
the observed frequencies $f_k$:
\[
  p_{ki}^{t+1} = \frac{1}{m+1} (m p_{ki}^t + f_{ki}) ,
\]
(3.19)
where $m$ is a “memory” parameter: when $m = 0$, the new probabilities are simply
the marginal frequencies (standard UMDA), when $m \neq 0$, the algorithm mitigates
potential sampling errors in the frequencies by giving them a smaller weight. The
value of $m$ reflects the user’s confidence in the frequencies’ accuracy.

To investigate the effect of memory, we considered two variants of UMDA: $A_1$, the
best scheme found in Section 3.2.2 \{µ = 50, λ = 166\} and $A_2$, the algorithm
that displayed the fastest initial convergence \{µ = 10, λ = 33\} (see Figure 3–5). Six
values of $m$ were tested: $m = 0, 1, 2, 5, 10, \text{and } 20$. The reliability of the two variants
for the five values of memory are compared to the basic UMDA in Figure 3–14. Not

Figure 3–14: Effect of memory on UMDA with two parameter settings, max $A_{11}$
problem ($n = 10$): \{µ = 10, λ = 33\} (solid lines) and \{µ = 50, λ = 166\} (dashed lines).
surprisingly, adding memory to the best scheme $A1$ caused the reliability to deteriorate: the population sizes had been adjusted for the simple UMDA, but they are no longer appropriate when substantial modifications are made to the algorithm, hence the decrease in reliability. In contrast, $A2$’s reliability improves dramatically when memory is used: the final reliability increases monotonically with $m$. However, this improvement is achieved at the cost of convergence velocity, as large memory clearly adds inertia to the search by giving a large weight to past observations. As a result, the benefit of memory is debatable, and increasing population sizes may be a more effective way of preventing distribution degeneracy, as suggested by the comparison between the memoryless $A1$ and the $A2$ versions, which incorporate memory.

### 3.3.2 Mutation

A customary strategy for preventing premature convergence in evolutionary algorithms is to use a perturbation operator called mutation. In the context of estimation of distribution algorithms, two approaches can be adopted to implement mutation: one can either apply the perturbations to the points or directly to the distribution. Few instances of mutation are provided in the literature. Baluja (1994) proposed a mutation consisting in shifting the probabilities by a fixed value with a given probability. He concluded that while some improvement could be achieved, the operator was not as critical as in standard genetic algorithms. In this work, we chose the alternate approach of applying perturbations to the points generated by the (unperturbed) probability distribution. Indeed, in EDAs, the statistical model reflects our knowledge of promising regions, hence there is no reason to degrade this information
arbitrarily. We prefer to provide a separate diversity injection mechanism: if the proposed perturbation is validated through selection, it will be incorporated into the model at the next iteration.

The mutation operator used in this work was an “adjacent mutation”, where the value of each variable $x_k$ was changed with probability $p_m$ to one of the two neighboring values\(^6\) (for instance, 30° could be changed into 15° or 45° with equal probabilities if the values were chosen from $\{0°, 15°, 30°, 45°, 60°, 75°, 90°\}$).

UMDA was applied to the 10-variable $A_{11}$ maximization problem presented in the previous sections with seven values of the mutation rate: $p_m = 0, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, \text{ and } 0.3$. As in the previous section, two schemes were considered, $A1$, the best scheme found in Section 3.2.2 ($\{\mu = 10, \lambda = 33\}$) and $A2$, the algorithm that displayed the fastest initial convergence ($\{\mu = 50, \lambda = 166\}$). Figure 3–15 shows the influence of the mutation rate on the reliability of the algorithm for the two variants.

Like memory, mutation had a negative impact on $A1$, because the values of $\lambda$ and $\mu$ chosen to maximize the reliability of UMDA were no longer the best in a modified context. In contrast, using mutation in combination with $A2$ led to a dramatic improvement of the reliability of the algorithm: while the original algorithm had a very low maximum reliability of 16%, even with a low mutation rate of 0.005 the

\(^6\) The boundary values 0° and 90° required a special treatment: when mutation was applied to these values, the variable had equal chances of remaining unchanged, or being shifted to the next value (15° for an initial value of 0°, 75° for an initial value of 90°).
Figure 3–15: Influence of mutation on the reliability for two UMDA variants: the best scheme without mutation $\{\mu = 50, \lambda = 166\}$, and the one that displays the fastest initial convergence $\{\mu = 10, \lambda = 33\}$. Mutation dramatically improves the former variant’s performance by preventing complete disappearance of the optimal variable values, but affects the latter’s reliability negatively, because it perturbs the estimation of distributions.

algorithm’s reliability reaches 80% in approximately the same number of evaluations as A1. The reliability can be further improved by increasing $p_m$, the best performance being observed for $p_m = 0.2$. Beyond this value, a sharp decline in the reliability takes place. The low sensitivity of the reliability to the mutation rate $p_m$ over a wide range of values constitutes a positive attribute, as it reduces the amount of tuning necessary to use the algorithm.

Unlike memory, mutation did not affect the convergence velocity substantially. when $p_m$ is low, mutation acts as a background operator that compensates for detrimental effects of working with a finite population. The role of mutation in the optimization can be observed in Figure 3–16: in the first five iterations, it prevents the optimal value $x_1 = 0^\circ$ from disappearing from the distribution; the algorithm then
identifies it as the optimal value through the process of selection and distribution estimation.

Figure 3-16: Effect of mutation for $p_m = 0.005$. Thanks to mutation, the optimal value $x_1 \equiv \theta_1 = 0$, which would otherwise have disappeared from the distribution, is reintroduced through mutation. Since the perturbation level is low, the distribution eventually converges to the optimal value.

### 3.3.3 Bound on the probabilities

We have seen in previous sections that one of the major concerns in estimation of distribution algorithms was to maintain the algorithm’s ability to visit any point in the design space. This can be most easily achieved by formally preventing marginal probabilities to fall below a threshold $\epsilon$. Let $p_{ki} = p(x_k = a_i)$. The following procedure was implemented for each marginal distribution $p(x_k)$:

1. if a probability $p_{ki}$ is smaller than $\epsilon$, it is set to $\epsilon$;

2. the sum of the differences $\epsilon - p_{ki}$ of all the corrected probabilities is evenly distributed over the other values.
The effect of imposing a lower bound on the marginal probabilities was investigated for the two UMDA schemes used in previous sections. The reliability for three values of $\epsilon$ is shown in Figure 3–17 (values of $\epsilon$ are given as fractions of the uniform probability $p_{ki} = 1/c$). As for the other diversity injection mechanisms presented previously, imposing bounds on the marginal probability had an adverse effect on the reliability of the already close to optimal scheme $A1$. However, it had a dramatic impact on $A2$: with values of $\epsilon$ between $\frac{1}{40c}$ and $\frac{1}{10c}$, the reliability reached 100% in less than 2,000 evaluations without appreciable deterioration of the convergence velocity.

### 3.3.4 Elitism

A strategy often used to guarantee a monotonic increase of the fitness in evolutionary computation is elitism, which consists in automatically copying the best
parent in the child population. The underlying principle is that the population constitutes the memory of past observations, hence good individuals encountered during the search should not be lost, as they are evidence for good regions. In the EDA framework, the notion of individual is less important, as the utility of populations of points is only to characterize the distribution of promising regions; after information has been extracted from good points, the population becomes useless. However, some authors have proposed replacement strategies allowing a portion of a population to be kept for the next iteration (e.g. Cho and Zhang, 2002; Bosman and Thierens, 2000).

The influence of elitism was investigated for the two UMDA schemes $A_1$ and $A_2$: at each iteration, $\lambda - 1$ were created by sampling from $p(x)$, and the best point of the previous generation was added to obtain a population of $\lambda$ points. The reliability of elitist algorithms is compared to that of non-elitist algorithms in Figure 3–18. Elitism did not help for either of the two UMDA schemes considered: in the case of $A_1$, the
population sizes were large enough to guarantee a good accuracy of the distribution, so that no information about good regions was lost and elitism was not needed; in the case of A2, copying the best point of a population in the next iteration’s population reinforced the algorithm’s tendency to premature convergence, which caused the reliability to further deteriorate.

3.3.5 Conclusion of the parameter study

In this section, four possible modifications to the original univariate marginal distribution algorithm were investigated. The main goal was to provide a mechanism to prevent premature convergence of the search distribution. Three different approaches were proposed: memory, mutation, and a lower bound on the marginal probabilities. All three strategies yielded substantial performance improvement. In particular, they made the use of small populations viable, thereby allowing savings of function evaluations, either by allowing lost values to be recovered, or by preventing the loss of variable values. The three approaches share the common characteristic that they can hamper final convergence when used too massively. This calls for adaptive strategies to decrease the influence of these perturbing operators over time, which is beyond the scope of this study. Finally, a study of the effect of elitism was carried out for complet this strategy commonly did not lead to a performance improvement for the two variants investigated.

3.4 Comparison on three problems

In this section, we compare the performance of UMDA to two other stochastic optimization algorithms: a stochastic hill-climber, and a standard genetic algorithm.
The three algorithms are applied to three laminate optimization problems carefully chosen to assess the relative performance of the algorithms on representative fitness landscapes.

3.4.1 Presentation of the algorithms

An UMDA with truncation selection of ratio $\tau = 0.3$ was compared to two stochastic optimization algorithms: a stochastic hill-climber (SHC), presented in Section 3.2.3, which is a point-based random search algorithm, and a standard genetic algorithm (GA) (see Appendix D for an introduction to genetic algorithms) with rank proportional roulette selection (i.e. linear ranking) selection and two-point crossover with probability $p_c = 1.0$. No elitist strategy was implemented in UMDA and GA: we want to compare the fundamental search mechanisms of the algorithms. The study was primarily focused on UMDA and SHC: a limited parameter study was performed to obtain good values of the mutation rate $p_m$ and population size $\lambda$. GA’s performance for the best UMDA setting is provided to allow readers more familiar with that algorithm to assess UMDA’s performance.

3.4.2 Constrained maximization of the first natural frequency

The first problem was maximizing the first natural frequency of a simply supported graphite-epoxy laminated plate of length $L = 50''$ and width $W = 15''$ subject to a constraint on the effective Poisson’s ratio $\nu^l \leq \nu_{\text{eff}} \leq \nu^u$, with $\nu^l = 0.48$ and $\nu^u = 0.52$. The first natural frequency is given by

$$F = \frac{\pi^2}{\sqrt{\rho_m h}} \sqrt{\frac{D_{11}}{L^4} + \frac{2(D_{12} + 2D_{66})}{L^2 W^2} + \frac{D_{22}}{W^4}}$$

(3.20)
where \( h \) is the total laminate thickness, \( \rho_m \) designates the mass density, and the \( D_{ij} \)'s are the bending stiffness coefficients.

The effective Poisson’s ratio is given by

\[
\nu_{\text{eff}} = \frac{A_{12}}{A_{22}} = \frac{U_1 - U_2 V_1^* + U_3 V_3^*}{U_4 - U_3 V_3^*} \tag{3.21}
\]

where the in-plane lamination parameters \( V_1^* \) and \( V_3^* \) are obtained by

\[
V_1^* = \frac{1}{n} \sum_{k=1}^{n} \cos 2\theta_k, \quad V_3^* = \frac{1}{n} \sum_{k=1}^{n} \cos 4\theta_k. \tag{3.22}
\]

The material properties used for this problem are shown in Table 3–3.

<table>
<thead>
<tr>
<th>Material property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longitudinal modulus ( E_1 )</td>
<td>( 2.18 \times 10^7 ) psi</td>
</tr>
<tr>
<td>Transverse modulus ( E_2 )</td>
<td>( 1.38 \times 10^6 ) psi</td>
</tr>
<tr>
<td>Shear modulus ( G_{12} )</td>
<td>( 1.55 \times 10^5 ) psi</td>
</tr>
<tr>
<td>Poisson’s ratio ( \nu_{12} )</td>
<td>0.26</td>
</tr>
<tr>
<td>Weight density ( \rho )</td>
<td>0.057 lb/in(^3)</td>
</tr>
<tr>
<td>Ply thickness ( t )</td>
<td>0.005 in</td>
</tr>
</tbody>
</table>

None of the three algorithms accommodates constraints, therefore a penalty approach was used, where the fitness function \( F_p \) of infeasible designs was decreased in proportion to the constraint violation:

\[
F_p = \begin{cases} 
F(x) & \text{if } g(x) \leq 0 \quad \text{(feasible)} \\
F(x) - pg(x) & \text{if } g(x) > 0 \quad \text{(infeasible)} 
\end{cases} \tag{3.23}
\]

where \( p \) is the penalty parameter whose value was adjusted empirically to ensure that the algorithm yielded feasible designs. We used \( p = 2 \times 10^3 \) for this study. The constraint term \( g(x) \) was defined as

\[
g(x) = \max \left( 1 - \frac{\nu_{\text{eff}}(x)}{\nu^i}, \frac{\nu_{\text{eff}}(x)}{\nu^u} - 1 \right). \tag{3.24}
\]
The constraint on the Poisson’s ratio forces the points to remain in a narrow channel in the design space, which makes the problem particularly difficult for hill-climbing algorithms because many of the random perturbations result in infeasible designs.

Clearly, this problem did not satisfy the variable independence assumption, as the value of one variable influences the optimal value of other variables through the constraint.

Two numbers of variables were considered: \( n = 8 \) and \( n = 15 \). Without the constraint, the optimal orientation would be \( 90^\circ \) for all the plies. The effective Poisson’s ratio would then be \( \nu_{\text{eff}} = \nu_{21} = 0.0165 \). The Poisson’s ratio constraint forces \( 30^\circ \), \( 45^\circ \) \( 60^\circ \), and \( 75^\circ \) plies into the inner layers of the laminate, where they are the least damaging to the frequency. The optimum\(^7\) for \( n = 8 \) was \([90_2/ \pm 75/\pm 45_5/ \pm 30]_s\), which has a first natural frequency of \( F = 670 \text{ Hz} \) and an effective Poisson’s ratio of \( \nu_{\text{eff}} = 0.482 \). For \( n = 15 \), the optimum was \([90_4/ \pm 75/ \pm 60_2/ \pm 45_5/ \pm 30_5]_s\), with \( F = 1,262.6 \text{ Hz} \) and \( \nu_{\text{eff}} = 0.481 \).

For \( n = 8 \), two population sizes, \( \lambda = 20 \) and \( \lambda = 50 \), and three mutation rates, \( p_m = 0.1 \), \( p_m = 0.2 \) and \( p_m = 0.3 \) per variable were tried for UMDA. In order to select the best scheme, the reliability reached at 2000 evaluations was used as criterion. The highest reliability (88%) was achieved with \( \lambda = 20 \) and \( p_m = 0.2 \). A similar parameter study was conducted for SHC with \( p_m \) ranging from 0.1 to 0.5, and the highest

---

\(^7\) For this problem, as for all the problems addressed in this work, the best solution found in all the runs performed is considered as the optimum, unless the global optimum is known a priori.
reliability was achieved for $p_m = 0.4$. Figure 3–19 compares the mean best fitness and reliability of the three algorithms for these best settings (the same population size and mutation rate were used for GA and UMDA, to ensure comparable variability of the search distributions). SHC converges faster than GA and UMDA to high fitness regions, as shown in Figure (a), however this advantage does not translate into a higher reliability than UMDA. SHC’s failure to finalize the search can be attributed to the high mutation rate, which is an indication that an adaptive scheme may improve the performance. GA’s apparent poor reliability is the result of a weaker selection procedure (linear ranking, cf. Section 2.4). However, considering that it is one the most commonly used, the observed behavior can arguably be regarded as representative of a typical GA. Clearly, both SHC and UMDA find the optimum more reliably than GA on this example.

For $n = 15$, a population size of $\lambda = 100$ individuals and a moderate mutation
probability $p_m = 0.1$ were chosen. The larger population is justified because more individuals are needed to ensure that all variable values are present in the population.

For SHC, the best mutation probability was found to be $p_m = 0.2$ (tested values were $p_m = 0.1, 0.2, 0.3, 0.4, 0.5$). Figure 3–20 compares the performance criteria for the three algorithms. This time, UMDA seemed to benefit from the use of a global probabilistic model, which allows it to escape local minima and was able to reliably find the global optimum: after 10,000 evaluations, the reliability of the optimization had reached 64%, whereas the SHC only found the optimum in 30% of the runs. In 70% of the runs, SHC converged to a high quality solution but failed to yield the true optimum. For instance, one of the solutions was $[90_6 / \pm 60 / \pm 45_{10} / \pm 30]_s$, which had a fitness of $F_p = 1,279.3$ ($F = 1,257.9$, $\nu = 0.481$). In order to obtain the global optimum $[90_4 / \pm 75 / \pm 50_2 / \pm 45_5 / \pm 30_5]_s$ ($F_p = 1,262.6$, $F = 1,262.6$, $\nu = 0.481$), six variables have to be mutated. However, all single mutations lead to a reduction
in the fitness function, either because they make the design infeasible (variable 5) or because they decrease the vibration frequency. Consequently, multiple mutations must occur simultaneously for the fitness function to improve. The probability of that event decreases as \( n \) increases, thus making further progress of SHC unlikely. These results agree with He and Xin (2002) in which an SHC was shown to have an exponential time complexity for a multimodal function, while the cost of a population based EA was only polynomial.

### 3.4.3 Minimize \( A_{66} \)

SHC is misled by local minima, while the relationship between UMDA and the objective function is more complex. It is nevertheless known that, if the objective function is separable\(^8\) and the population size is larger than a critical value \( \lambda^* \), UMDA converges to the global optima (Mühlenbein and Mahnig (1999)). An example of such an objective function where the reliability of UMDA tends to 1 while that of SHC is nearly 0 is the minimization of the in-plane shear stiffness of a composite laminate, \( A_{66} \), over ply orientations that are bounded between \( 0^\circ \) and \( 75^\circ \),

\[
\min_{0^\circ \leq x_i \leq 75^\circ} A_{66}, \tag{3.25}
\]

where

\[
A_{66} = U_5 h - U_3 \sum_{i=1}^{n} t_i \cos 4x_i. \tag{3.26}
\]

\(^8\) A more general result is given in Mühlenbein and Mahnig (1999) where the convergence of “Factorized Decomposition Algorithm” (FDA) to the optima is proved for additively decomposed functions. Separable functions are a special case of additively decomposed functions and UMDA is the corresponding simplified FDA.
The global optimum is \( x^*_i = 0^\circ, \ i = 1, n \). Replacing any of the \( x^*_i \) with 75° creates a local optimum whose basin of attraction starts at \( x = 45^\circ \). For an \( n \)-dimensional case, there are \( 2^n \) local optima. Numerical experiments with \( n = 12 \) averaged over 50 runs confirm that the SHC reliability is 0 (it is theoretically \((45/75)^{12} = 2.10^{-3}\)) while that of an UMDA with \( m = 500 \) reaches 1 after 3,500 analyses. Figure 3-21 shows the evolution of the probability distribution of the outermost ply \( p(x_1) \). It is interesting to note that in the early stages of the search, both the probability of 75° and the probability of 0° (the two local optima) increase. But after about 2,000 analyses, the probability of 75° starts to decrease and the algorithm converges to the global optimum.

![Figure 3-21: Evolution of the marginal distribution \( p(x_1) \), UMDA, max \( A_{66} \) problem.](image)
3.4.4 Strength maximization

Many practical laminate optimization problems exhibit a multimodal relationship between the variables and the fitness function. A typical example of such behavior is the maximization of the strength of a laminate. In this section, we considered the problem of maximizing the load factor $\lambda_s$, using the first-ply-failure criterion based on the maximum strain (Gürdal et al., 1998, Chapter 6), for a glass-epoxy laminate subjected to the in-plane loading $N_x = -1000 \times 10^3$ N/m, $N_y = 200 \times 10^3$ N/m, $N_{xy} = 400 \times 10^3$ N/m:

$$\text{maximize } \lambda_s = \min_{k=1}^{n} \left( \min \left( \frac{\epsilon_{\text{ult}}}{\epsilon_1(k)}, \frac{\epsilon_{\text{ult}}}{\epsilon_2(k)}, \frac{\gamma_{\text{ult}}}{\gamma_{12}(k)} \right) \right)$$

where the load factor $\lambda_s$ is the coefficient by which the load has to be multiplied for the structure to fail. The material properties used for this problem are shown in Table 3–4. The total thickness of the laminate was $h = 0.02$ m.

Table 3–4: Material properties of glass-epoxy.

<table>
<thead>
<tr>
<th>Material property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longitudinal modulus $E_1$</td>
<td>69.0 GPa</td>
</tr>
<tr>
<td>Transverse modulus $E_2$</td>
<td>10.0 GPa</td>
</tr>
<tr>
<td>Shear modulus $G_{12}$</td>
<td>4.5 GPa</td>
</tr>
<tr>
<td>Poisson’s ratio $\nu_{12}$</td>
<td>0.31</td>
</tr>
<tr>
<td>Material invariant $U_1$</td>
<td>31.58 GPa</td>
</tr>
<tr>
<td>Material invariant $U_2$</td>
<td>27.91 GPa</td>
</tr>
<tr>
<td>Material invariant $U_3$</td>
<td>6.48 GPa</td>
</tr>
<tr>
<td>Material invariant $U_4$</td>
<td>9.63 GPa</td>
</tr>
<tr>
<td>Material invariant $U_5$</td>
<td>10.98 GPa</td>
</tr>
<tr>
<td>Tensile strength $X_t$</td>
<td>500.0 MPa</td>
</tr>
<tr>
<td>Compressive strength $X_c$</td>
<td>410.0 MPa</td>
</tr>
<tr>
<td>Tensile strength $Y_t$</td>
<td>35.0 MPa</td>
</tr>
<tr>
<td>Compressive strength $Y_c$</td>
<td>110.0 MPa</td>
</tr>
<tr>
<td>Shear strength $S$</td>
<td>70 MPa</td>
</tr>
</tbody>
</table>

We considered the case $n = 8$ design variables. The laminate is subjected to an
in-plane loading: \( N_x = -1000 \text{ kN/m}, \ N_y = 200 \text{ kN/m}, \ N_{xy} = 100 \text{ kN/m}. \) For this problem, the optimum laminate was \([0/90]_s\) or its permutations, and the optimum load factor was \( \lambda_s = 5.39. \) Depending on the orientation of the fibers, one of the three possible failure modes (fiber failure, matrix cracking, shear failure) becomes critical. The combination of these three failure modes results in a multimodal fitness function\(^9\).

Figure 3–22: Reliability (a) and mean best fitness (b) of UMDA and SHC for the strength problem. Unlike SHC, which was trapped in local optima, UMDA was able to find the global optimum reliably. When the maxima of the objective function are far apart in the Hamming space, SHC is not able to escape local maxima, hence its poor reliability. In contrast, UMDA’s global search strategy allows it to locate high function evaluation regions and to reliably find the global maximum.

For UMDA the population size was \( \lambda = 50 \) and a moderate mutation rate of \( p_m = 0.1 \) per variable was implemented to prevent premature convergence. Several variants of SHC were compared to obtain the best competitor to UMDA. For this

\(^9\) Note that it is not necessary to consider all the failure modes for the response to be multimodal, as individual failure modes can be multimodal for specific combinations of material properties and loading.
problem, the same adjacent mutation operator was implemented and the best rate was $p_m = 0.2$ (the same values of $\lambda$ and $p_m$ were used for GA). Both the reliability (Figure 3–22-a) and the mean best fitness (Figure 3–22-b) clearly show the superiority of UMDA for this problem. The reliability of SHC increases faster than that of UMDA, but culminates at 36%, while UMDA was able to converge reliably to the optimum. SHC often converged to local optima ($[0_8/90_8]_s$, $\lambda_s = 4.97$, $[0_6/ \pm 30/90_8]_s$, $\lambda_s = 4.47$ or $[0_8/ \pm 30/90_6]_s$, $\lambda_s = 4.02$) and was not able to escape even when large mutation rates were used because unlikely coordinated mutations were needed to reach the basin of attraction of the global optimum.

### 3.5 Conclusion

Univariate marginal distribution algorithms constitute the simplest form of estimation of distribution algorithms. They estimate distributions based on the assumption that the variables are statistically independent in the populations of selected points (but they are often applied to problems that violate this assumption). In this chapter, the influence of major components of the algorithm was investigated on a laminate optimization problem. The selection pressure determines the convergence velocity, while the population size affects the variance of the marginal frequencies, hence the accuracy of the search distribution. UMDA’s asymptotic behavior was compared to that of a hill-climbing algorithm. The study concluded that the advantage of UMDA does not lie in pure convergence velocity. Instead, its global search capabilities become beneficial for functions that exhibit narrow channels, or multimodal landscapes, as demonstrated on three laminate optimization example problems.
CHAPTER 4
THE DOUBLE-DISTRIBUTION OPTIMIZATION ALGORITHM

In this chapter, we propose a physics-based method for incorporating variable dependencies into the probabilistic model of promising regions. The apparent complexity of the distribution of selected points can often be explained as the joint action of a small number of latent variables. The distribution of selected points can be reconstructed by the cooperation of simple models of the primary variables and of the latent variables, used as auxiliary variables. This representation of the distribution of promising points is the basis of the Double-Distribution Optimization Algorithm, introduced here.

4.1 Motivation

In Chapter 3, the distribution of selected points was approximated by a product of univariate marginal distributions, which neglected any statistical variable dependencies. While this representation yielded comparable to superior performance than a standard evolutionary algorithm, it may not be appropriate for problems with strong variable interactions. Figure 4–1 illustrates the impact of the choice of a univariate model when the distribution generated by selection is more complex. In the 2D problem considered (which is the problem of maximizing the first natural frequency subject to a constraint on Poisson’s ratio introduced in Chapter 3), the contours of the objective function assume roughly the shape of a narrow ridge. After uniform sampling in the \((\theta_1, \theta_2)\)-plane, and application of truncation selection, the promising
Figure 4–1: Selected points (a) and univariate distribution (b) for $(\theta_1, \theta_2) \in \{0, 5, 10, \ldots, 85, 90\}^2$ for the constrained vibration problem of Chapter 3. When the distribution of selected points is approximated by a product of the univariate marginal distributions $p(\theta_1)$ and $p(\theta_2)$, high-probability regions do not coincide with high-fitness regions. 

points are distributed as shown in Figure 4–1-a. Ideally, one would want to sample from the high evaluation regions marked by the selected points. However, when the joint distribution $p(\theta_1, \theta_2)$ is approximated by a product of the univariate marginal distributions $p(\theta_1)$ and $p(\theta_2)$, new points will be created according to the distribution whose histogram is shown in Figure 4–1-b. Clearly, the high probability areas defined by that distribution do not coincide with high evaluation regions, so that some knowledge about optimal regions contained in the selected points will be lost.

---

1 In a univariate model, maximum probability areas are parallel to the axes. For example, for $\mathbf{x} = (x_1, x_2)$, the maximum probability at a given value of $x_2$ is $\max_{x_1} p(x_1)p(x_2) = p(x_2) \max_{x_1} p(x_1) = p(x_2)p(x_1^*)$, where $p(x_1^*)$ is the maximum of the marginal distribution $p(x_1)$: the value of $x_1$ that maximizes $p(x_1, x_2)$ does not depend on $x_2$. 
for the next EDA generation.

One way of improving the distribution accuracy would be to use a higher-order statistical model that incorporates the conditional probabilities $p(\theta_2|\theta_1)$. In higher dimensions, this approach could be generalized by using a Bayesian network to represent all conditional probabilities $p(\theta_i|\pi_i)$, where $\pi_i$ designates the parents of the variable $\theta_i$, as proposed by Pelikan et al. (1999) in the Bayesian Optimization Algorithm. However, this approach presents three strong limitations:

- the number of parameters required to describe high-order dependencies in the distribution increases exponentially with the number of variables $n$ ($c^n - 1$ parameters are necessary to express the full joint distribution when each of the $n$ variables can take $c$ values). The computational cost associated with the estimation of those parameters may become prohibitive as $n$ becomes large;

- more importantly, the standard error associated with the parameters increases with the flexibility of the model. While a flexible model can asymptotically approximate complex distributions (a fortiori simple distributions), obtaining an accurate estimation of the parameters may require a very large sample (cf. Section 2.5.3). In the context of EDAs, this implies that large population may have to be used, so that larger progress may be accomplished at each iteration, but at the cost of many function evaluations;

- lastly, using a too high-order model violates the premises of optimization, which is based on the assumption that there exists an underlying structure present in the distribution $p(\mathbf{x})$: the goal is to infer and use this structure to find the optima. This implies a trade-off between accuracy and generalization from the
sample: high-order models will provide an accurate estimation on the “training set”, but extrapolation to other regions of the design space will be poor.

Alternative approaches are used routinely by statisticians in the field of exploratory data analysis. The idea is to simplify the data by looking for structure in the distributions, as will be presented in the next section.

4.2 Principles

4.2.1 Identification of dependencies and data simplification

A typical example of an effort to distinguish between data structure and random variations is exploratory factor analysis (EFA), where the goal is to identify underlying (latent) factors that determine the distribution of a population, and use them to describe the distribution, thereby as shown in Figure 4–2. By supposing that

\[
V_1, V_2, \ldots, V_m
\]

\[
X_1, X_2, X_3, \ldots, X_n
\]

Figure 4–2: Interpretation of variable dependencies as the joint action of hidden variables \(V_1, \ldots, V_m\). Even if the hidden variables have a simple probability distribution, the distribution of the observed variables (the \(X\)’s) can be complex.

the (possibly complex) observed distribution is the result of joint actions of a small number \(m\) of latent variables, whose distribution is simple, one is able to simultaneously reduce the number of meaningful variables, as well as the complexity of the distribution (the low dimensional distribution of independent latent variables are easier to estimate than the joint distribution of the observed variables). Such methods
were implemented by Shin et al. (2001), using several latent variable models: they modeled the distribution of selected points by Helmolz machines, and probabilistic principal component analysis (PPCA) to represent apparently complex distributions as the result of the joint action of several latent variables, whose distribution is simple (multivariate normal).

This general method is appropriate when nothing is known about the structure of the distribution, however it only allows simple (typically linear) relationships to be identified, and learning even these relationships has a computational cost. In many situations, a priori knowledge about the problem is available. The benefits of providing the distribution structure was investigated by Baluja (2002), who showed that it led to drastic performance improvement. In this work, we propose a fully different approach for incorporating information about the structure to enhance the accuracy of the estimated distribution by directly providing the algorithm with its structure.

In many instances, variable dependencies among selected points often reflect the fact that the overall response of the system is really a function of integral quantities, so that many combinations of the design variables can produce the same response. For example:

- the dimensions of the section of a beam determine its flexural behavior through the moment of inertia $I$,
- the aerodynamic properties of a vehicle are captured by the drag coefficient $C_D$,
- the flow through a porous medium is described by the permeability $k$,
- etc.
These quantities are often inexpensive to calculate, and their number is limited and insensitive to the number of design variables. The proposed strategy combines an independent model for the primary variables $x_k$, and a complementary model of the auxiliary variables $f(V)$ as shown in Figure 4–3. The distribution of auxiliary variables introduces information about statistical variable dependencies into the estimated model of primary variables$^2$.

![Figure 4–3: Incorporation of variable dependencies through auxiliary variables.](image)

### 4.2.2 Sampling from two distributions

Now that the principles have been established, the actual procedure implemented in the optimization algorithm needs to be specified. The goal is to devise a strategy by which populations of $\lambda$ points reflecting the two distributions $p(x)$ and $p(V)$ can be

---

$^2$ A related strategy consisting in using a simple model, which is incrementally improved by incorporating information about the distribution of particular combinations of the variables is used in statistics, namely in estimation of distribution by projection pursuit (Friedman et al., 1984).
created\textsuperscript{3}. The particular strategy adopted determines the relative influence of either distribution in the creation process. Three different schemes have been investigated in this work: in all cases, the general principle is to (1) create a large pool of $\nu$ candidate points by sampling from $p(x)$, and (2) retain $\lambda$ points based on the distribution $p(V)$. The three methods differ in the procedure they use to decide which points to keep.

**Deterministic acceptance**

The probability $p(V)$ of each of the $\nu$ candidate points in the auxiliary variable space is calculated. Then, the $\lambda$ points having the higher evaluations of $p(V)$ are accepted as the new population. Preliminary tests indicated that this strategy produced a very conservative algorithm, which did not allow the exploration of new regions.

**Probabilistic acceptance**

The probability $p(V)$ of each of the $\nu$ candidate points in the auxiliary variable space is calculated. Then, $\lambda$ points are drawn among the pool of candidates, with a probability proportional to $p(V)$:

$$p(V_i) = \frac{p(V_i)}{\sum_{j=1}^{\nu} p(V_j)}.$$  

\textsuperscript{3} Unlike standard factor analysis methods, which are generative because the observed variables are explicit functions of the hidden variables, the proposed two-distribution approach does not provide a direct way of generating samples.
This approach is attractive, as the final probability distribution under this procedure is simply obtained as the product of the two distributions:

\[ p_{\text{final}}(x_1, \ldots, x_n) \propto \left( \prod_{k=1}^{n} p(x_k) \right) p(V_1(x_1, \ldots, x_n), \ldots, V_m(x_1, \ldots, x_n)) . \]  

(4.1)

This method is simple to analyze, and makes a clear parallel with well-established methods, such as projection pursuit density estimation (Friedman et al., 1984). However, the approach presents two disadvantages: first, even though the resulting distribution constitutes a compromise between the two distributions, the respective contribution of each distribution to the final distribution is arbitrarily fixed. There is no theoretical justification for this imposed compromise. Second, such a scheme leads to a conservative algorithm, as the acceptance procedure results in a reduction of the population variability even when the variance of the selected point distribution is greater than that of the candidate point distribution, as illustrated in Figure 4–4. The results of empirical tests are shown in Appendix C for a laminate optimization problem.

![Figure 4–4: Even when the variance of the selected point distribution is greater than that of the candidate point distribution, the probabilistic acceptance scheme results in a reduction of the variance.](image-url)
**Target point approach**

This strategy is based on the hypothesis that the distribution of selected points is best represented in the auxiliary variable space. To this aim, a set of \( \lambda \) target points are sampled from \( p(\mathbf{V}) \), then the new population is obtained by retaining the candidate points that are closest to these target points\(^4\). Depending on the degree of confidence placed in the distributions \( p(\theta_1, \ldots, \theta_n) \) and \( p(\mathbf{V}) \), their relative weight in the creation of new points can be adjusted through the ratio \( \nu/\lambda \): when \( \nu/\lambda = 1 \), the lamination parameter distribution \( p(\mathbf{V}) \) plays no role in the search; when \( \nu/\lambda \to \infty \), the optimization is based primarily on information in the lamination parameter space.

In contrast to the previous approach, which tended to reduce the variance of the search distribution, the target point strategy allows the auxiliary variable distribution to correct the univariate distribution either by decreasing the variance, or by increasing it, as shown schematically in Figure 4–5. If the variability of the population generated from the univariate distributions is greater than that of the target distribution (assumed more accurate), target points will proportionately pick out more points in central regions than in the tails, thereby reducing the distribution variance (Figure (a)). Conversely, the variance can be increased by favoring peripheral regions if the points generated by the univariate distribution are too focused around the mean (Figure (b)).

---

\(^4\) Each target point is considered in turn, and the candidate which has the smallest Euclidean distance to it in the \( \mathbf{V} \)-space is copied in the new population and removed from the pool, thereby preventing multiple copies of a point from appearing in the new population.
Figure 4–5: Influence of the two-distribution scheme on the variance. The variability of the search distribution can be adjusted based on the auxiliary variable distribution (in a real situation, the means of the target and candidate distributions are different, but the same mechanisms operate).
4.3 The double-distribution optimization algorithm

4.3.1 General algorithm

The representation of the distribution of selected points provided by the target point procedure was adopted here because it has a clearer interpretation, and allows the relative contribution of each of the distributions to be adjusted depending on the confidence placed in each model. The resulting estimation of distribution algorithm, named Double-Distribution Optimization Algorithm (DDOA) is shown in Figure 4–6.

Figure 4–6: Flowchart of the double-distribution optimization algorithm.

The main steps of the algorithm are identical to the general EDA presented in Section 2.3, but the general form of \( p(\mathbf{x}) \) is substituted with the two-distribution statistical model consisting of the univariate distributions of the design variables.
\( p(x_k) \), and the auxiliary variable distribution \( p(V) \).

At the initialization and estimation steps, the two distributions are treated in parallel, and the generation of new points at the sampling step is performed according to the procedure presented in Section 4.2.2. The specific form of the distribution \( p(V) \), as well as the procedures implemented to estimate it and to sample points from it will be presented in the next section. As in UMDA, two alternative diversity preserving mechanisms were added to the algorithm to prevent premature convergence to a local optimum: a conventional mutation operator, applied with probability \( p_m \) per variable after the sampling step, and a lower bound \( \epsilon \) on the marginal probabilities (cf. Section 3.3), which ensures that each point of the design space has a non-zero probability of being visited at any time of the search.

4.4 Application to composite optimization

The algorithm is specialized to composite laminate optimization through the choice of proper auxiliary variables and applied to three problems presenting different features: an in-plane problem, where two auxiliary variables capture the overall response, an in-plane/out-of-plane problem, where four auxiliary variables must be used, and a problem where the auxiliary variables provide only partial information about the objective function.

4.4.1 DDOA for composites

The double-distribution optimization algorithm models variable interactions through the distribution of higher-order quantities that capture joint influences of several variables. In the case of composite laminates, the primary design variables are the fiber
angles $\theta_k, k = 1, \ldots, n$. Auxiliary variables appear naturally during the homogenization through the thickness of stiffness properties: the laminate stiffness is obtained as a product of material invariants and geometric contributions, called \textit{lamination parameters} (Tsai and Pagano, 1968). The only non-zero extensional lamination parameters $V^*_i$ and flexural lamination parameters $W^*_i$ of a balanced symmetric laminate $[\pm \theta_1, \pm \theta_2, \cdots, \pm \theta_n]_s$ are, respectively, the following\(^5\):

\[
V^*_{{1,3}} = \frac{2}{h} \int_0^{h/2} \{ \cos 2\theta, \cos 4\theta \} dz = \frac{1}{n} \sum_{k=1}^{n} \{ \cos 2\theta_k, \cos 4\theta_k \} \tag{4.2}
\]

and

\[
W^*_{{1,3}} = \frac{24}{h^3} \int_0^{h/2} \{ \cos 2\theta_k, \cos 4\theta_k \} z^2 dz = \frac{1}{n^3} \sum_{k=1}^{n} a_k \{ \cos 2\theta_k, \cos 4\theta_k \} , \tag{4.3}
\]

where $h$ designates the total laminate thickness and $a_k = (n - k + 1)^3 - (n - k)^3$ (cf. Appendix A–1 for more details). Note that Equations (4.2) and (4.3) define a feasible domain $\mathcal{L}$ in $(V^*_1, V^*_3, W^*_1, W^*_3)$, which is the set of the images of all possible laminates $[\pm \theta_1/\cdots/\pm \theta_n]_s$.

\(^5\) Here, the asterisk does not denote the optimum, but is a common way of distinguishing the normalized lamination parameters from their basic form, not used in this work.
The utility of lamination parameters for composite optimization has been demonstrated by several researchers (e.g. Miki, 1986; Todoroki and Haftka, 1998). Performing the optimization in the lamination parameter space presents several advantages: in some cases, it forces convexity of the objective function (Foldager et al., 1998), it reduces dimensionality, it renders the problem amenable to continuous optimization methods. However, the problem of finding the laminate that corresponds to the optimum lamination parameters \( V_{\text{opt}} = (V_{1,\text{opt}}, V_{3,\text{opt}}, W_{1,\text{opt}}, W_{3,\text{opt}}) \) is non-trivial: it constitutes an optimization problem in itself, and can have one solution, or several solutions, or even none.

The algorithm for laminate optimization uses a simple univariate model to represent the angle distribution \( p(\theta_1, \ldots, \theta_n) \), and adds variable dependencies by biasing the search based on the distribution of selected points in the lamination parameter space. Since the fiber angles \( \theta_k \) are discrete, the lamination parameters can also assume only a finite number of discrete values. However, the number of values that they can take increases rapidly with the problem dimension \( n \). Furthermore, these values fill up the whole lamination parameter space when \( n \) becomes large. For these reasons, their distribution is best described by a continuous model.

We will apply the general DDOA algorithm presented in Section 4.3.1, substituting the primary variable distributions \( p(x_k), k = 1, \ldots, n \) with the fiber angle distributions \( p(\theta_k) \), and the lamination parameter distribution \( p(V) \) (\( V \) is the set of lamination parameters considered and depends on the problem at hand: \( V = (V_{1,\text{opt}}, V_{3,\text{opt}}) \) for in-plane problems, \( V = (W_{1,\text{opt}}, W_{3,\text{opt}}) \) for out-of-plane problems, and \( V = (V_{1,\text{opt}}, V_{3,\text{opt}}, W_{1,\text{opt}}, W_{3,\text{opt}}) \) for problems involving both the extensional and flexural properties of the laminate).
as auxiliary variable distribution.

4.4.2 Issues associated with the representation of the auxiliary variable distribution

Two aspects of the algorithm deserve special consideration: the representation and the estimation of the probability distribution $p(V)$ and the target point creation procedure.

**Representation of $p(V)$**

Lamination parameters are inherently continuous, therefore a continuous probability density function is used to represent $p(V)$. Most of the works on continuous EDAs to date use univariate normal distributions to represent the distribution of selected points (Sebag and Ducoulombier, 1998; Gallagher et al., 1999). The disadvantages of that model is that it does not model dependencies between variables, and assumes a unimodal symmetric distribution. Bosman and Thierens (2000) proposed to use a non-parametric kernel density estimation (KDE) method to achieve a more accurate approximation. For this method to provide an accurate estimate of distributions, a good coverage of the space is required. Since the number of lamination parameters that we consider is small (two or four) and independent of the problem dimension $n$, good space coverage can be achieved with reasonable sample sizes, so that KDE is appropriate for modeling the distribution $p(V)$.

In the kernel density estimation method, a kernel $K(u)$ is placed at each sample point. The distribution $p(V)$ is obtained as

$$p(V) = \frac{1}{\mu} \sum_{i=1}^{\mu} K(V - V_i).$$  \hspace{1cm} (4.4)
In this work, we used Gaussian kernels:

\[ K(u) = \frac{1}{(2\pi)^{d/2}\sigma^d} \exp \left( -\frac{u^T u}{\sigma^2} \right) \]  (4.5)

where \( d \) is the dimension of \( u \) and the variance \( \sigma^2 \) is the bandwidth that needs to be adjusted: a small value of \( \sigma \) increases the resolution but also increases the variance of the estimate when few data points are available. Several methods for adjusting the value of \( \sigma \) exist, such as trial-and-error, maximum likelihood, or adaptive strategy. In this work, a maximum likelihood method was used (see Appendix B for a description of the method, and details about the procedure).

**Target point creation procedure**

The use of unbounded-support density functions, such as normal kernels, for the representation of \( p(V) \) complicates the sampling of points in the bounded lamination parameter space, because infeasible target points (points that do not lie in the feasible domain \( L \), see Appendix A) can be generated. Two strategies were considered:

- one can either force the target points to lie in the feasible domain by sampling new points until feasible points are obtained,
- or allow infeasible target points to be created, considering that all the points of the new generation will be feasible by construction (the points are previously created in the \( \theta \)-space, cf. Figure 4–6).

The former approach has two disadvantages: first, the exact boundaries of the feasible domain are only known for a few simple combinations of lamination parameters ((\( V_1^* \), \( V_3^* \)) and (\( W_1^* \), \( W_3^* \))). For general combinations, approximate relations only are
available (Diaconu et al., 2002a). Second, tests showed that the performance of algorithms based on the rejection of infeasible target points deteriorates when the optima lie close to the boundary (cf. Appendix B). Consequently, the latter scheme only was used in this work.

### 4.4.3 2D didactic example

The validity of the approach is now demonstrated on the 2D fitness landscape presented in Section 4.1. We saw that the probability distribution yielded by a univariate model approximated the distribution of selected points (reproduced in Figure (a)) very poorly. Figure (b) shows the distribution obtained by the double-distribution procedure with linear ranking selection, $\lambda = 30$, $\mu = 30$, $\nu = 300$, averaged over 100 independent runs. Clearly, using the information about the distribution of auxiliary variables dramatically improves the accuracy of the search distribution: the high-probability area around (55,55) caused by univariate sampling is still observable, but

![Figure 4–7: Effect of the two-distribution sampling procedure. The auxiliary-variable-based acceptance scheme introduces variable dependencies (compared to Figure 4–1).](image-url)
its relative weight in the distribution is considerably reduced. On the contrary, the shape of the valley, which was hardly visible in the univariate distribution appears clearly in the distribution produced by DDOA.

### 4.4.4 Extensional problem

The first optimization problem was maximizing the transverse in-plane stiffness coefficient $A_{22}$ subject to a constraint on the effective Poisson’s ratio $\nu_{\text{eff}}$:

$$\text{maximize } A_{22} = h(U_1 - U_2 V_1^* + U_3 V_3^*)$$  \hspace{1cm} (4.6)

such that $\nu_l \leq \nu_{\text{eff}} \leq \nu_u$

The effective Poisson’s ratio $\nu_{\text{eff}}$ is a function of $V_1^*$ and $V_3^*$:

$$\nu_{\text{eff}} = \frac{A_{12}}{A_{22}} = \frac{U_4 - U_3 V_3^*}{1 - U_2 V_1^* + U_3 V_3^*},$$  \hspace{1cm} (4.7)

and $U_1$, $U_2$, $U_3$, $U_4$ are material invariants, obtained from the material properties shown in Table 3–3 p. 68, and the total laminate thickness was fixed at $h = 0.2$ in. The lower and upper limits of the Poisson’s ratio were $\nu_l = 0.48$ and $\nu_u = 0.52$. The inequality constraints were enforced by a penalty approach:

$$F(\theta) = \begin{cases} A_{22} & \text{if } g(\theta) \geq 0 \\ A_{22} + p g(\theta) & \text{if } g(\theta) < 0 \end{cases}$$  \hspace{1cm} (4.8)

---

6 In practice, the ply thickness is fixed, and the total thickness depends on the number of plies. However, we used a fixed total thickness in this work to observe the effect of dimensionality for a given objective function (the orders of magnitude of $A_{22}$ and $\nu_{\text{eff}}$ remain the same for a fixed $h$).
where

$$g(\theta) = \min \left( \frac{\nu_{\text{eff}}}{\nu_1} - 1, 1 - \frac{\nu_{\text{eff}}}{\nu_u} \right).$$

(4.9)

The value of the penalty parameter $p$ was set to $5.0 \times 10^6$ so as to guarantee a feasible solution. The fiber orientations were chosen from $\{0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ\}$. The fitness landscape for $n = 2$ is plotted in Figure 4–8: the function presents two nearly circular high-fitness regions, corresponding to the feasible domain. The maximum is located in the outer region, while the inner high-fitness region is only locally maximum. The curved shape of these areas will give rise to variable dependencies in selected points, hence making univariate statistical model inaccurate (in these models maximum probability areas are always aligned with the axes, cf. Section 4.1).

Figure 4–8: Fitness landscape of the penalized $A_{22}$ for $n = 2$.

This problem is an ideal case for the DDOA algorithm because the objective
function can be expressed entirely in terms of $V_1^*$ and $V_3^*$, so that the distribution of promising points can be characterized completely in the $(V_1^*, V_3^*)$-plane. In this problem, variable interactions originate from two factors. First, the constraint creates a narrow “ridge” in the design space and gives rise to variable dependencies similar to those observed in Figure 4–1. Second, since the order of the plies is irrelevant, several configurations of the optimal laminate may be present in the set of selected points, hence creating additional variable dependencies\(^7\).

Two cases were tested: $n = 6$ and $n = 12$. In the former case, the optimum stacking sequence was $[\pm 45/ \pm 67.5]_s$ (or any permutation of the same angles), with a transverse stiffness $A_{22} = 1.897 \times 10^6$ lb/in, and an effective Poisson’s ratio $\nu_{\text{eff}} = 0.519$. In the case $n = 12$, the optimum stacking sequence was $[\pm 45_9/ \pm 67.5/90_4]_s$ (or any permutation of the same angles), with a transverse stiffness $A_{22} = 1.922 \times 10^6$ lb/in, and an effective Poisson’s ratio\(^8\) $\nu_{\text{eff}} = 0.489$.

The DDOA algorithm was compared to two evolutionary algorithms: a standard genetic algorithm (population size of 30, two-point crossover with probability $p_c = 1.0$, linear ranking parent selection) and the univariate marginal distribution algorithm (Mühlenbein and Mahnig, 2000) with linear ranking selection, population

---

\(^7\) For instance, if the optimum laminate is $[\pm \theta_1^*/ \pm \theta_2^*]_s = [\pm 30/ \pm 45]_s$, $[\pm 45/ \pm 30]_s$ is also a solution. If we know that $\theta_1 = 30$, the probability of $\theta_2 = 45$ being optimum (denoted by $p(\theta_2 = 45|\theta_1 = 30)$) is 1 and the probability of $\theta_2 = 30$ being optimum is 0: clearly, $p(\theta_2|\theta_1)$ is a function of $\theta_1$, which is the definition of variable dependence.

\(^8\) The constraint is not strictly active because of the discreteness of the variables.
size $\lambda = 30$ and selected population size $\mu = 30$ (duplicates were allowed in the selected population). These settings were also used for DDOA, and a candidate pool size $\nu = 150$ was chosen. In the three algorithms, mutation consisted in changing the value of a variable to one of the two nearest values, and a fixed mutation rate $p_m = 0.02$ per variable was used. Natural coding was employed for the three algorithms (no transformation was applied to the variables before using them in the algorithms). No elitist strategy was implemented, to isolate the effect of the probabilistic model\textsuperscript{9}. The bandwidth was chosen by a maximum likelihood method, which provides an upper bound for the optimal bandwidth by determining the value of $\sigma$ that best approximates the distribution of selected points at the first generation of the algorithm (see details in Appendix B). Using this approach we obtained $\sigma = 0.2$ for $n = 6$ and $\sigma = 0.15$ for $n = 12$.

The performance of the optimization was assessed by estimating the mean objective function evaluation of the best point at each generation, and the reliability $R$, estimated over 50 independent runs.

Figure 4–9 compares the three algorithms for $n = 6$. DDOA is particularly efficient in beginning of the search: as early as the first generation, its mean objective function is significantly higher than that of the two other algorithms. This emphasizes the importance of initialization: by imposing a uniform initial distribution in the lamination parameter space, DDOA reduces the bias toward laminates made of

\textsuperscript{9} However the best solution found at each generation is recorded, so that even though it does not influence the search, it is not lost for the user.
multiple angles (when the ply order is irrelevant, laminates such as $[0_2/\pm 45/90_2]$ can appear under six permutations, whereas $[0_6]$ has only one configuration, hence it is less likely to appear by uniform sampling in the angle space). DDOA’s higher mean objective function translates into a higher reliability, indicating that the algorithm not only converges to high objective function regions faster than its competitors, but yields the true optimum more consistently.

![Figure 4-9: Mean maximum fitness and reliability for the constrained max $A_{22}$ problem, $n = 6$ (50 runs).](image)

The performance improvement can be attributed to the more efficient initialization, and to a more accurate estimation of the distribution of selected points achieved by incorporating variable dependencies via the lamination parameters. Working in the $(V^*_1, V^*_3)$-plane was especially beneficial for this problem where the ply order is irrelevant because all the configurations of a given laminate get mapped into a single point in the $(V^*_1, V^*_3)$-plane, so that the distribution becomes very simple in that space, while a univariate model in the $\theta$-space describes the distribution only imperfectly.
In the case $n = 12$, the convergence criterion was relaxed by using a “practical” optimum, or acceptable solution, defined as 99% of the maximum objective function, instead of the true optimum. The performance of the three algorithms is compared in Figure 4–10. The gain achieved by incorporating information about the lamination parameter distribution is very significant for this problem. DDOA’s superiority is particularly dramatic in the initial phase of the search, from the start of the search to 400 evaluations, where convergence toward high objective function regions benefits from the information about promising designs in the lamination parameter space.

The higher comparative performance of DDOA for larger $n$ can be explained by the fact that DDOA performs part of the data analysis in the $(V^*_1, V^*_3)$-plane, whose dimension is invariant, consequently, using the lamination parameters to guide the search partially neutralizes the effect of dimensionality on the reliability.
4.4.5 Extensional-flexural problem

The second problem was minimizing the absolute value of the longitudinal coefficient of thermal expansion (CTE) \( \bar{\alpha}_x \) of a simply supported rectangular graphite-epoxy plate subject to a lower bound on the first natural frequency \( f_1 \). The problem was formulated as follows:

\[
\text{minimize } |\bar{\alpha}_x| \\
\text{such that } f_1 \geq f_{\text{min}}
\] (4.10)

where the longitudinal CTE is a function of \( V_1^* \) and \( V_3^* \) (Gürdal et al., 1998, p. 154, see Appendix A for details):

\[
\bar{\alpha}_x = \frac{V_1^*(K_2 U_1 - K_1 U_2 + K_2 U_4) - V_1^{*2}K_2 U_2 + 2K_1 U_3 V_3^* + K_1(U_1 - U_4)}{2(2V_3^* U_3(U_4 + U_1) - V_1^{*2}U_2^2 + (U_1^2 - U_4^2))} \] (4.11)

where

\[
K_1 = (U_1 + U_4)(\alpha_1 + \alpha_2) + U_2(\alpha_1 - \alpha_2) \\
K_2 = U_2(\alpha_1 + \alpha_2) + (U_1 + 2U_3 - U_4)(\alpha_1 - \alpha_2) \\
K_3 = U_2(\alpha_1 + \alpha_2) + 2(U_3 + U_5)(\alpha_1 - \alpha_2) .
\]

The first natural frequency is a function of \( W_1^* \) and \( W_3^* \) through \( D_{11}, D_{12}, D_{22}, \) and \( D_{66} \) (cf. Appendix A for details):

\[
f_1 = \frac{\pi^2}{\sqrt{\rho h}} \sqrt{\frac{1}{a^4} D_{11} + \frac{2}{a^2b^2} (D_{12} + 2D_{66}) + \frac{1}{b^4} D_{22}} . \] (4.12)

Note that the objective function can be expressed in terms of the lamination parameters only, as in the previous problem.

The minimization problem was recast into a maximization problem by defining
the objective function as follows:

\[
F(\theta) = \begin{cases} 
\alpha_2 - |\alpha_x| & \text{if } g(\theta) \geq 0 \\
\alpha_2 - |\alpha_x| + p g(\theta) & \text{if } g(\theta) < 0
\end{cases}
\] (4.13)

where

\[
g(\theta) = \frac{f_1}{f_{\text{min}}} - 1 .
\] (4.14)

For this problem, the material properties shown in Table 3–3 p.68 were used. The coefficients of thermal expansion in the longitudinal and transverse directions were \( \alpha_1 = 0.02 \times 10^{-6} \text{ K}^{-1} \) and \( \alpha_2 = 22.50 \times 10^{-6} \text{ K}^{-1} \), respectively, and the dimensions of the plate were \( a = 30 \text{ in} \) and \( b = 15 \text{ in} \). The minimum frequency was \( f_{\text{min}} = 150 \text{ Hz} \). The fitness landscape for \( n = 2 \) is shown in Figure 4–11.

![Figure 4–11: Fitness landscape of the penalized CTE problem for \( n = 2 \).](image)

Two cases were considered in the optimization: \( n = 6 \) and \( n = 12 \). In the
former case, three feasible laminates yielded the minimum longitudinal CTE of $\alpha_x = 8.54 \times 10^{-7}$ K$^{-1}$: $[90_2/\pm 67.5/0_8]_s$, which had a first natural frequency of $f_1 = 156.25$ Hz, $[90_2/0_2/\pm 67.5/0_6]_s$, with $f_1 = 152.53$ Hz, and $[90_2/0_8/\pm 67.5]_s$, with $f_1 = 151.36$ Hz (these values are to be compared to the maximum first natural frequency of the plate $f_1 = 183.71$ Hz for $[90_{12}]_s$ and the minimum CTE for the unconstrained problem $\alpha_x = \alpha_1 = 0.2 \times 10^{-7}$ K$^{-1}$ for $[0_{12}]_s$). In the case $n = 12$, the optimum laminate was $[90_4/\pm 67.5/0_6/\pm 22.5_6]$, with $\alpha_x = 5.95 \times 10^{-7}$ K$^{-1}$ and $f_1 = 150.13$ Hz (the maximum frequency is $f_1 = 183.71$ Hz for $[90_{24}]_s$, and the minimum CTE is $\alpha_x = -1.82 \times 10^{-9}$ K$^{-1}$ for $[90_2/ \pm 45_2/ \pm 22.5_3/0_{12}]_s$).

The penalty parameter was set to $p = 1.0 \times 10^{-4}$. The allowable fiber orientations, genetic operators, and other parameters were the same as for the in-plane problem. The bandwidth found by the maximum likelihood method were $\sigma = 0.2$ for $n = 6$ and $\sigma = 0.15$ for $n = 12$. The performance of the standard GA, UMDA and DDOA is shown in Figures 4–12 and 4–13.

![Figure 4–12: Mean maximum fitness and reliability for the constrained min CTE problem, n = 6.](image-url)
Figure 4–13: Mean maximum fitness and reliability for the constrained min CTE problem, $n = 12$ (50 runs).

For this problem, a clear difference in the behavior of the algorithms can be observed between the two cases $n = 6$ and $n = 12$. In the former case, the optimum laminates do not exploit ply interactions: the best laminates are just obtained by using as many 0-degree plies as possible, as they possess the lowest longitudinal coefficient of thermal expansion, and complementing by two stacks of plies oriented at large angles in outer layers to satisfy the frequency constraint (we shall call these laminates solutions of Type I). In the case $n = 12$, the optimum is more complex because its response results from the combined action of all the plies: the $\pm 22.5$ stacks alone would yield a negative CTE of $\alpha_x = -3.95 \times 10^{-6}$ K$^{-1}$. When used in association with $0_2$, $\pm 45$, and $90_2$ stacks, which have positive CTEs, they produce very low CTE laminate. Because they are made of moderate to large angles, such laminates easily satisfy the frequency constraint (we shall refer to these laminates as solutions of Type II). The $\pm 22.5$ stacks are not particularly good individually: they are only useful insofar as they compensate for larger-CTE plies, giving rise to strong
ply interactions.

The performance of the algorithms reflects this difference in the structure of the solutions: in the case $n = 6$, no significant difference between the three algorithms can be seen. In contrast, in the case $n = 12$, DDOA’s superiority appears clearly, both in the average best objective function and in the reliability. While GA and DDOA tend to converge to sub-optimal Type I solutions, DDOA reliably finds Type II solutions by identifying promising angle combinations through the lamination parameters.

### 4.4.6 Strength problem

The third problem was maximizing the load factor $\lambda_s$, using the first-ply-failure criterion based on the maximum strain, for a glass-epoxy laminate subjected to the in-plane loading $N_x = -1000 \times 10^3$ N/m, $N_y = 200 \times 10^3$ N/m, $N_{xy} = 400 \times 10^3$ N/m:

$$\text{maximize } \lambda_s = \frac{n}{k=1} \left\{ \min \left[ \max \left( \frac{\epsilon_1^t}{\epsilon_1(k)}, -\frac{\epsilon_1^c}{\epsilon_1(k)} \right), \max \left( \frac{\epsilon_2^c}{\epsilon_2(k)}, -\frac{\epsilon_2^t}{\epsilon_2(k)} \right) \right] \right\}$$

where the load factor $\lambda_s$ is the coefficient by which the load has to be multiplied for the structure to fail. The material properties used for this problem are shown in Table 3–4 p. 74. The total thickness of the laminate was $h = 2$ cm. The particularity of this problem is that the lamination parameters provide only partial information about the objective function:

$$\lambda_s = \lambda_s(V_1^*, V_3^*, \theta_1, \ldots, \theta_n) .$$

We applied the three algorithms to the case $n = 12$. The optimal laminate for
that problem was $[0_{14}/ \pm 67.5\%]_s$ (or any permutation of the same angles), which yielded a load factor $\lambda_s = 4.74$. The parameter settings were kept unchanged from the previous problems. A bandwidth of $\sigma = 0.15$ was used for DDOA. The mean best objective function and the reliability (for a practical optimum of 95% of the optimum) are presented in Figure 4–14.

The algorithms’ performance is directly influenced by the amount of variable interactions used to guide the search: UMDA, which treats all the variables independently quickly converges to an average best objective function of $\lambda_s \approx 3.6$ in 400 evaluations and stagnates in this sub-optimal region afterward; two-point-crossover-based GA preserves some linkage between genes and reaches a higher average best objective function of $\lambda_s \approx 3.85$. DDOA, which explicitly handles laminate-level variable dependencies through the lamination parameters, achieves a substantially higher average objective function of $\lambda_s \approx 4.1$. DDOA’s advantage is even more remarkable when we look at the reliability: the algorithm’s probability of finding a solution within 5% of the global optimum reaches 62% for 1,500 function evaluations, while GA and UMDA stagnate at 12% and 14% respectively.

In contrast to the two previous problems, where variable dependencies originated from constraint satisfaction considerations, the variables of this problem are related with strong dependencies because changing the orientation of a particular ply causes a load redistribution, which directly affects the optimal orientation of other plies. By allowing groups of variables to be treated together, GA’s two-point crossover changes the value of some variables, while keeping the others constant, it searches the best orientation of these plies without causing significant load redistribution. DDOA provides
a more explicit mechanism for separating overall response and local ply adjustment by allowing multiple laminates that have similar stiffness properties (hence similar strains) to be generated. UMDA, which extrapolates promising regions based on insufficient ply-level information, is not able to reliably produce high-strength laminates, hence its poor average best function evaluation, and the absence of progress.

Figure 4–14: Mean best fitness and reliability for the strength problem, $n = 12$ (50 runs)

4.4.7 A comparison of diversity in UMDA and DDOA

The diversity of the populations generated during the search is one of the important factors that determine the efficiency: excessive entropy in the search distribution hampers the convergence to high-fitness regions, while insufficient diversity leads to a localized search that fails to extract information about globally optimal regions (“premature convergence” in EAs).

In addition, in some situations, population diversity can be a desirable attribute in itself. For instance, the user may be interested in having the option to choose
between several alternative solutions to a problem (sometimes it gives him a chance to account for factors that were left out in the problem definition). Likewise, in constraint satisfaction problems, where one is concerned with the identification of whole regions of the space, maintaining a high level of diversity is essential.

The issue of the preservation of diversity is an essential but often ignored aspect of estimation of distribution algorithms. Indeed, when one applies an EDA as presented in Section 2 to practical problems, one immediately faces the problem of “premature convergence” well known in genetic algorithms: after a few iterations, the probability of whole regions of the search space vanishes, consequently these areas are excluded from the search domain for the subsequent iterations. If the remaining regions contain the optimum, this phenomenon is beneficial, however, this is not generally the case. If, at some point of the optimization, the probability of the optimum vanishes, a basic EDA will never be able to find it. Two factors contribute to the loss of optimum from the distribution:

1. fitness-based selection aims at identifying good sub-solutions (variables or groups of variables) that make up the optimum. The frequency of these sub-solutions depends on their mean fitness in the population. Due to the influence of the other variables, this mean fitness is a random variable. Depending on its variability, the discrimination of sub-solutions may be difficult, and there is a chance that no instance of the optimal values of the variables considered will be present in the selected points, which immediately leads to a zero probability for the optimum;

2. the effect of noisy discrimination is compounded by the fact that even in the
absence of fitness-based selection, iteratively sampling from a discrete distribution and estimating the updated distribution from that sample results in the degeneracy of the distribution to a single point. Even if optimal values of all the variables are represented in the selected population, there is a finite probability that they will be lost in the sampling process.

The population diversity can be monitored on two levels\(^\text{10}\): at the variable level, the diversity in the design variable space is an indicator of the portion of the space covered by the search distribution; at the fitness level, the diversity can be used as an indirect measure of the entropy of the search distribution, although a low fitness diversity does not imply a low entropy of the search distribution (if many different designs get mapped into a single point in the fitness space, a large population diversity in the design variable domain can produce a low diversity in the criterion space). While the diversity in the design variable domain is a more meaningful indicator of the strength of exploration, it is more difficult to define, and computationally more expensive to evaluate than the diversity in the criterion space.

Diversity in the criterion space \(D_{\text{crit}}\) was directly measured by the standard deviation of the fitness in the population. Diversity in the variable space was measured by the average pairwise “city-block” distance in the population:

\[
D_{\text{var}} = \frac{2}{\lambda(\lambda - 1)} \sum_{i=1}^{\lambda-1} \sum_{j=i+1}^{\lambda} d_{ij},
\]

\(^{10}\) In evolutionary computation, these are often referred to as genotypical and phenotypical diversity.
where

$$d_{ij} = \frac{1}{n} \sum_{k=1}^{n} |x_k^{(i)} - x_k^{(j)}| . \quad (4.17)$$

The evolution of $D_{\text{crit}}$ and $D_{\text{var}}$ was monitored during the maximization of the laminate strength of Section 4.4.6. Figures 4–15 and 4–16 show the criterion domain and variable domain diversities, respectively (for the whole population and the top third of the population, which approximately constitutes the basis for the construction of the statistical model). While the population standard deviation of UMDA drops very rapidly, DDOA maintains an almost constant level of diversity throughout the search. Maintaining a high level of fitness diversity is neither good nor bad a priori, as random exploration exhibits such variability, but often wastes many function evaluations on poor regions: this information has to be examined jointly with the mean fitness, shown earlier. For this problem, DDOA manages to keep a sustained level of variability, while displaying a substantially higher mean best fitness than
Figure 4–16: Diversity in the variable domain for UMDA and DDOA.

UMDA. This indicates that DDOA not only maintains variability, but it focuses it preferentially on fitter regions.

Variable domain diversity provides a different insight into the search mechanisms at work in UMDA and DDOA: clearly, the diversity of UMDA quickly vanishes, unlike that of DDOA, which reaches a plateau, after an initial drop in the first iterations. The lack of diversity (especially in the upper part of the population) makes further exploration difficult, as the statistical model will only generate many duplicates of a single solution. In contrast, DDOA does not suffer from this shortcoming: it guarantees a minimum level of diversity, thus enabling further progress.

A more detailed picture of the search distributions generated by UMDA and DDOA is provided in Figure 4–17, which shows the evolution of the fitness distributions of the best solution found at each iteration of the optimization. In the case of UMDA, Figure (a) clearly shows a concentration of points around $\lambda_s \approx 3.1$, which do
not correspond to any maximum, and two high density regions around the local maxima $L_1 = [0_{10}/ \pm 45_{4}/ \pm 67.5/90_{4}]_s$ ($\lambda_s = 4.32$) and $L_2 = [0_{8}/ \pm 45_{5}/90_{6}]_s$ ($\lambda_s = 3.89$). The optimum $\lambda_s = 4.74$ does not appear in the distribution. In contrast with DDOA, the weight of the “good random solutions” ($\lambda_s \approx 3.1$) that were present in UMDA quickly vanishes. Instead, the algorithm focuses on the high-fitness local optimum $L_1$ and the global optimum.

![Image](a) UMDA  
(b) DDOA

Figure 4–17: Distribution of the fitness of the best solution at each iteration for UMDA and DDOA.

This diversity preserving effect is the direct manifestation of the diversity compensating effect mentioned in Section 4.4.2: the double-distribution algorithm and its target point approach provide a convenient way of controlling the variability of the search distribution via the value of the bandwidth $\sigma$ and of the candidate pool size $\nu$.

**Diversity injection mechanism**

In theoretical EDAs, no diversity preservation mechanism is implemented, because infinite populations are assumed, so every point (in particular the optimum)
has a non-zero probability of being visited at any time of the search. However, practical implementations employ finite populations, and it becomes essential to guard against loss of diversity.

A second reason for adding diversity injection mechanisms to EDAs is to compensate for an inaccurate statistical model: indeed, the understanding of diversity preservation mechanisms as mere means to allow a pure EDA search procedure to converge when distributions are estimated from finite samples assumes that the EDA converges in the first place. This supposes that an accurate statistical model is used, and that judicious exploitation of the information summarized in the model leads to the optimum. Often, a simplified statistical model is used, and exploitation only cannot yield the optimum. In those situations, an exploratory component has to be added in the form of a perturbation: diversity injection mechanisms play such a role.

Finally, even when an accurate model is used, a combination of exploitative and exploratory search components may turn out to be the most effective strategy.

In this work, we provided two different diversity preserving mechanisms (cf. Section 3.3):

**Mutation:** a perturbation is applied with probability $p_m$ to each variable $\theta_k$ of each of the $\lambda$ created points. The perturbation consists in changing the value of $\theta_k$ to one of the neighboring values with equal probability (e.g. $45^\circ$ can be changed to $22.5^\circ$ or $67.5^\circ$).

**Bounds on the search distribution:** that does not allow the marginal probabilities $p(\theta_k)$ threshold $\epsilon$.

Mutation applies to all the variables and is independent of the state of the
search distribution. The bound on marginal distributions directly affects the search
distribution, but its action is more localized, as it only applies to variables of low
entropy.

The influence of the two diversity preservation operators was investigated on the
strength maximization problem, for \( n = 12 \). The reliability (for a practical optimum
of 95% of the optimum) of UMDA and DDOA for values of \( p_m \) ranging between 0
and 0.05 (without limitation on the probability distribution) is shown in Figure 4–18.

When no mutation is used (\( p_m = 0 \)), UMDA never finds the optimum because

![Figure 4–18: Effect of mutation for UMDA (a) and DDOA (b).](image)

the distribution quickly degenerates, which prevents any further progress. DDOA
reaches a slightly higher reliability of 12%. When the mutation rate is increased, the
effectiveness of both algorithms improves, though UMDA’s reliability never exceeds
10% for \( p_m = 0.04 \). With the same mutation rate, DDOA reaches a reliability of
74%. Further increasing the amount of random perturbation causes both algorithm’s
performance to deteriorate.

The influence of “repairing” degenerate probability distributions by imposing
lower bounds on the $p(\theta_k)$’s is shown in Figure 4–19 for UMDA (a) and DDOA (b), which presents the algorithms’ reliability for seven values of $\epsilon$, ranging from 0 to 0.08 (which represents $2/5$ of the uniform distribution $p(\theta_k = c_l) = 1/5 = 0.2$). This

![Figure 4–19: Effect of bounds on the probability distributions $p(\theta_k)$ for UMDA (a) and DDOA (b).](image)

diversity preservation mechanisms is even more beneficial to the search performance than mutation: UMDA’s reliability reaches 60% for $\epsilon = 0.04$, and DDOA’s reliability at the end of the optimization increases to 80% for the same value of $\epsilon$. As in the case of mutation, higher values of $\epsilon$ cause the performance of both algorithms to drop. The advantage of bounds on the probability over mutation can be explained by the fact that mutation is a costly way of reintroducing lost variable values. Recovering a lost value through mutation may take a large number of iterations if $p_m$ is not sufficiently high, or cause large perturbations if $p_m$ is large. Repairing the distribution guarantees that all values have a chance of being generated, while inflicting only marginal perturbation to the convergence process.

While both mutation and the lower bound on marginal probabilities have an
obvious positive influence on both algorithms, DDOA clearly benefits more than UMDA from this injection of randomness: even with high mutation rates, UMDA’s reliability remains poor, and when limits on the probabilities are imposed, DDOA performs better than UMDA on average, and its reliability is less sensitive to the value of $\epsilon$ than that of UMDA. This can be explained by observing that the method used by DDOA to generate new points allows it to either compensate for a lack of diversity if the population becomes too uniform, or reduce its diversity when heavy perturbations have been introduced (cf. Section 4.2.2). In the Gaussian kernel model of the auxiliary variable distribution, the bandwidth $\sigma$ determines the minimum amount of diversity in the lamination parameter space, hence the intrinsic exploration in the algorithm (without additional injection of diversity).

4.5 Performance with optimized parameters

So far, the emphasis was placed on studying the influence of a small number of parameters. In this section, a parameter study is performed for each of the three algorithms (GA, UMDA, and DDOA), and the optimized algorithms are compared.

4.5.1 Parameter study and best setting

The test problem is the strength maximization problem presented in Section 4.4.6, with $n = 12$. The characteristics of the algorithms are presented in Table 4–1. For each configuration, the algorithms were run 50 times, and the reliability was computed at each iteration.

For each of the three algorithms, the best setting was determined based on the reliability at the end of the search, set to 3,000 function evaluations. The overall best
Table 4–1: Characteristics of the three algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Selection</th>
<th>Child population size $\lambda$</th>
<th>Parent population size $\mu$</th>
<th>Mutation rate $p_m$</th>
<th>Bandwidth $\sigma$</th>
<th>Candidate pool size $\nu$</th>
<th>Probability limit $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>Linear ranking</td>
<td>${12, 15, 20, 40, 60, 80, 100}$</td>
<td>$\lambda = \mu$</td>
<td>${0.005, 0.01, 0.02, 0.03}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UMDA</td>
<td>Linear ranking</td>
<td>${12, 15, 20, 40, 60, 80, 100}$</td>
<td>$\lambda = \mu$</td>
<td>${0.005, 0.01, 0.02, 0.03}$</td>
<td>${0.05, 0.1, 0.2, 0.3}$</td>
<td>${2\lambda, 5\lambda, 20\lambda}$</td>
<td>${0.01, 0.02, 0.04, 0.06}$</td>
</tr>
<tr>
<td>DDOA</td>
<td>Linear ranking</td>
<td>${12, 15, 20, 40, 60, 80, 100}$</td>
<td>$\lambda = \mu$</td>
<td>${0.005, 0.01, 0.02, 0.03}$</td>
<td>${0.05, 0.1, 0.2, 0.3}$</td>
<td>${2\lambda, 5\lambda, 20\lambda}$</td>
<td>${0.01, 0.02, 0.04, 0.06}$</td>
</tr>
</tbody>
</table>

configurations were obtained for that criterion were the following:

**GA:** $\lambda = \mu = 80, p_m = 0.02$;

**UMDA:** $\lambda = \mu = 40, \epsilon = 0.06$;

**DDOA:** $\lambda = \mu = 40, \sigma = 0.1, \nu = 200, \epsilon = 0.06$.

Note that GA requires a substantially larger population than the two EDAs. In the case of DDOA, the best performance is not achieved for the larger value of the pool size, but for $\nu = 5\lambda$: this can be explained by the fact that the auxiliary variables capture only part of the overall response, hence their distribution is not sufficient to guide the search.

The reliability of the optimized algorithms is shown in Figure 4–20. The same hierarchy as in the previous section (where “reasonable” values of the parameters were used, without extensive parameter study) is observed: DDOA remains the most efficient algorithm, although its superiority is significantly reduced (to be compared to Figure 4–14). GA benefits from a larger population, while placing bounds on
UMDA’s marginal distributions drastically improves its behavior, as seen in Section 4.4.7.

Another interesting result of the parameter study concerns the sensitivity of the algorithms’ performance to the value of the parameters: in a realistic setting, one does not have the luxury to perform many tests in order to obtain the best algorithm parameter values. It is therefore important to examine how changes in the parameters affect the performance. Figure 4–21 shows the reliability of the three algorithms for seven different populations sizes, the other parameters being set to their best found value. The graphs reveal a striking difference in the algorithms’ sensitivity to changes in the population size: while the two EDAs display a relative stability to the choice of \( \lambda \), the GA’s reliability is strongly dependent on its value. In particular, lower values of \( \lambda \) yield very poor reliability: unlike UMDA and DDOA, which rely on an explicitly mechanism that prevents the distributions from converging (the bounds on marginal
distributions), GA must resort to the more disruptive mutation operator, or use large populations.

To summarize, even though GA and UMDA compare well with DDOA at their best settings, departure from these optimal parameter values (population size for GA, diversity preserving mechanism for UMDA) may cause a severe drop in the performance, while DDOA consistently provides a reliable solution.
4.6 Generalization of DDOA to continuous optimization problem

The Double-Distribution strategy for incorporating statistical variable dependencies into the model of promising regions is general and can be applied whenever auxiliary variables are available. The goal of this section is to demonstrate the generality of the method by applying it to continuous problems.

4.6.1 Problem description

We consider the problem of maximizing the distance to a point $x_0$:

$$
\text{maximize } d = \sqrt{\sum_{i=1}^{n} (x_i - x_{0i})^2}
$$

such that

$$
g_1(x) = 1 - \frac{r^2}{R^2} \leq 0
$$

and

$$
g_2(x) = \frac{r^2}{(R + \delta)^2} - 1 \leq 0
$$

with $x \in [0, 10]^n$, $R = 1$, and $r^2 = \sum_{i=1}^{n} x_i^2$. The reference point $x_0$ was defined in such a way that one of the vertices of the feasible space $x_v$ was the unique global optimum:

$$
x_0 = x_m + \epsilon (x_m - x_v)
$$

where $x_m$ is the middle of the diagonal of the unit cube ($x_m = \frac{1}{2\sqrt{n}}(e_1 + e_2 + \cdots + e_n)$).

In our problem, we chose $x_v = (1 + \delta) e_1$, $\delta = 0.01$, $\epsilon = R/2$.

The constraint was enforced by a penalty approach:

$$
F = \begin{cases} 
F & \text{if } \max(g_1, g_2) \leq 0 \\
F - p \max(g_1, g_2) & \text{otherwise}
\end{cases}
$$

We used a penalty parameter $p = 50$. The constraints create a narrow circular ridge at $r = 1.005$. Figure 4–22 show the contours of the penalized function for $n = 2$. The
Figure 4–22: Contours of the penalized fitness function for $n = 2$. The reference point $x_0 = (0.02, 0.53)$ is marked by a ‘x’.

topology of this problem presents similar features to the didactic example of Section 4.1: due to the curvature of the high fitness regions, and the particular location of the optimum (at a corner of the feasible domain), the most probable point yielded by a univariate model will be an infeasible design, far from the maximum of the fitness function. Noting that this particular topology is caused by the presence of constraints,\textsuperscript{11} we decided to used $g_1$ as auxiliary variable\textsuperscript{12}.

4.6.2 The algorithms

Continuous versions of UMDA and DDOA, respectively called cUMDA and cDDOA, were implemented. In cUMDA, the continuous probability density of selected

\textsuperscript{11} This is a common situation in optimization: the optima typically lie on the boundary of the feasible domain, so that the topology of the fitness around the optimum is for the most part determined by the constraints.

\textsuperscript{12} Choosing $g_2$ would yield identical results.
points was represented by a product of marginal densities:

\[ p(x_1, \ldots, x_n) = \prod_{k=1}^{n} p(x_k) , \quad (4.21) \]

where the marginal densities were approximated by univariate normal distributions whose mean \( \mu_k \) and standard deviation \( \sigma_k \) were estimated from the population of selected points. A lower bound on \( \sigma_{\text{min}} \) was imposed on the standard deviation to prevent loss of diversity.

The same representation was used for cDDOA. The density of the auxiliary variable \( V \equiv g_1 \) was represented by a normal distribution as well. The same two-step algorithm as in Section 4.3.1 was implemented for the continuous case. The goal is to force points to lie in the feasible domain by filtering out poor candidates based on \( p(V) \).

4.6.3 Results and discussion

The two algorithms were applied to the problem (4.18) for two numbers of variables, \( n = 2 \) and \( n = 10 \). A population size of \( \lambda = 100 \) was chosen, and truncation selection of ratio \( \tau = 0.3 \) was applied. The pool size used for cDDOA was \( \nu = 400 \) points. Three different values of the minimum standard deviation were tried: \( \sigma_{\text{min}} = 0.001 \), \( \sigma_{\text{min}} = 0.01 \), and \( \sigma_{\text{min}} = 0.1 \) (for the two distributions \( p(x) \) and \( p(V) \)). Figure 4–23 compares the evolution of the best fitness for cUMDA and

\[ ^{13} \text{In this particular case, one may be tempted to discard all infeasible points, but it is often a counter-productive strategy, as this precludes any short-cut through the infeasible domain.} \]
cDDOA.

Figure 4–23: Evolution of the best fitness for cUMDA and cDDOA, \( n = 2 \) (average over 50 runs).

For the three values of \( \sigma_{\text{min}} \) tested, cDDOA outperforms cUMDA. In the case \( \sigma_{\text{min}} = 0.001 \), the initial convergence of cUMDA is faster than that of cDDOA until the population has found the tunnel: in this initial phase, the auxiliary distribution hampers the convergence to the feasible domain, but once the distribution is centered in the tunnel, it speeds up convergence toward the global optimum \( F = 1.11 \) (hence the change in slope), and cDDOA becomes more efficient than cUMDA. The best performance is achieved for \( \sigma_{\text{min}} = 0.01 \) for both algorithms. For this value of the minimum standard deviation, cDDOA needs three times fewer evaluations than cUMDA (10,000 versus 30,000) to find the global optimum.

As was observed for the discrete DDOA, the two-distribution scheme reduces the algorithm’s sensitivity to diversity controlling parameters, here \( \sigma_{\text{min}} \). This is visible for \( \sigma_{\text{min}} = 0.1 \): for that value, cUMDA exhibits a slow convergence because
the large value of $\sigma_{\text{min}}$ prevents the algorithm from focusing on high-fitness regions. The auxiliary variable based filter substantially improves the search efficiency by discarding meaningless points.

Figure 4–24 provides another perspective on the convergence of the distributions for the continuous double-distribution algorithm. Figure (a) shows the value of $\mu_1$ and $\mu_2$ during the optimization, and Figure (b) monitors $\mu_V$. Starting from an initial distribution centered around $(\sqrt{2}/2, \sqrt{2}/2)$, the means of the distributions follow the tunnel slightly on the outside, as indicated by the auxiliary variable distribution ($V$ is negative outside the circle $r = 1$, until it finds the optimum (1,0).

![Figure 4–24: Convergence of the primary and auxiliary distributions for cDDOA, $n = 2$ (50 runs). Starting from the center of the design space, the distributions first converge to the “tunnel”, then they follow it on its outside down to the optimum (1,0).](image)

In the case $n = 10$, cDDOA displayed a very poor efficiency compared to cUMDA, as shown in Figure 4–25. While the latter quickly reached high-fitness regions, the progress of the double-distribution algorithm was hindered by the auxiliary variable distribution.
Figure 4-25: Evolution of the best fitness for cUMDA and cDDOA (“dd-cumda” on graph), $n = 10$ (average over 50 runs).

This example reveals one of EDA’s limitations, which is enhanced by DDOA: the distribution $p(x)$ theoretically represents the distribution of promising regions. However, when the space is only sparsely covered by the population, entire regions may not be visited and hence they will be ignored in subsequent iterations of the search (this is the selection error introduced in Section 2.3: EDAs can only approximate the selection probability $p^*(x)$, so that good designs may get a zero selection probability). This effect is illustrated in Figure 4–26: low values of $x$ are not sampled, consequently, they do not contribute to the estimated distribution of promising regions. As a result, the search will focus on already visited regions and convergence to the optimum will be slowed down.

The ironic consequence of this effect is that poor statistical models, such as cUMDA, which inaccurately represent promising regions at time $t$, will sample more often from areas where no observations of high-fitness points have been made. In
Figure 4–26: When regions of the design space are not covered by the population, no information about them is collected, and they are excluded from the search in subsequent iterations.

Figure 4–26: When regions of the design space are not covered by the population, no information about them is collected, and they are excluded from the search in subsequent iterations.

particularly, they will more often be able to create points where no observation at all has been made, which helps cUMDA escape early “good” (but overall poor) regions. This is what is observed for the 10 variable problem. In contrast, cDDOA, which constructs a more accurate model of $p(x)$, more efficiently constrains the search to already visited areas, even though they do not contain the optimum. Specifically, the auxiliary variable $V$ favors candidates that are located on the same sphere $r = \text{constant}$ as the selected points, which prevents convergence toward high-fitness regions in the early stages of the optimization, when the population is far from the feasible domain.

4.6.4 Improvement to the algorithm

One way of addressing the problem identified in the previous section is to drastically increase the population so as to ensure a good coverage of the entire design space. However, this is not a practical situation in high-dimensional spaces, as the number of points required would be prohibitive. A new strategy was proposed to address the case where insufficient information is available to build a satisfactory
statistical model of promising regions. The idea is to detect situations in which the statistical model cannot be trusted. One such situation occurs when the selected and non-selected points are linearly separable in a given space. In such a case, the optimum is likely to lie outside the region marked by the selected points in the direction perpendicular to the separation boundary going from the non-selected to the selected points, in a simplex-like manner (Nelder and Mead, 1965). In these identified pathological cases, the proposed strategy consists in keeping the $\mu$ candidate points with the most extreme variable values in the direction of increasing fitness. The decision to disregard the statistical model and to base the search on directionality instead is made on the basis of the degree of separation of selected and non-selected points in the auxiliary variable space.

Figure 4–27: Evolution of the best fitness for cUMDA and cDDOA (“dd-cumda” on graph), $n = 10$ (average over 50 runs).

This strategy was implemented for the 10-variable problem by looking at separation in the $V$-space (which is easy since it is unidimensional here). The mean best
fitness is shown in Figure 4-27. The modified cDDOA clearly outperforms cUMDA. The detection of pathological cases based on good and bad points (when EDAs usually consider only good points), and the implementation of an appropriate strategy to deal with these situations leads to a dramatic improvement of cDDOA’s convergence velocity. The concept of alternating between probabilistic and directional search has proven its validity.

4.7 Conclusion

In this chapter, an original strategy for representing variable dependencies was proposed: the idea is to combine two simple statistical distributions: a univariate distribution of the design variables, and an approximate distribution of a small number of auxiliary variables to obtain a complex distribution.

The auxiliary variables represent joint influences of the design variables on the fitness function. Such variables can be found in many optimization problems. In this chapter, the two-distribution algorithm (DDOA) was applied to discrete laminate optimization problems, and to a continuous optimization problem. Experimental results showed that the proposed strategy consistently outperforms UMDA and a GA. An analysis of the diversity revealed that DDOA maintains a high level of entropy in the search distribution, and that it effectively allocates that diversity to high-fitness regions.

Finally, a strategy combining distribution-based and directional search techniques was proposed to address one of EDA’s limitations, namely the difficulty to
identify promising regions: the method detects pathological situations where the dis-
tribution of promising areas is not to be trusted, and where a directional approach
is more appropriate. Experimental results on a continuous problem show that this
strategy can dramatically improve the efficiency of the algorithm.
CHAPTER 5
CONCLUSION

5.1 Summary of the findings

The generalization of the use of composite materials in recent years has spurred intensive research on effective optimization techniques suited for the particular problems that arise when designing a composite structure. In particular, composite laminate optimization (determination of the stacking sequence, or number of plies, ply angles, materials, etc) often leads to difficult combinatorial problems which conventional optimization techniques cannot solve efficiently.

To address difficulties inherent to laminate optimization, such as the discreteness of the design variables (the ply angles), and the multimodality of the objective function, new algorithms have been proposed over the years. In this work, the implementation of a new class of algorithms, called estimation of distribution algorithms (EDAs), for laminate optimization, was investigated. These algorithms are stochastic optimization methods that explore the design space based on a search distribution: at each iteration, the distribution of promising points is estimated from a set of points selected for their high objective function evaluation, and that distribution is used to bias the search toward good regions at the next iteration.
5.1.1 Incorporation of variable dependencies through physics-based auxiliary variables

The first contribution of this work is the use of physics-based knowledge to improve the statistical model of promising regions. The idea is to enhance the efficiency of the optimization by modeling the distribution of good designs more accurately. The majority of existing EDAs are based on stringent statistical independence assumptions between variables in the populations of selected points or resort to complex statistical models of these interactions, leading to high computational costs to estimate the model parameters. In this work, we propose to model the complex distribution of selected points by combining two simple probability distributions: the distribution in the space of the design variables, represented by a univariate model (no variable interactions), and the distribution in the space of auxiliary variable chosen to capture joint actions of the design variables. This is a way of incorporating information about the form of the distribution and improving the model accuracy at low computational cost.

In the case of composite laminate optimization, the design variables are the fiber angles, and laminate-level geometric stiffness quantities called lamination parameters are a natural choice of auxiliary variables. The resulting algorithm, named double-distribution optimization algorithm (DDOA), was applied to three laminate optimization problems: two problems (one purely extensional problem an one extensional/flexural problem) where the response depends entirely on the lamination parameters, and one extensional problem where they capture only part of the response. In the three problems, the implemented strategy outperformed an algorithm
based on univariate distributions only (UMDA) and a standard genetic algorithm, in particular in highly constrained fitness landscapes. Experiments showed that the advantage of DDOA increases with the problem dimension.

Even though most of this work concerns composite laminate optimization, the optimization framework proposed can be applied in other fields. The last part of this work demonstrated the validity of the method for a general continuous optimization problem. In that problem, one of the constraints played the role of auxiliary variable, thus demonstrating one of the possible uses of auxiliary variables: ensure the feasibility of the points generated. Experimental results showed that using the distribution of auxiliary variables improves the convergence velocity.

5.1.2 Control of the diversity

A second emphasis of this work concerned the role of the diversity of the search distributions was investigated. Indeed, the amount, as well as the nature of variability directly influences the efficiency of stochastic algorithms: too little variability results in a local optimum, while too much randomness in the search leads to wasteful function evaluations in poor regions (this is the well-known exploitation/exploration compromise).

An analysis of two diversity preservation mechanisms, mutation and a lower bound on marginal probabilities, was conducted for two estimation of distribution algorithms: a UMDA and DDOA. The main goal was to prevent premature convergence of the search distribution. The study demonstrated that such a diversity preservation mechanism is essential to guarantee EDAs’ efficiency. It appeared that the best
strategy is to explicitly prevent convergence of the search distribution by imposing bounds on the distribution (in contrast to standard EAs, where such a strategy is impossible) because it inflicts smaller perturbations to the search compared to the mutation operator.

A side effect of the double-distribution algorithm was its stability with respect to the setting of the parameters governing the level of randomness in the search (mutation rate, bound on the probabilities, ...): the two-distribution search strategy implemented in DDOA provided a compensation mechanism for insufficient or excessive variability. An analysis of the diversity for UMDA and DDOA revealed that the double-distribution approach provided a way of injecting variability in high-fitness regions of the criterion space, making a more efficient use of the function evaluations.

5.1.3 Combination of distribution-based and directional search mechanisms

Finally, a hybrid search strategy combining distribution-based and directional mechanisms was proposed. The idea is to detect situations where the probabilistic model of promising regions is flawed, due to an insufficient sampling of the design space, and to switch to a simplex-like directional search mechanism more appropriate when the population is away from high-fitness regions. Such pathological cases occur when selected points and non-selected points are linearly separable in a given space. The direction perpendicular to the boundary, going from the non-selected to the selected points was successfully used to complement the EDA search.
5.2 Potential avenues for future research

Many aspects of the double-distribution optimization algorithm have not been fully studied. Among the factors that need further investigation, we can mention the following issues:

- **effect of initialization**: it is widely recognized (see, for example, Kallel and Schoenauer, 1997) that an improved initialization procedure can dramatically increase the efficiency of evolutionary algorithms. DDOA can provide a convenient way of assuring a more uniform sampling of the criterion space by increasing the pool size $\nu$ for the first iteration;

- **adaptive scheme** for the determination of the algorithm parameters, in particular the relative importance of the primary and auxiliary variables in the search (through the value of $\nu$), choosing the more appropriate space for the optimization (a strategy akin to “automatic coding choice”);

- **elaboration of a unified “framework for stochastic and directional optimization”**, following the preliminary work conducted in Section 4.6.4. The objective would be to give the algorithm the ability to choose the best search mechanism during the optimization.
6.1 Optimisation de stratifiés composites

Les matériaux composites sont aujourd’hui couramment utilisés dans des domaines aussi divers que l’aéronautique, la construction navale, l’automobile ou dans des équipements de loisir, où leurs propriétés spécifiques élevées sont très appréciées. Les matériaux composites se distinguent de beaucoup d’autres matériaux plus traditionnels par le fait que leurs propriétés mécaniques peuvent être adaptées à la structure dans laquelle ils sont employés. Ceci constitue bien sûr un grand avantage, mais rend la conception de structures plus complexe puisqu’il ne s’agit plus seulement de choisir le matériau et la géométrie de la structure, mais également de définir l’organisation interne du matériau. Les stratifiés sont constitués d’un empilement de couches (plis) formées de différents types de renforts (dans les cas qui nous intéressent de fibres alignées selon une direction privilégiée) dont la cohésion est assurée par une résine. Leurs propriétés mécaniques dépendent directement de l’orientation des fibres. L’objectif de l’optimisation est donc de choisir l’orientation de chaque pli de manière à rendre extrémal un certain critère, par exemple le déplacement en un point, le poids, le coût, la première fréquence propre, etc., tout en respectant un certain nombre de contraintes telles que des critères de résistance (par exemple la contrainte ou la déformation maximale, l’énergie de déformation maximale), la facilité de mise en œuvre. À cause de contraintes de fabrication, l’orientation des fibres et typiquement
limitée à quelques valeurs discrètes (par exemple 0°, 45°, 90°). La conception de structures composites conduit par conséquent à des problèmes d’optimisation combinatoire, formulés de la façon suivante :

\[
\text{maximiser } F(\theta_1, \ldots, \theta_n) \\
\text{tel que } g_i(\theta_1, \ldots, \theta_n) \geq 0, \ i = 1, \ldots, r, \\
\text{avec } \theta_j \in A, \ j = 1, \ldots, n
\]

où \( F \) est la fonction coût, aussi appelée “fitness” dans le cadre de l’optimisation évolutionnaire, les \( g_i \) sont les contraintes, et \( A \) est l’ensemble des valeurs possibles pour les variables de conception \( \theta_k \), qui représentent ici l’orientation des fibres dans chacun des plis du stratifié.

Dans cette optique, nous nous proposons de développer des méthodes d’optimisation de type statistique.

### 6.2 Optimisation statistique

Diverses approches ont été utilisées pour résoudre les problèmes d’optimisation de stratifiés. Ces dix dernières années ont vu l’émergence de méthodes stochastiques, avec le recuit simulé, et plus récemment, les algorithmes évo- lutionnaires. Les algorithmes évo- lutionnaires (dont les algorithmes génétiques sont un exemple connu) explorent l’espace de recherche en associant des portions des individus les plus prometteurs de populations de points (croisement) et en appliquant des perturbations aléatoires (mutation). Ces algorithmes se montrent efficaces pour de nombreux problèmes, cependant leur efficacité optimale n’est pas prouvée, et la principale justification de leur performance est basée sur la métaphore biologique de la théorie
Les algorithmes à estimation de distribution (EDA pour Estimation of Distribution Algorithms) constituent une formalisation statistique des algorithmes évolutionnaires. Ils reprennent l’idée d’exploration par des populations de points, mais utilisent des méthodes statistiques pour extraire explicitement de l’information sur la localisation de l’optimum. La méthode consiste en un générateur de point sous la forme d’une distribution de probabilité selon laquelle des points sont échantillonnés, qui biaise l’exploration en faveur de régions qui ont de fortes chances de contenir l’optimum.

L’algorithme général est présenté sur la Figure 6–1. Supposons que l’on souhaite trouver le maximum d’une fonction $F(\mathbf{x})$, $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$. En l’absence de toute information sur la localisation du maximum (ou des maxima) $\mathbf{x}^*$, la distribution de recherche $p(\mathbf{x})$ est initialisée comme une distribution uniforme sur $\mathbb{R}^n$. Une
population de $\lambda$ individus est créé par échantillonnage de $p(x)$, et la valeur de $F$ est calculée pour chacun de ces points. Parmi ces points, $\mu$ bons points sont choisis en utilisant une procédure de sélection basée sur la valeur de $F$. Alors, la distribution de ces points est évaluée et utilisée pour mettre à jour la distribution de recherche $p(x)$. La procédure d’échantillonnage/sélection/mise à jour est répétée jusqu’à ce qu’un critère d’arrêt soit satisfait.


Un bon modèle statistique doit donc être à la fois simple (peu de paramètres) et
posséder une flexibilité adaptée à la topologie de $p(x)$. Ces deux préoccupations sont à la base des approches proposées dans cette dissertation : un premier algorithme, appelé "Univariate Marginal Distribution Algorithm" (UMDA) utilise un modèle simple dans lequel les variables sont considérées indépendantes (la valeur d’une variable n’affecte pas la façon dont les autres variables agissent sur la fitness). Ce modèle possède l’avantage de la simplicité, mais n’est pas approprié pour représenter des distributions complexes dans lesquelles les variables interagissent fortement les unes avec les autres.

Dans certaines situations, la complexité apparente d’une fonction à approximer tient à un mauvais choix de la base d’approximation. Une façon de représenter ces fonctions en apparence complexes (par rapport à la base d’approximation) consiste à effectuer un simple changement de base. Ainsi, on peut obtenir une meilleure précision en conservant le même nombre de paramètres. Par exemple, une infinité de termes est nécessaire pour représenter une fonction linéaire par une série de Fourier, alors qu’un seul terme suffit si l’on utilise une base polynomiale. De la même manière, il est possible d’améliorer considérablement la précision d’un modèle statistique en injectant de la connaissance sur la structure de la distribution $p(x)$.

Dans le cadre de l’optimisation de stratifiés composites, comme dans de nombreux autres domaines, la connaissance de la physique des problèmes traités peut être utilisée avec profit afin d’extraire de l’information sur la forme de la distribution $p(x)$. En effet, dans de nombreuses applications, la réponse du système dépend de grandeurs macroscopiques qui capturent l’influence conjointe de certaines variables sur le comportement global. En d’autres termes, ces grandeurs caractérisent des couplages
physiques de ces variables. Il semble donc justifié d’inclure l’information sur la
distribution de ces grandeurs afin de raffiner le modèle statistique. En particulier, les
propriétés de rigidité des stratifiés (qui entrent dans le calcul de la réponse globale de
la structure) sont complètement déterminées par un petit nombre de grandeurs macro-
scopiques appelées “paramètres de stratification”, qui sont définis comme sommes de
cosinus des angles des fibres dans les différents plis. Les variables d’orientation des
fibres n’influencent la fonction coût qu’à travers ces grandeurs. Par conséquent, la dis-
tribution des points sélectionnés peut être exprimée simplement à l’aide d’un nombre
réduit de paramètres dans l’espace des paramètres de stratification.

Sur la base de ces observations, un nouvel algorithme, appelé “Double-Distribution
Optimization Algorithm” (DDOA) est proposé. Cet algorithme combine un modèle
simple (modèle univarié) de la distribution des variables d’orientation et un modèle de
la distribution des paramètres de stratification, afin de prendre en compte certaines
interactions entre les variables.

6.3 Contenu de la thèse

Cette thèse est organisée comme suit:

• le chapitre 1 propose une introduction générale à l’optimisation de stratifiés
  composites, en insistant sur le type de problèmes auxquels elle conduit. Les
  méthodes existantes, ainsi que leurs forces et faiblesses, sont présentées. Le
  chapitre termine par une brève introduction sur les méthodes d’optimisation
  évolutionnaires, et les méthodes d’optimisation par algorithmes à estimation de
  distribution.
• Dans le chapitre 2, les algorithmes à estimation de distribution (EDA) sont présentés dans le contexte plus général de l’optimisation stochastique. La mise au point d’algorithmes est placée dans un cadre statistique: les différents constituants des EDA sont détaillés, et les enjeux théoriques liés à la sélection de bons points et à l’estimation de distribution sont précisés. Le but de ce chapitre est d’introduire les concepts qui vont présider à l’élaboration d’algorithmes dans le reste de cette thèse.

• Dans le chapitre 3, UMDA, un EDA basé sur un modèle univarié de la distribution des points prometteurs, est modifié pour traiter des problèmes d’optimisation de stratifiés. L’influence de divers paramètres de l’algorithme original (taille de population et intensité de sélection) est observée sur un problème simple de maximisation de la rigidité longitudinale. L’évolution de la performance avec l’augmentation du nombre de variables est étudiée de manière empirique et théorique. On observe que l’efficacité de l’algorithme est limitée par des erreurs de sélection (les valeurs optimales peuvent ne pas être présentes dans la population des points sélectionnés) et d’échantillonnage (même si le modèle représente parfaitement la distribution des points prometteurs, les région optimales peuvent ne pas être visitées du fait de la taille finie des populations).

Sur la base de ces conclusions, des améliorations de l’algorithme sont proposées. Ces additions à l’algorithme de base ont pour objectif commun de préserver la diversité de la distribution, et ainsi sa capacité d’exploration. Trois approches sont testées : l’ajout d’un opérateur de mutation similaire à ceux utilisés dans les algorithmes évolutionnaires conventionnels, l’adoption d’une
nouvelle procédure de mise à jour des distributions incorporant une mémoire des générations précédentes, et un contrôle explicite de la dégénérescence des distributions (on force les distributions marginales à conserver une probabilité non nulle pour chacune des valeurs de la variable concernée).

Enfin, nous avons appliqué UMDA à trois problèmes d’optimisation de stratifiés composites : la maximisation sous contraintes de la première fréquence propre d’une plaque stratifiée, la minimisation de la rigidité en cisaillement dans le cas où l’espace de recherche est dissymétrique, et la maximisation de la résistance d’un stratifié, selon un critère de déformations maximales. Ces trois problèmes ont été choisis avec soin de façon à révéler le comportement de l’algorithme sur des topologies caractéristiques des divers problèmes rencontrés lors de la conception de structures composites : multimodalité, espace de recherche étroit, espace très contraint, etc. Sur ces problèmes, l’algorithme a été comparé à deux autres algorithmes stochastiques : un algorithme génétique standard, et un algorithme de recherche par incrément locaux (stochastic hill-climber, SHC). Pour ces problèmes, UMDA se montre aussi efficace ou plus efficace que l’algorithme génétique, et possède un avantage sur SHC quand le problème est très contraint ou fortement multimodal.

• Le chapitre 4 introduit un nouvel algorithme à évolution de distribution, appelé Double-Distribution Optimization Algorithm (DDOA), présenté sur la Figure 6–2. L’idée est d’améliorer la qualité de l’approximation de \( p(x) \) en injectant de l’information sur la structure de la distribution des points sélectionnés.
L’algorithme est basé sur l’étude de deux distributions : la distribution des variables d’orientation des fibres, représentée par un modèle à variables indépendantes, et la distribution des paramètres de stratification, qui capturent des couplages physiques entre variables, représentée par une somme de noyaux gaussiens. Les deux distributions coopèrent de la façon suivante : un grand nombre de points est généré par échantillonnage de la distribution \( p(x_1, \ldots, x_n) \), puis la population finale est obtenue en filtrant cette population à travers la distribution \( p(V) \). En faisant varier le rapport entre la population de points candidats et la taille finale de la population, il est possible de régler l’importance relative de chacune des distributions dans la création de nouveaux points.
Nous avons appliqué DDOA à trois problèmes d’optimisation de stratifiés composites : la maximisation de la rigidité transversale $A_{22}$ sujette à une contrainte sur le coefficient de Poisson, la minimisation du coefficient de dilatation longitudinal $\alpha_x$ tel que la première fréquence propre ne soit pas inférieure à une certaine valeur $f_{\text{min}}$, et la maximisation de la résistance, selon un critère de déformation maximale. Pour ces trois problèmes, DDOA se montre plus efficace que UMDA et qu’un algorithme génétique standard, qui ne gèrent pas, ou gèrent mal les dépendances entre variables. En outre, l’avantage de DDOA croit avec la taille du problème (nombre de plis), car l’effet de réduction du nombre de variables par le passage dans l’espace des variables auxiliaires joue pleinement.

L’influence de deux mécanismes de préservation de la diversité (mutation et limitation explicite des distributions marginales) est ensuite examinée. L’étude révèle que la méthode la plus efficace pour empêcher la convergence prématurée de la distribution consiste à placer des bornes explicites sur les distributions marginales.

Un des effets inattendus de l’algorithme à deux distributions est la plus grande stabilité de l’efficacité par rapport au réglage des paramètres de variabilité (taux de mutation ou limite sur les distributions). La stratégie proposée offre un mécanisme de correction de la distribution de recherche : elle permet de réduire ou augmenter la variabilité selon qu’elle est excessive ou insuffisante.

Afin de démontrer le caractère général de la méthode proposée, une version continue de DDOA est enfin présentée, dans laquelle à la fois la distribution des variables du problème et la distribution des variables auxiliaires sont continues.
Dans l’algorithme proposé, les deux distributions sont représentées par un produit de distributions normales univariées. Cette version continue de DDOA est testée sur un problème général de maximisation d’une fonction quadratique, sujette à des contraintes définissant un domaine faisable étroit et courbé. Comme les contraintes définissent contribuent pour une large part à la topologie de la fitness, on choisit une contrainte comme variable auxiliaire. Les résultats de tests montrent que l’utilisation de cette information améliore la performance de l’algorithme quand la population initiale se trouve dans le domaine faisable. Dans le cas contraire, elle peut freiner la convergence vers ce domaine. Une stratégie pour traiter ces situations est proposée et mise en œuvre. Avec cette modification, DDOA converge plus rapidement que UMDA vers l’optimum.

6.4 Conclusions

Dans cette thèse, des méthodes d’optimisation statistique appelées algorithmes à estimation de distribution ont été adaptées au cas particulier de l’optimisation de stratifiés composites. Après une étude approfondie du comportement d’un algorithme existant (UMDA), une méthode d’optimisation permettant d’incorporer de l’information sur la structure du problème dans le but d’améliorer l’exactitude du modèle statistique des régions prometteuses est proposée. L’algorithme proposé, appelé DDOA, utilise deux distributions pour guider la recherche : la distribution des variables de conception, et la distribution de variables auxiliaires représentant l’action conjointe de plusieurs variables.

L’algorithme a montré son efficacité sur plusieurs problèmes d’optimisation de
composites, où une réduction du temps de convergence a été observée par rapport à UMDA et à un algorithme génétique. La méthode est toutefois générale, et peut être appliquée à tout problème dans lequel des variables auxiliaires pertinentes peuvent être introduites.
REFERENCES


APPENDIX A
COMPOSITE LAMINATE OPTIMIZATION

A.1 Basic mechanics of laminated plates

A.1.1 Stiffness matrices

Laminated plates are made up of a number $n_p$ layers, also called lamina or plies, stacked on top of each other. Usually, the materials used for the lamina are made from fibers aligned according to a particular direction (unidirectional lamina) or two perpendicular directions, which gives them orthotropic properties of symmetry. Since the layers are thin, one usually makes the assumption of plane stress, which means that all transverse stresses $\sigma_{13}$, $\sigma_{23}$, and $\sigma_{33}$ are neglected. The constitutive relation for such materials then becomes

\[
\begin{bmatrix}
\epsilon_{11} \\
\epsilon_{22} \\
\gamma_{12}
\end{bmatrix} =
\begin{bmatrix}
\frac{1}{E_1} & -\frac{\nu_{12}}{E_2} & 0 \\
-\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & 0 \\
0 & 0 & \frac{1}{G_{12}}
\end{bmatrix}
\begin{bmatrix}
\sigma_{11} \\
\sigma_{22} \\
\sigma_{12}
\end{bmatrix}
\]  

(A.1)

The stresses are given by the inverse relation

\[
\begin{bmatrix}
\sigma_{11} \\
\sigma_{22} \\
\sigma_{12}
\end{bmatrix} =
\begin{bmatrix}
\frac{E_1}{1-\nu_{12}\nu_{21}} & \nu_{12}\frac{E_2}{1-\nu_{12}\nu_{21}} & 0 \\
\nu_{12}\frac{E_2}{1-\nu_{12}\nu_{21}} & \frac{E_2}{1-\nu_{12}\nu_{21}} & 0 \\
0 & 0 & G_{12}
\end{bmatrix}
\begin{bmatrix}
\epsilon_{11} \\
\epsilon_{22} \\
\gamma_{12}
\end{bmatrix}
\] ,

(A.2)

or

\[
\sigma = Q\epsilon
\] (A.3)
When the laminae are rotated with respect to a reference coordinate system $(x, y, z)$, the stiffness matrix in that coordinate system, $\bar{Q}$, is obtained by

$$\bar{Q} = T_\epsilon^T \bar{Q} T_\epsilon$$  \hspace{1cm} (A.4)

where $T_\epsilon$ is the rotation matrix for strains from an arbitrary coordinate system $(x, y, z)$ to the material directions $(1, 2, 3)$ ($\epsilon_{12} = T_\epsilon \epsilon_{xy}$):

$$T_\epsilon = \begin{bmatrix} c^2 & s^2 & cs \\ s^2 & c^2 & -cs \\ -2cs & 2cs & c^2 - s^2 \end{bmatrix}$$  \hspace{1cm} (A.5)

with $c = \cos \theta$ and $s = \sin \theta$.

In the classical laminated plate theory, the behavior of a laminate is described by the in-plane force resultant $N$ and the moment resultant $M$:

$$N = \begin{bmatrix} N_x \\ N_y \\ N_{xy} \end{bmatrix} = \int_{-h/2}^{h/2} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix} dz$$

$$M = \begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} = \int_{-h/2}^{h/2} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix} zdz$$  \hspace{1cm} (A.6)

After introducing the cinematic relation

$$\epsilon = \epsilon_0 + z \kappa$$  \hspace{1cm} (A.7)

where $\epsilon_0$ is the mid-plane extensional strain and $\kappa$ is the approximated curvature:

$$\kappa = - \begin{bmatrix} \frac{\partial^2 w}{\partial x^2} \\ \frac{\partial^2 w}{\partial y^2} \\ \frac{\partial^2 w}{\partial x \partial y} \end{bmatrix}$$  \hspace{1cm} (A.8)
one obtains the relationships between the in-plane force and moments, and the mid-plane strains and the curvatures:

\[ N = A\epsilon_0 + B\kappa \]  \hspace{1cm} (A.9)

\[ M = B\epsilon_0 + D\kappa \]  \hspace{1cm} (A.10)

where \( A \), \( B \), and \( D \) are the extensional stiffness matrix, the extension-bending coupling matrix, and the flexural stiffness matrix, respectively:

\[ A = \int_{-h/2}^{h/2} \bar{Q}(z) dz = \sum_{k=1}^{n_p} t_k \bar{Q}_k \]  \hspace{1cm} (A.11)

\[ B = \int_{-h/2}^{h/2} \bar{Q}(z)z dz = \sum_{k=1}^{n_p} t_k \bar{Q}_k \]  \hspace{1cm} (A.12)

\[ D = \int_{-h/2}^{h/2} \bar{Q}(z)z^2 dz = \sum_{k=1}^{n_p} t_k \bar{Q}_k \]  \hspace{1cm} (A.13)

### A.1.2 Lamination parameters

In this work, we will only consider laminates that have a plane of symmetry and that satisfy the balance condition (for each \( \theta \)-ply, there is a \( -\theta \)-ply in the laminate). More specifically, we will restrict our study to laminates of the form \([\pm \theta_1/\pm \theta_2/\ldots/\pm \theta_n]\). The symmetry condition causes the coupling stiffness matrix \( B \) to vanish. The balance condition causes the in-plane extension-twisting coefficients \( A_{16} \) and \( A_{26} \) to be zero, and \( D_{16} \) and \( D_{26} \) to be negligible. The stiffness matrices of such laminates can be expressed in terms of homogenized geometrical stiffness quantities \( V_i^* \) and \( W_i^* \),
called *lamination parameters*, and material invariants \( U_1 \) to \( U_5 \):

\[
\begin{align*}
\{A_{11}\} &= \frac{1}{h} \begin{bmatrix} U_1 & U_2 & U_3 \\ U_1 & -U_2 & U_3 \\ U_5 & 0 & -U_3 \end{bmatrix} \begin{bmatrix} 1 \\ V_1^* \end{bmatrix}, \\
\{A_{22}\} &= \frac{1}{h} \begin{bmatrix} U_1 & U_2 & U_3 \\ U_1 & -U_2 & U_3 \\ U_5 & 0 & -U_3 \end{bmatrix} \begin{bmatrix} 1 \\ V_3^* \end{bmatrix}, \\
\{A_{12}\} &= \frac{1}{12} \begin{bmatrix} U_1 & U_2 & U_3 \\ U_1 & -U_2 & U_3 \\ U_5 & 0 & -U_3 \end{bmatrix} \begin{bmatrix} 1 \\ W_1^* \end{bmatrix}, \\
\{A_{66}\} &= \frac{1}{12} \begin{bmatrix} U_1 & U_2 & U_3 \\ U_1 & -U_2 & U_3 \\ U_5 & 0 & -U_3 \end{bmatrix} \begin{bmatrix} 1 \\ W_3^* \end{bmatrix}, \\
\end{align*}
\]  

(A.14)

where the lamination parameters are given by

\[
V_{\{1,3\}}^* = \frac{2}{h^3} \int_0^{h/2} \{ \cos 2\theta, \cos 4\theta \} dz
\]

\[
= \frac{1}{n} \sum_{k=1}^{n} \{ \cos 2\theta_k, \cos 4\theta_k \}  
\]

(A.16)

and

\[
W_{\{1,3\}}^* = \frac{24}{h^3} \int_0^{h/2} \{ \cos 2\theta_k, \cos 4\theta_k \} z^2 dz
\]

\[
= \frac{1}{n^3} \sum_{k=1}^{n} a_k \{ \cos 2\theta_k, \cos 4\theta_k \} . 
\]

(A.17)

The ply-numbering convention adopted here is shown in Figure A–1.
The material invariants are given by

\[ U_1 = \frac{1}{8} (3Q_{11} + 3Q_{22} + 2Q_{12} + 4Q_{66}) \] (A.18)

\[ U_2 = \frac{1}{2} (Q_{11} - Q_{22}) \] (A.19)

\[ U_3 = \frac{1}{8} (Q_{11} + Q_{22} - 2Q_{12} - 4Q_{66}) \] (A.20)

\[ U_4 = \frac{1}{8} (Q_{11} + Q_{22} + 6Q_{12} - 4Q_{66}) \] (A.21)

\[ U_5 = \frac{1}{8} (Q_{11} + Q_{22} - 2Q_{12} + 4Q_{66}) \] . (A.22)

The equations (A.16) and (A.17) that define the lamination parameters determine feasible and infeasible domains in the \( \mathbf{V} \)-space: some combinations of \( V^*1, V^*_3, W^*_1, \) and \( W^*_3 \) are not acceptable. For instance, if we consider an in-plane problem, where \( V^*1, V^*_3 \) only influence the objective function, \( V^*_3 = -1 \) implies that the value of all the ply angles must be \( 45^\circ \), hence the value of \( V^*1 \) cannot be chosen arbitrarily, but has to be 0.

The boundaries of the feasible domain has been derived for simple combinations of the lamination parameters:
• for \((V^*1, V^*_3)\), Miki and Sugiyama (1991) identified the feasible space

\[
V^*_3 \geq 2V^*_1^2 - 1 \quad V^*_3 \leq 1 ,
\]

(A.23)

• similarly, for \((W^*1, W^*_3)\), the feasible space is defined by

\[
W^*_3 \geq 2W^*_1^2 - 1 \quad W^*_3 \leq 1 .
\]

(A.24)

These relationships define a parabolic feasible domain (white area), shown in Figure A–2.

For more complex combinations, one has to resort to approximate relationships. Diaconu et al. (2002b) derived approximate relationships for an arbitrary combination of lamination parameters using variational methods. In particular, approximate
boundaries for the case \((V^*, V^*_3, W^*_1, W^*_3)\) are given by

\[ 4(W^*_1 - 1) \geq (V^*_1 - 1)^3 \]  \hspace{1cm} (A.25)

\[ 4(W^*_3 - 1) \geq (V^*_3 - 1)^3 \]  \hspace{1cm} (A.26)

\[ 4(W^*_1 + 1) \geq (V^*_1 + 1)^3 \]  \hspace{1cm} (A.27)

\[ 4(W^*_3 + 1) \geq (V^*_3 + 1)^3 \]  \hspace{1cm} (A.28)

\[ V^*_3 \geq 2V^*_1^2 - 1 \]  \hspace{1cm} (A.29)

\[ W^*_3 \geq 2W^*_1^2 - 1 \]  \hspace{1cm} (A.30)

\[ V^*_3 \leq 1 \]  \hspace{1cm} (A.31)

\[ W^*_3 \leq 1 \]  \hspace{1cm} (A.32)

### A.2 Mechanical properties used in this work

#### A.2.1 Poisson’s ratio

In the case of isotropic materials, the Poisson’s ratio is defined as the ratio between the transverse extension over the longitudinal extension for a uniaxial loading \((\sigma_{22} = \tau_{12} = 0)\):

\[ \nu = \frac{\epsilon_{22}}{\epsilon_{11}} \]  \hspace{1cm} (A.33)

For anisotropic materials, such as composite laminates, this quantity depends on the orientation. Nevertheless, by analogy with isotropic materials, an effective Poisson’s ratio \(\nu_{\text{eff}}\) has been defined for laminates:

\[ \nu_{\text{eff}} = \frac{\epsilon_{yy}}{\epsilon_{xx}} \]  \hspace{1cm} (A.34)

\[ \nu_{\text{eff}} = \frac{A_{12}}{A_{22}} \]  \hspace{1cm} (A.35)
A.2.2 Coefficient of thermal expansion

Temperature-related expansions in orthotropic materials are different in the longitudinal and in the transverse directions. This temperature-strain is expressed by

\[ \epsilon^T = \epsilon^M + \epsilon^{FT}, \]  

(A.36)

where \( \epsilon^T \) is the total observed strain, \( \epsilon^M \) the mechanical strain (induced by applied stresses), and \( \epsilon^{FT} \) is the stress-free thermal strain. For a given state of observed strain \( \epsilon^T \) and thermal strain \( \epsilon^{FT} \), the induced stresses are obtained from the material constitutive relation (A.2)

\[ \sigma = Q \left( \epsilon^T - \epsilon^{FT} \right), \]  

(A.37)

expressed in the material directions.

For a ply orientated at an arbitrary angle \( \theta_k \) with respect to a reference coordinate system, this equation becomes

\[ \sigma_{xy} = \bar{Q} \left( \epsilon_{xy}^T - T^{-1} \epsilon_{12}^{FT} \right), \]  

(A.38)

where the thermal strains in the material directions \( \epsilon_{12}^{FT} \) are given by

\[
\begin{pmatrix}
\epsilon_1^{FT} \\
\epsilon_2^{FT} \\
\gamma_{12}^{FT}
\end{pmatrix}
= 
\begin{pmatrix}
\alpha_1 \\
\alpha_2 \\
0
\end{pmatrix}
\Delta T.
\]  

(A.39)
Imposing the free load condition \( N = 0 \), we get

\[
N = \int_{-h/2}^{h/2} \sigma_{xy} \, dz \\
= \int_{-h/2}^{h/2} \bar{Q} \left( \epsilon_{xy}^T - T^\epsilon \epsilon_{12}^{FT} \right) \\
= 2 \sum_{k=1}^{n_p} \bar{Q} \left( \epsilon_{xy}^T - T^\epsilon \epsilon_{12}^{FT} \right) \\
= 0 . \tag{A.40}
\]

Hence

\[
\sum_{k=1}^{n_p} \bar{Q} \epsilon_{xy}^T = \sum_{k=1}^{n_p} \bar{Q} T^\epsilon \epsilon_{12}^{FT} \\
\tag{A.42}
\]

and, introducing the cinematic relations,

\[
A \epsilon_0^T + B \kappa_0^T = \sum_{k=1}^{n_p} \bar{Q} T^\epsilon \epsilon_{12}^{FT} = N^T , \tag{A.43}
\]

where \( N^T \) represents the thermal loads.

The effective coefficients of thermal expansion \( \bar{\alpha}_x = \epsilon_x / \Delta T \) and \( \bar{\alpha}_y = \epsilon_y / \Delta T \) for a symmetric laminate \( (B = 0) \) can then easily be obtained as

\[
\left\{ \begin{array}{c}
\bar{\alpha}_x \\
\bar{\alpha}_y
\end{array} \right\} = \frac{h}{2} [A]^{-1} \begin{bmatrix} K_1 + K_2 V_1^* \\ K_1 - K_2 V_1^* \end{bmatrix} , \tag{A.44}
\]

where the material constants \( K_1, K_2, \) and \( K_3 \) are defined by

\[
K_1 = (U_1 + U_4)(\alpha_1 + \alpha_2) + U_2(\alpha_1 - \alpha_2) \\
K_2 = U_2(\alpha_1 + \alpha_2) + (U_1 + 2U_3 - U_4)(\alpha_1 - \alpha_2) \\
K_3 = U_2(\alpha_1 + \alpha_2) + 2(U_3 + U_5)(\alpha_1 - \alpha_2) . \tag{A.45}
\]
A.2.3 First natural frequency

Consider a simply-supported laminated plate as illustrated in Figure A–3. In the classical lamination theory, assuming small displacements, the natural vibration frequencies are given by

\[ f_{mn} = \frac{\pi}{2\sqrt{\rho h}} \sqrt{D_{11} \left(\frac{m}{L}\right)^4 + 2(D_{12} + 2D_{66}) \left(\frac{m}{L}\right)^2 \left(\frac{n}{W}\right)^2 + D_{22} \left(\frac{n}{W}\right)^4}, \quad (A.46) \]

where \( L \) and \( W \) are the length and the width of the plate, and \( m \) and \( n \) designate the number of half waves in the \( x \) and \( y \) directions.

A.2.4 Strength

Let us consider a single ply under a strain state \( \epsilon_x, \epsilon_y, \gamma_{xy} \). The maximum strain criterion states that failure occurs when one of the strains in the material directions exceed the ultimate strains. The safe condition is expressed by:

\[ \epsilon_1 < \epsilon_1^t \quad (traction) \quad \text{or} \quad -\epsilon_1 < \epsilon_1^c \quad (compression) \]
\[ \epsilon_2 < \epsilon_2^t \quad (traction) \quad \text{or} \quad -\epsilon_2 < \epsilon_2^c \quad (compression) \]
\[ |\gamma_{12}| < \gamma^s \quad (A.47) \]
This can be condensed in the equation

$$\lambda_s = \min \left[ \max \left( \frac{\epsilon_1^f}{\epsilon_1}, -\epsilon_1 \right), \max \left( \frac{\epsilon_2^f}{\epsilon_2}, -\epsilon_2 \right), \frac{\gamma^s}{|\gamma_{12}|} \right] > 1 \quad (A.48)$$

The first-ply-failure (FPF) criterion considers that a laminate has failed as soon as one ply has failed. This is simply expressed by:

$$\min_{k=1,...,n_p} \lambda_s > 1 \quad (A.49)$$
B.1 Kernel density estimation

Suppose that a random variable $x$ has a probability density $p(x)$. The task of finding the best approximation to $p$ from a set of $N$ data points sampled from $p(x)$ is called density estimation. Two families of methods are available to address density estimation problems:

**parametric methods** supposed that the data originate from a known parametric family of density functions (for example a normal distribution with mean $\mu$ and standard deviation $\sigma$). The goal is to find the values of the parameters that best describe the data;

**non-parametric methods** make no (at least less stringent) assumptions about the structure of the distribution but let the data dictate its shape. Histograms are an example of non-parametric density estimates.

In the kernel density estimate (KDE), also known as Parzen windows estimate (Duda et al., 2000), the estimated density function $\hat{p}$ is obtained as the sum of symmetric functions, or kernels, centered around each data point $x_i$:

$$p(x) = \frac{1}{N} \sum_{i=1}^{N} K(x - x_i). \quad (B.1)$$
Many different kernels $K(u)$ can be used. The most common is the Gaussian kernel:

$$K(u) = \frac{1}{(2\pi)^{d/2}\sigma^d} \exp\left(-\frac{u^Tu}{\sigma^2}\right). \quad (B.2)$$

The parameter $\sigma$ is called the bandwidth. It determines the smoothness of the estimate: if $\sigma$ is set to a small value, the estimate $\hat{p}$ will exhibit spikes at the data points; on the contrary, if a large value of $\sigma$ is chosen, local variations of $p$ will be smoothed out, as shown in Figure B–1, where the true distribution $p$ is the sum of three normal distributions. The figure shows the estimate $\hat{p}$ based on a sample of 100 data points, for three values of $\sigma$. Clearly, a bandwidth of 0.3 provides the best approximation of $p$.

![Figure B–1: Effect of the bandwidth. With small values of $\sigma$, the estimate tends to overfit the data, with large values of $\sigma$, local variations of $p$ are smoothed out.](image)

While the value of the bandwidth can dramatically affect the accuracy of the estimate, determining the optimal setting is a non-trivial task when the underlying distribution is unknown. In fact, when no prior information about $p$ is available, no value of the bandwidth can be deemed better than others. The next section introduces the heuristic used in this work.
B.2 Determination of the bandwidth

B.2.1 Effect of $\sigma$ on the accuracy

A preliminary study of the influence of the value of $\sigma$ on the accuracy of $p(V)$ was conducted in the case $V = (V_1^*, V_3^*)$. We considered the problem of maximizing $A_{22}$ of a balanced symmetric laminate $[\pm \theta_1 / \pm \theta_2]_s$, with $\theta_k \in \{0^\circ, 5^\circ, 10^\circ, \ldots, 90^\circ\}$, subject to a constraint on Poisson’s ratio:

$$\text{maximize} \quad A_{22},$$

$$\text{such that} \quad 0.48 \leq \nu_{\text{eff}} \leq 0.52,$$

where the constraints were enforced using a penalty approach (cf. Section 4–7 for more details).

First, the exact distribution of selected points $p(V_1^*, V_3^*)$ was emulated by

1. computing the fitness of all possible combinations $[\theta_1/\theta_2]$;
2. assigning a selection probability $p_s(\theta_1, \theta_2)$ to each point of the design space, proportional to its rank based on the fitness;
3. generating a population of $N = 50,000$ points by sampling from $p_s(\theta_1, \theta_2)$ and computing the value of the auxiliary variables $V_1^*$ and $V_3^*$ of the created points;
4. estimating the distribution $p(V_1^*, V_3^*)$ of these “selected” points by a kernel density method, using a small value of the kernel$^1$. The best value of $\sigma$ is

$^1$ In theory, the approximation error for a continuous density $p(x)$ converges to zero when the number of sample points $N$ tends to infinity (Duda et al., 2000) $\sigma = 0.1$, and the bandwidth $\sigma$ tends to zero. In our problem, the cardinality of the space is finite, consequently, the bandwidth cannot be chosen arbitrarily small.
determined graphically. The “ideal” distribution \( p(V_1^*, V_3^*) \) obtained by this procedure is shown in Figure B–2.

![Figure B–2](image)

Figure B–2: Accurate estimate of the distribution \( p(V_1^*, V_3^*) \) of selected points for the constrained maximization of \( A_{22} \).

The influence of \( \sigma \) on the accuracy of \( \hat{p}_\sigma(V_1^*, V_3^*) \) was then investigated by calculating the average approximation error for seven values of the bandwidth: \( \sigma = 0.05, 0.1, 0.12, 0.15, 0.18, 0.2, \) and \( 0.25 \), and samples of \( \mu = 50 \) selected points. The error was estimated by the distance between the approximate distribution \( \hat{p}_\sigma(V_1^*, V_3^*) \) and the true distribution \( p(V_1^*, V_3^*) \):

\[
d_\sigma = \frac{1}{m} \sum_{i=1}^{m} |\hat{p}_\sigma(V_{1i}^*, V_{3i}^*) - p(V_{1i}^*, V_{3i}^*)|,
\]

where the \( m \) points \((V_{1i}^*, V_{3i}^*)\) are arranged according to a 50 by 50 grid in \([-1, 1]^2\). To obtain an estimate of the expected value of the error, the procedure is repeated 50 times with a different sample obtained by linear ranking selection among all possible points. The evolution of the average error \( \bar{d}_\sigma \) for \( \mu = 50 \) sample points is plotted in Figure B–3. The error is very large for small values of \( \sigma \), because the model provides
Figure B–3: Influence of the bandwidth $\sigma$ on the approximation error for a sample size of $\mu = 50$ points.

Poor estimates of $p(V_1^*, V_3^*)$ at points that were not used to build the approximation.

Then, the error remains stable in a wide range between 0.12 and 0.2, before increasing for large values, because the model then fails to capture local variations of the density.

An example of approximation obtained with the best value of the bandwidth $\sigma = 0.18$ and $\mu = 50$ is shown in Figure B–4.

Figure B–4: Example of an approximation $\hat{p}_\sigma(V_1^*, V_3^*)$ obtained for $\mu = 50$ points and $\sigma = 0.18$. 
B.2.2 Maximum likelihood method

One the most common parameter estimation methods used in parametric density estimate is the maximum likelihood method, where one seeks to maximize the likelihood of the data by adjusting the value of the model parameters $\pi$. The likelihood is defined as:

$$L(\pi) = \prod_{i=1}^{N} \hat{p}(x_i; \pi).$$

(B.5)

This method cannot be directly applied to kernel density estimation, as the method would lead to the trivial solution $\sigma \to 0$ (Dirac delta at each data point). To improve the generalization performance of KDE, a cross-validation method described by Turlach (1993) was implemented. The method is a modified maximum likelihood approach, where the likelihood is replaced by the pseudo-likelihood:

$$\tilde{L}(\sigma) = \prod_{i=1}^{\mu} p_{\sigma, i}(V_i),$$

(B.6)

where $p_{\sigma, i}$ is a leave-one-out estimation of the density at $V_i$, based on the $\mu - 1$ other data points:

$$p_{\sigma, i} = \sum_{j=1, j \neq i}^{\mu} K(V - V_i).$$

(B.7)

B.3 Application in the context of DDOA

The choice of the bandwidth $\sigma$ is essential for a good estimation of $p(V)$ and hence for the efficiency of DDOA. An upper bound$^2$ of the “optimal” bandwidth

\[2\text{ Since the population converges as the optimization progresses, the variability will decrease, and so will } \sigma.\]
can be found by obtaining the value of $\sigma$ that provides the best estimation of the distribution of selected points at the first generation.

The bandwidth $\sigma$ was allowed to take values from $\{0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.12, 0.15, 0.2, 0.3, 0.4\}$. To reduce the variance of the likelihood estimation, the total likelihood $L^{\text{all}}$ over 100 independent runs was used as a measure of the goodness of the density estimation, for each value of $\sigma$:

$$L^{\text{all}} = \prod_{r=1}^{100} L_r(\sigma),$$

where $L_r(\sigma)$ is the likelihood obtained for the $r^{\text{th}}$ run. Given the small sample size of $\mu = 30$ and the sample creation procedure ($\mu = 30$ points are selected out of $\lambda = 30$ points by linear ranking selection), the set of selected points is highly likely to contain duplicates of the same individual, hence compromising the validity of the cross-validation. Therefore, the contribution of $p_{\sigma,i}(V_i)$ at points that had been used to construct the approximation was ignored. The procedure was applied for the first two optimization problems treated in Chapter 4: the constrained maximization of $A_{22}$ and the constrained minimization of the longitudinal CTE, for $n = 6$ and $n = 12$. The log likelihood as a function of $\sigma$ is shown in Figure B-5 for the constrained maximization of $A_{22}$ problem, for the case $n = 6$ (the other cases would display the same trend): it is characterized by a very low likelihood for small values of $\sigma$ ($\sigma < 0.1$), followed by a relatively flat region. This indicates that when too small kernels are used, the predictive power of the approximation is compromised, while the quality of the estimation is not very sensitive to the choice of $\sigma$ beyond $\sigma = 0.2$.

For the two problems, the best bandwidth was $\sigma^* = 0.2$ for $n = 6$ and $\sigma^* = 0.15$
for \( n = 12 \). This trend parallels the evolution of the average error observed in the

![Log likelihood plot](image)

Figure B–5: Log likelihood \( L^{\text{all}} \) as a function of \( \sigma \) for \( n = 6 \), \( \mu = 30 \), constrained maximization of \( A_{22} \)

previous section, thus confirming the validity of the method for bandwidth selection.

### B.4 Effect of the bandwidth: empirical results

The influence of the choice of the bandwidth \( \sigma \) was investigated for the first optimization problems addressed in Section 4.4: the constrained maximization of the transverse in-plane stiffness \( A_{22} \). Four values of the bandwidth were tried: \( \sigma = 0.1 \), \( \sigma = 0.15 \), \( \sigma = 0.2 \), and \( \sigma = 0.3 \).

When integrated into the full algorithm, the kernel density estimate component interacts with other parts of the algorithm, consequently the best setting of the bandwidth may differ from the “off-line” setting yielded by the maximum likelihood method. It is therefore necessary to take into consideration the way the estimate is used by the algorithm to create new points. To this aim, two target point creation schemes were tested, in association with the different values of \( \sigma \): a rejection method, where the feasibility of target points was enforced by sampling from a particular kernel until a feasible point was created (the domain of all the possible values of the
lamination parameters is defined by trigonometric relationships, cf. Appendix A), and a more forgiving method that allowed infeasible points to be used as targets. Figure B–6 shows the mean best fitness for all the variants tested, for $n = 6$, and $n = 12$. Surprisingly enough, neither the target point creation scheme nor the bandwidth of

Figure B–6: Effect of the target point creation procedure on the best fitness

the kernel density estimate significantly influenced the algorithm’s performance: in the case $n = 6$, all but three DDOA variants (“reject”, $\sigma = 0.2$ and $\sigma = 0.3$, and “no reject”, $\sigma = 0.15$) outperformed, or had comparable performance to GA and UMDA; and in the case $n = 12$, all variants exhibited a clear advantage over GA and UMDA. Only one variant, “reject” associated with a bandwidth of 0.3 failed to consistently converge to high fitness individuals. This observation confirms the results of the maximum likelihood bandwidth determination, which showed a stable region between $\sigma = 0.1$ and $\sigma = 0.2$.

Even though the two target point creation schemes displayed comparable performance, it appears that the mean fitness of the “no reject” scheme is more stable with
respect to the choice of $\sigma$. This was particularly visible for $n = 12$: in contrast to the “no reject” schemes which exhibited no significant difference, the performance of the “reject” scheme tended to deteriorate for larger values of $\sigma$. This can be explained by the fact that the “reject” scheme redistributes the contribution of the infeasible portion of the kernel over all the feasible domain, whereas the “no reject” scheme amounts to assigning all that contribution to the boundary, as sketched in Figure B–7. Considering that optimum laminates often lie near the boundary of the feasible domain, and that reject situations mainly occur when the optimum is close to the boundary, it seems sensible to favor peripheral regions over the rest of the design space, hence the “no reject” approach seems to be appropriate. On the contrary, the overhead associated with the “reject” scheme is not only costly, but detrimental to the algorithm’s performance.

![Figure B–7: Influence of the boundaries on the search distribution: “reject” and “no reject” target point creation schemes when $x = 5$ is the boundary of the feasible domain.](image-url)
APPENDIX C
LAMINATION PARAMETER PROPORTIONAL ACCEPTANCE: EXPERIMENTAL RESULTS

The influence of the procedure implemented to sample from the two distributions used in DDOA, the univariate distribution of the primary variables $p(x) = p(x_1)p(x_2) \ldots p(x_n)$, and the distribution of auxiliary variables $p(V_1^*, V_3^*)$ was investigated for the following laminate optimization problem:

$$\text{maximize } A_{22} = h(U_1 - U_2 V_1^* + U_3 V_3^*)$$

such that $\nu_l \leq \nu_{\text{eff}} \leq \nu_u$ (C.1)

in the case $n = 6$, with the material properties given in Table 3–3 p. 68.

The auxiliary variable probability proportional selection (cf. Section 4.2.2) is compared to the target point approach (no rejection scheme). Figure C–1 compares the mean best fitness of DDOA with probability proportional selection for four value of the pool size $\nu$ and with the target point approach ($\nu = 150$) to mean fitness of UMDA (a bandwidth of $\sigma = 0.2$ was used for the two DDOA variants). The best performance is achieved by DDOA with the target point approach. With the probability proportional point creation method, DDOA displays a poor performance, regardless of the value of $\nu$: in this DDOA implementation, the lamination parameter distribution slows down the optimization with respect to the basic UMDA algorithm.
Figure C–1: Lamination parameter probability proportional acceptance causes DDOA to be very conservative, thus resulting in slower convergence to the optimum than UMDA.
APPENDIX D
MULTIOBJECTIVE OPTIMIZATION OF LAMINATES

The attractive qualities of composite materials are their very high stiffness-to-weight or strength-to-weight ratios. However, the bulk and processing cost of these materials increases dramatically with performance. Although in aerospace applications, design is driven by structural weight considerations, cost becomes a factor in applications intended for the general public, such as automobiles. In such cases, it may be advantageous to resort to combinations of efficient but expensive, and less expensive but less stiff materials in order to reduce cost while ensuring high performance. Combining the two materials makes cost savings possible without compromising performance significantly. Finding the best combinations requires the trade-off between performance and cost to be carefully studied. This can be done in the framework of multiobjective optimization. The present chapter investigates the combination of two materials for balancing cost and weight. The methodology is demonstrated on a simple laminate optimization problem.

D.1 Problem description

A simply supported laminated plate of length $a = 36$ in and width $b = 30$ in is to be optimized for minimum weight, $W$, and cost, $C$, subject to a lower bound $f_{\text{min}}$ of 25 Hz on the first natural frequency $f$. A compromise solution between those two competing objectives can be achieved by combining two materials: graphite-epoxy and glass-epoxy. The material properties are given in Table
The stiffness-to-weight ratio of graphite-epoxy is about four times higher than that of glass-epoxy, with $E_1/\rho = 345$ against $E_1/\rho = 87.5$. However it is also more expensive, with a cost per pound that is about 8 times higher than that of glass-epoxy. If the first priority is weight, then graphite-epoxy will be preferred; while if cost is paramount the optimum laminate will obviously contain glass-epoxy plies. The design of this simple rectangular plate leads us to study the trade-off between the two objective functions: weight and cost.

The ply orientation can take on a set of 19 values ranging from 0 to 90 degrees in steps of 5 degrees. Hence, the problem consists in finding the number of plies and the stacking sequence of the laminate, which is the orientation and material of each ply. The laminate is symmetric and balanced (for every angle $\theta$ different from 0° and 90°, the angle $-\theta$ is also present in the laminate). These constraints are imposed to ensure the manufacturability of the structure and are usual in laminate design. They simplify the analysis considerably, as the symmetry assumption implies that the extensional-flexural coupling matrix $B$ of the classical lamination theory is zero and the balance constraint suppresses the normal-shear extensional coupling ($A_{16} = 0$ and $A_{26} = 0$) and minimizes the magnitude of the torsion-bending coupling terms $D_{16}$ and $D_{26}$.

The multiobjective optimization problem can be formulated as follows:

\begin{equation}
\text{Minimize:} \quad W \text{ and } C \\
\text{by changing:} \quad \text{the orientation } \theta_i \text{ and the material } m_i \text{ of the plies} \quad \text{(D.1)}
\end{equation}

\begin{equation}
\text{such that:} \quad g = \frac{f}{f_{\text{min}}} - 1 \geq 0
\end{equation}
Table D–1: Material properties of graphite-epoxy and glass-epoxy

<table>
<thead>
<tr>
<th></th>
<th>Graphite-epoxy</th>
<th>Glass-epoxy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longitudinal modulus</td>
<td>20.01</td>
<td>6.3</td>
</tr>
<tr>
<td>(Msi), $E_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transverse modulus</td>
<td>1.30</td>
<td>1.29</td>
</tr>
<tr>
<td>(Msi), $E_2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>In-plane shear modulus</td>
<td>1.03</td>
<td>0.66</td>
</tr>
<tr>
<td>(Msi), $G_{12}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Poisson modulus, $\nu_{12}$</td>
<td>0.3</td>
<td>0.27</td>
</tr>
<tr>
<td>Density (lb/in³), $\rho$</td>
<td>0.058</td>
<td>0.072</td>
</tr>
<tr>
<td>Thickness (in), $t$</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>Cost factor (lb⁻¹), $c$</td>
<td>8.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Using classical lamination theory, the first natural vibration frequency is given by the following expression:

$$f = \frac{\pi}{2\sqrt{\rho h}} \sqrt{\frac{D_{11}}{a^4} + 2\frac{D_{12} + 2D_{06}}{a^2b^2} + \frac{D_{22}}{b^4}}$$

(D.2)

where $D$ is the flexural stiffness matrix, $\rho$ is the average density and $h$ the total thickness of the plate.

D.2 Introduction to multiobjective optimization

In multiobjective optimization, $m$ objective functions $F_1(x)$ to $F_m(x)$ are considered simultaneously. In general, it is not possible to find a point $x$ that minimizes all objectives because some objectives conflict with one another. Instead, we seek points where one cannot improve one objective without causing another objective to deteriorate. Such points are called non-dominated because one cannot find points that outperform them in all objective functions. Non-dominated points are said to be Pareto-optimal. That is, a point $x^*$ is Pareto-optimal if and only if there is no point $x$ with the characteristics:

$$F_i(x) \leq F_i(x^*) \quad \text{for all } i, i = 1, 2, \ldots, m$$
and

\[ F_i(x) < F_i(x^*) \quad \text{for at least one } i, \; 1 \leq i \leq m \]

Pareto-optimal points make up the Pareto set. Multiobjective optimization is concerned with characterizing the Pareto set. From a mathematical viewpoint, no point in the Pareto set is better than the others. It is the role of the designer to interpret the Pareto surface in order to choose between the elements of the Pareto set.

Many methods have been developed to generate the Pareto set (Deb, 1999). The simplest approaches transform the multi-objective problem into a single-objective optimization. One approach, termed min-max method, consists in finding the so-called “ideal point” \( F^* \), which is made of the optimum value of the individual objectives when optimized separately and looking for the design whose distance to that ideal point in the objective space is minimum. That method was employed in laminate optimization by Saravanos and Chamis (1992) to solve the problem of minimizing resonance amplitude \( F_1 \), structural weight \( F_2 \) and cost \( F_3 \) of a cantilever graphite-epoxy beam by changing the fiber volume fraction, ply thicknesses and ply angles. Another way of reducing the multiobjective problem to a single-objective one is to combine all objectives into a single composite function \( F(x) = \alpha_1 F_1(x) + \alpha_2 F_2(x) + \ldots + \alpha_m F_m(x) \). The weighting factors allow the design to prioritize the objectives. However, both of these methods give only one point of the Pareto set. In situations where the entire set is needed, other methods must be used. A popular method, known as weighting
method, consists in repeating the single-objective optimization of the composite function just described using different a combination of the weighting factors each time, thus obtaining a different Pareto-optimal point every time. The constraint approach keeps one objective function and reformulates all other objectives as constraints. By changing the weighting coefficients of $F$ or the constraint limits, the whole Pareto set can be found. Alternatively, there are evolutionary algorithms that approximate the Pareto set in a single optimization. For example, Hajela and Lin (1992) investigated such genetic strategies in the context of structural optimization.

In laminate optimization, several approaches have been taken to approximate the Pareto surface. Adali et al. (1996) used a weighting method to solve the three-objective problem of maximizing prebuckling, buckling and postbuckling strength of a laminated plate. Kumar and Tauchert (1992) observed that non-convex Pareto front can exist in the case of maximum in-plane or bending strength or stiffness maximization, and therefore recommended using the weighting min-max method, which is a modified min-max algorithm where the contribution of each objective to the distance between a particular design and the ideal point is given a different weight.

In this work the Pareto set is generated by optimizing a convex combination of the two objectives, weight $W$ and cost $C$ into a composite function $F$:

$$F = \alpha C + (1 - \alpha)W$$  \hspace{1cm} (D.3)

for $0 \leq \alpha \leq 1$.

The trade-off curve between weight and cost is then constructed by performing
a succession of the following single-objective optimizations for several values of $\alpha$:

\begin{align*}
\text{Minimize:} & \quad F \\
\text{by changing:} & \quad \text{the orientation } \theta_i \text{ and the material } m_i \text{ of the plies} \quad (D.4) \\
\text{such that:} & \quad g = \frac{f}{f_{\text{min}}} \geq 0
\end{align*}

A well-known limitation of the method is that it fails to capture concave portions of the Pareto set. Due care will be taken when interpreting the results.

### D.3 Genetic algorithm

Genetic algorithms (GA) are inspired by Darwin’s principle of evolution, which states that a population of individuals is capable of adapting to its environment because individuals that possess traits that make them less vulnerable than others are more likely to have descendents and therefore to pass on their desirable traits to the next generation. One can think of this process of adaptation as an optimization process that probabilistically creates fitter individuals through selection and recombination of good characters. Genetic algorithms are simplified computer models of evolution, where the environment is emulated by the objective function to maximize, and the structure to optimize plays the role of the individuals.

A flowchart of a genetic algorithm is presented in Figure D–1. GAs start by initializing a population of individuals at random. Each individual encode of a particular candidate structure in the form of one or several chromosomes, which are strings of finite length. Then the objective function of each individual, called fitness function in the context of evolutionary computation, is computed. The fitness of each
individual determines its probability of being selected for reproduction. Recombination and mutation operators are then applied to the selected individuals (the parents) to create a population of children. Finally a survival rule determines the individuals among the parent and child population that will be kept to form the new population.

![Genetic algorithm diagram]

Figure D–1: Genetic algorithm

D.3.1 Laminate encoding

In the problem under consideration, the design variables are the number of plies, the material and the fiber orientation of each ply. The plate dimensions are fixed. Hence a laminate can be represented by two strings (chromosomes), one for the fiber orientation, the other for the material where each character (gene) describes the orientation or material of a particular ply. Variable thickness is achieved by
allowing empty plies. Since only balanced symmetric laminates are acceptable, the number of genes required to represent a laminate is only about one quarter of the number of plies, because only half of the laminate needs to be encoded and balance is enforced by using pairs of plies with opposite angles. Since the balance constraint does not affect 0° and 90° plies when loadings with respect to the laminate coordinates are considered, only these angles are allowed to appear as individual plies, thereby suppressing unnecessary layers and providing more flexibility in the design.

The 19 possible angles (from 0° to 90° in 5° steps), are simply coded as integers between 1 and 19, with 1 corresponding to 0° and 19 to 90°. Empty plies were denoted by E. Similarly, the ply material was coded by two integers: 1 for graphite-epoxy and 2 for glass-epoxy.

As an example, the individual:

Orientation \[4/7/12/1/3/E/E\]

Material \[1/2/2/1/1/E/E\]

will be decoded as: \[±15°/±30°/±55°/0°/±10°\], where normal characters denote graphite-epoxy and bold characters denote glass-epoxy.

**D.3.2 Fitness function**

Genetic algorithms discriminate between promising and poor individuals by comparing all members of a population on the basis of a scalar score, called fitness function, which summarizes the individual performances. Genetic algorithms do not naturally accommodate constraints, however several heuristics can be used to direct the search toward feasible regions. The most effective and inexpensive way of enforcing constraints is to use data structuring, i.e. to hard-code the constraint into the
chromosomes. For example, in this work, symmetry and balance constraints are automatically met because the decoding produces only symmetric balanced laminates. However, most constraints cannot be easily hard-coded into the chromosomes. For such constraints, two main techniques: repair and penalty approaches are available (see Michalewicz et al. (1996)). Repair strategies use heuristics to replace an individual that violates constraints with its “projection” onto the constraints, so that its genetic content is not lost altogether. Repair strategies have proven effective on many problems. For instance, Todoroki and Haftka (1998) used a phenotype repair strategy to enforce balance and contiguity constraints. However, there is no standard repair method, and repairing an individual can be very difficult when the constraint is a complex function of the design variables. In our case, the frequency constraint falls into that category; consequently repair is not practical. We instead use a penalty approach, where the fitness of individuals that violate constraints is decreased to account for the cost of satisfying the constraint. In this work, we adopted the following formulation for the fitness function $F$:

$$F = \begin{cases} 
- F(1 - \epsilon g) & \text{if } g \geq 0 \quad \text{(feasible)} \\
-1.1 F + pg & \text{otherwise} \quad \text{(infeasible)}
\end{cases}$$

(D.5)

where $p$ denotes the penalty parameter, which is selected through experiments to be high enough to ensure that the design with the lowest value of $F$ does not violate the constraint. The multiplier $\epsilon$ (taken to be 0.01) in the expression for feasible design is used to reward designs that satisfy the constraint with larger margins.
D.3.3 Genetic operators

Three types of operators are used to produce new individuals from a parent population: selection, recombination and mutation. The role of selection is to identify individuals that potentially contain partial solutions (building blocks), recombination mixes genetic material from the parents to generate children with the hope that good traits from all parents get combined. However, if only these two operators are used, the population loses genetic diversity, and once the population has become uniform, no more progress is possible. To prevent premature convergence, a perturbation operator called mutation is used to add a random exploration component to the algorithm.

Selection. One of the key factors of evolutionary algorithms is selection. In this work, rank proportional selection was used. The main advantage of rank based selection is that it is insensitive to the objective function scaling. It allows the selection pressure to remain constant throughout the optimization, in contrast to fitness proportional selection, which loses its discriminating capacity when all individuals become close to the optimum.

Crossover. The role of recombination, or crossover, is to exchange genetic material between two parents to create two children who combine traits from both parents. The most common forms of crossover operators are uniform crossover, one-point and two-point crossover. They differ in the amount of gene mixing they accomplish. In uniform crossover, each gene of the first child is chosen with equal probability from either parent, the second child receives the gene from the other parent. In one-point
crossover, a break point is chosen randomly in the chromosomes, and the offsprings are created by swapping the parents’ substrings. Two-point crossover chooses two break points at random, and swaps the substring limited by these two points, as depicted in Figure D–2.

| Parent 1: 4\ 7 12 1 3\ 10 | Child 1: 4 7 12 1 3 7 3 |
| Parent 2: 4\ 16 9 3 5\ 3 | Child 2: 4 16 9 3 5 2 10 |

Figure D–2: Two-point crossover

In this work, the two-point crossover was used because unlike one-point and uniform crossovers, it does not break linkage between inner and outer plies. It “respects” cooperation between these plies. It should be noted that outermost plies are preponderant in the bending behavior of the plate. Mixing occurs mainly within mid-range plies.

In the case of simultaneous optimization of the ply angles and materials, there are two options for applying crossover: the two chromosomes can be treated separately (distinct break points) or together. Extensive testing conducted prior to this study showed that the former option was more efficient.

**Mutations.** Traditionally, mutation operators are viewed as background operator in the genetic algorithm community. However, mutation, in a broad sense is often an invaluable tool for exploring new regions of the design space and prevent premature convergence. When mutation operators are specifically tailored to a class of problems, considerable time savings can be achieved. Three types of mutation were used in this work.
**Simple mutation**  A standard mutation operator is applied to both chromosomes of every child individual with some user-specified probabilities \( p^o_m \) (orientation) and \( p^m_m \) (material). Mutation chooses a gene at random and switches its value to any other possible value, as illustrated in Figure D–3.

<table>
<thead>
<tr>
<th>Before mutation: 4 7 12 1 3 2 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>After mutation: 4 7 3 1 3 2 10</td>
</tr>
</tbody>
</table>

Figure D–3: Simple mutation

**Permutation**  The permutation operator was introduced by Le Riche and Haftka (1993) to solve stacking sequence optimization problems. It is especially useful because it changes the bending properties of a laminate without affecting its in-plane properties. The permutation operator, applied with probability \( p_p \) to a chromosome, chooses two genes at random in that chromosome and flips the order of the substring contained inside these two genes (see Figure D–4).

<table>
<thead>
<tr>
<th>Before permutation: 4 7 12 1 3 2 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>After permutation: 4 1 12 7 3 2 10</td>
</tr>
</tbody>
</table>

Figure D–4: Permutation

**Addition and deletion**  Variable number of plies was accomplished by allowing the genes to take a special value \( E \), which denotes an empty ply. That value can only be set or lifted by dedicated mutation operators called addition and deletion, which are applied with user-defined probabilities \( p_a \) and \( p_d \). Addition and deletion operators are illustrated in Figure D–5.
### Addition

<table>
<thead>
<tr>
<th>Before addition:</th>
<th>After addition:</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 7 12 1 3 E</td>
<td>4 7 12 17 1 3 E</td>
</tr>
</tbody>
</table>

### Deletion

<table>
<thead>
<tr>
<th>Before deletion:</th>
<th>After deletion:</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 7 12 1 3 7 E</td>
<td>4 7 1 3 7 E</td>
</tr>
</tbody>
</table>

Figure D–5: Addition and deletion

### D.3.4 Implementation

A genetic algorithm for multiobjective optimization was developed using the GA template for laminate optimization developed by McMahon et al. (1998). That package provides a framework for building algorithms that can tackle a wide variety of composite structure design situations, including multiple materials, multiple laminates, continuous and discrete variables for stacking sequence and geometric variables. It is comprised of a GA module and a package of GA operators. The GA module defines a data structure that facilitates manipulation of populations or individuals in an object-oriented approach. The program was modified to incorporate the two-point crossover applied independently to the orientation and material chromosomes. The decoding procedure was modified to implement the balance constraint as described previously (pairs of angles of opposite signs are used except for $0^\circ$ and $90^\circ$).

### D.4 Results

First the minimum of the weight and cost objective functions were found using the genetic algorithm. The optimization started with a population of 10 designs made of 44 plies (eleven genes) whose orientation and material were set randomly. The GA was applied with the parameters shown in Table D–2 (all probabilities are probabilities of application per chromosome). These values were determined by trial and error in order to maximize reliability, which is the probability of reaching the
optimum for a given number of function evaluations. They are used in all cases in this study.

Table D–2: Parameters associated with the genetic operators

<table>
<thead>
<tr>
<th>Crossover probability</th>
<th>Mutation probability, orientation</th>
<th>Mutation probability, materials</th>
<th>Permutation probability</th>
<th>Addition probability</th>
<th>Deletion probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.3</td>
<td>0.2</td>
<td>0.2</td>
<td>0.05</td>
<td>0.1</td>
</tr>
</tbody>
</table>

The minimum weight \((\alpha = 1.0)\) was found to be 6.89 lb. It is reached when all plies are made of graphite-epoxy. The optimum design is then \([\pm 50_5/0]_s\), and has a relative cost of 55.12. The minimum cost \((\alpha = 0.0)\) was 16.33, 70% less expensive than the optimum of the weight objective function. The optimum design is \([\pm 50_{10}/0]_s\) comprised exclusively of glass-epoxy. Its weight is 16.33 lb, which represents an increase of 137% over the optimum of the weight objective function. In order to construct the Pareto front, the weighting factor \(a\) was varied from 0.0 to 1.0 and the composite objective function \(F\) was minimized using the GA. The optimum designs obtained as well as their weight and cost are summarized in Table D–3. The optimum designs are mainly made of \(\pm 50^\circ\) plies in order to maximize the first vibration frequency of the plate. In some cases a \(0^\circ\) or \(90^\circ\) ply is present in the core layers of the laminate. Although these plies do not contribute much to the frequency, it is advantageous to use them because unlike other angles they do not have to come in pairs (to satisfy balance), thereby saving unnecessary additional weight and cost. When a single material is used, the \(0^\circ\) and \(90^\circ\) plies always appear in the inner layers, where they are the least damaging for the performance of the plate. When two materials are used, they are placed in the innermost layer of each material block.
For the same reason, the less stiff glass-epoxy layers, when they have to be used, always appear in the inner layers. This creates the sandwich type composite shown in Figure D–6 where the structural function is assured by the stiff graphite layers, placed on the outside, where their contribution to the flexural properties of the laminate is maximal, while inner layers are merely used to increase the distance of the outer plies from the neutral plane.

Table D–3: Optimum designs of the minimization of the composite objective function

<table>
<thead>
<tr>
<th>Weighting factor $\alpha$</th>
<th>Cost</th>
<th>Weight</th>
<th>First natural frequency ($f_{\text{min}} = 25$ Hz)</th>
<th>Stacking sequence (plain numbers: graphite, boldfaced numbers: glass)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00</td>
<td>16.33</td>
<td>16.33</td>
<td>$\pm 50_{10}/0$$_s$</td>
</tr>
<tr>
<td>2</td>
<td>0.70</td>
<td>20.90</td>
<td>12.14</td>
<td>$\pm 50_{/ / \pm 50_{1}}$$_s$</td>
</tr>
<tr>
<td>3</td>
<td>0.80</td>
<td>27.82</td>
<td>10.28</td>
<td>$\pm 50_{2}/ \pm 50_{5}$$_s$</td>
</tr>
<tr>
<td>4</td>
<td>0.87</td>
<td>31.28</td>
<td>9.35</td>
<td>$\pm 50_{2}/90/ \pm 50_{4}$$_s$</td>
</tr>
<tr>
<td>5</td>
<td>0.90</td>
<td>38.96</td>
<td>8.27</td>
<td>$\pm 50_{3}/90/ \pm 50_{2}/0$$_s$</td>
</tr>
<tr>
<td>6</td>
<td>0.96</td>
<td>43.20</td>
<td>8.12</td>
<td>$\pm 50_{4}/ \pm 50_{2}$$_s$</td>
</tr>
<tr>
<td>7</td>
<td>1.00</td>
<td>55.12</td>
<td>6.89</td>
<td>$\pm 50_{5}/0$$_s$</td>
</tr>
</tbody>
</table>

Figure D–6: Sandwich structure of the optimum design 2 (Table D–3). The structural function is mainly assured by the graphite-epoxy layers placed on the outside, while the role of the inner glass-epoxy layers is to increase the distance of the outer plies to the neutral plane.

Design 5 is particularly instructive because it reveals how the GA was able to achieve cost savings while preserving performance. It exhibits the sandwich type structure characterized by stiffer graphite-epoxy plies on the outer layers, but inside
each layer weight and stiffness are traded by using the ±50° stacks where they are the most effective and padding with individual 0° and 90° plies that give more flexibility in the design.

Figure D–7 shows the Pareto front obtained using the weighed sum method. The solid black line represents the set of solutions of the composite objective function for the different values of α. The various symbols show all the feasible designs that were generated during the search. The Pareto front (the set of all the non-dominated solutions) corresponds to the lower left envelope of all the design points in the weight/cost plane. Out of the sampled feasible points, the seven designs making up the Pareto front have been found by the weighted sum approach.

A peculiarity of the current Pareto curve is its concavity at Point 6. In theory, the weighting method used to generate it cannot find such points, as illustrated in Figure D–8. The dashed lines indicate the contours of the composite function F in the (C, W)-plane, with values decreasing as one gets closer to the origin. A conceptual concave Pareto front is represented by the solid line. When the factor α is varied from 0 to 1 (the orientation of the dashed lines goes from vertical to horizontal), the concave part of the curve never yields the minimum of F. Only points on the convex hull of the Pareto set can be determined. Points 1, 2, 4 and 5 can be found by this method, but not point 6.

However, for α = 0.90, the algorithm alternatively yielded design 5 or design 6. In fact, out of a total of 50 runs, it converged 29 times to design 5 and 21 times to design 6. The true optimum is design 5, which gives $F = 11.34$, whereas the design 6 evaluates at $F = 11.63$, but design 5 is substantially more difficult to find because it
Figure D–7: Pareto curve for the minimum weight and cost design of the laminated plate. The solid black curve joins non-dominated points.

is located in a deep valley: all of its neighbors are either substantially heavier because the $0^\circ$ and $90^\circ$ plies are not present, or violate the frequency constraint. In contrast, design 6 is extremely robust because the single-ply stacks are not used and the margin of safety for the frequency is large. In this case, convergence to a local optimum is desirable, as it reveals concave parts of the Pareto set.

During the approximation of the Pareto set, we found that it was critical to adjust the penalty parameter $p$ in order to find all the non-dominated designs. When $p$ was too high, the search was biased toward non optimal thicker laminates, whereas too
Figure D–8: The weighting method is not appropriate to capture concave Pareto front in general. Points 1, 2, 4 and 5 can be found by this method, but not point 3.

Low values of $p$ cause the laminate to collapse to zero thickness laminates because the structural requirement was dominated by the weight/cost considerations. Since the penalty must be scaled to the magnitude of the objective function, which depended on $\alpha$, a good value of $p$ had to be determined by trial and error for each of its values. We found that a good value of $p$ was 15.0 for $\alpha = 0.0$ and 8.0 for $\alpha = 1.0$. The formulation of the weighting method can be perfected by normalizing the two objectives, so that the magnitude of the composite function does not change. This can be accomplished by dividing each objective by its maximum value:

$$\hat{F} = \hat{\alpha} \frac{W}{W_{\text{max}}} + (1 - \hat{\alpha}) \frac{C}{C_{\text{max}}}$$

The Pareto trade-off curve can be used to help the designer determine the optimal configuration for his problem. The final choice of the best design will depend on additional information that will enable him to assign priorities to the two objectives.
There is no single best design: depending on the application that is considered, the choice will be different. For example, if weight and cost are of similar importance, the design 5 obtained with $\alpha = 0.93$ in Table 3 may be attractive. The weight is 20% above the minimum weight, but the cost is significantly reduced by 30%.

D.5 Concluding remarks

The method used in this chapter for solving the multi-objective optimization problem that derives from multi-material composite design is simple. It is based on a standard genetic algorithm for composite laminate optimization. The Pareto-optimal front was constructed by optimizing a series of composite objective function linearly combining weight and cost. The method was shown to be able to capture important feature of the Pareto trade-off curve for that two-objective problem, providing the designer with a helpful decision-making tool. An example of a plate designed to minimize weight and cost subject to a constraint on the first natural frequency illustrated the approach. When cost was a primary consideration the plate was made from glass-epoxy, and when weight was a primary consideration, it was made of graphite-epoxy. Compromise designs are easily selected from the Pareto trade-off curve. In particular, the study revealed that substantial cost savings can be achieved with only a moderate weight increase by replacing some of the expensive graphite-epoxy inner layers by glass-epoxy.

Additional work remains to be done in order to improve the efficiency of the optimization. Performance may be marginally increased by changing the weighting method and penalty function formulations in order to improve the robustness of
the scheme against changes in the weighting factor $\alpha$. Preliminary investigation indicates that normalizing the objective functions can significantly reduce the amount of penalty parameter tuning needed and distribute the non-dominated points more uniformly in terms of the parameter $\alpha$.

The weighting method applied in this work requires multiple number of runs and can be computationally expensive. There exist multiobjective evolutionary algorithms (MOEA) approximate the Pareto set in one run by using Pareto dominance in the selection and a diversity preserving scheme in the objective space. Another way of reducing the number of analyses required is to improve the efficiency of each individual single objective optimization. Our purpose is to develop a generic algorithm that can be used for multiobjective optimization, but for single objective optimization as well. The main chapters of this dissertation are concerned with the investigation of efficient single objective laminate optimization methods.
BIOGRAPHICAL SKETCH

Laurent Grosset was born in Combourg, France, on May 3, 1977. He received a Bachelor of Engineering degree in mechanical engineering from the Institute of Applied Sciences of Rouen, France, 2000. During his engineering studies, he carried out several training periods in industry, in particular a three-month internship in the mechanical testing department of DaimlerChrysler in Stuttgart, Germany, and a six-month internship in the steam turbine design department at Hitachi Works, in Japan.