

STATISTICAL OPTIMIZATION OF COMPOSITE LAMINATES: INTRODUCTION OF COUPLING VIA A CHANGE OF VARIABLES

Laurent Grosset^{1,2} (AIAA member), Rodolphe Le Riche¹ and Raphael T. Haftka² (AIAA fellow)

¹ CNRS URA 1884 / SMS, Ecole des Mines de Saint Etienne, France.

² Dept. of Aerospace eng., Univ. of Florida, Gainesville, USA.

Abstract

The purpose of this paper is to improve the efficiency of a statistical optimization algorithm similar to Baluja's PBIL in two simple ways. First, the influence of the selection method is investigated and a selection scheme is proposed. Then a strategy for adding coupling in the probabilistic model at low computational cost is proposed. The idea is to use statistical information on intermediate variables (the lamination parameters of composite models) to complement the gene (fiber angle) probability distribution. The algorithms are applied to a composite laminate design.

Introduction

Evolutionary algorithms have been applied with great success to a variety of problems in many fields. In particular they have become one of the preferred methods for composite laminate optimization in the last decade because they are well suited for discrete, multi-modal problems. However the efficiency of standard evolutionary algorithms deteriorates when they are applied to problems that present strong interactions between variables. In that case, recombination operators often disrupt partial solutions and the good features of parent individuals are no longer propagated to the next generation. This observation prompted the emergence of new types of algorithms that do not break up groups of genes that are correlated. Some researchers developed "intelligent" recombination operators that take advantage of the structure of the problem (Gene Expression Messy Genetic Algorithm), [6]). Another branch of algorithms that abandon the reference to biologic evolution was developed following the effort of researchers, such as Baluja [1], Mühlenbein [8] and Rudolph [4], to provide a theoretical framework for the analysis of evolutionary algorithms. These algorithms are called statistical optimization algorithms because they construct a statistical model (probability distribution) that captures the relationship between the genes and the performance and use this model to bias the random search toward promising regions.

One of the most challenging tasks encountered in statistical optimization is to model distributions of coupled variables. Several approaches have been proposed. De Bonet et al. [5] included pairwise interactions via conditional probabilities in a chain-base model in the MIMIC algorithm, Pelikan and Mühlenbein [11], as well as Baluja and Davies [2] accounted for strong pairwise interactions via dependency graphs, Pelikan et al.

[10] used Bayesian networks to represent higher order interactions in their Bayesian Optimization Algorithm. Though these methods proved effective on some test problems, the effort involved in the construction of the model is sometimes prohibitive. In addition, the amount of data available does not always permit to achieve an acceptable accuracy of the model.

In the present paper, we first investigate how the method used to select these good individuals affects the algorithm efficiency in the case of a simple linear statistical model. Three different schemes are compared. Then we propose a novel approach for accounting for variable interactions in the case of a very coupled problem. The originality of our work lies in the fact that it does not resort to complex models such as Bayesian networks.

Theoretical background

Let $f : D \rightarrow \mathbb{R}$ be a fitness function to maximize over a finite set D . In our case the design space is the set of strings of length N on some alphabet \mathcal{A} of size m , in other words $D = \mathcal{A}^N$. Let $P : D \rightarrow \mathbb{R}$ be a probability measure over D . The aim of statistical optimization is to maximize the expectation $E_P(f) = \sum_{x \in D} f(x)P(x)$ with respect to the probability P over the the set of probabilities over D .

One of the key issues in statistical optimization is the representation of the probability distributions. The choice of the probability model will greatly influence the accuracy of the predictions and determine the effort required to learn the model. In the most general case, i.e. when all the variables are coupled, $m^N - 1$ parameters are necessary to describe the whole distribution. The number of parameters to learn is prohibitive, as it would require an exhaustive search of the design space to deter-

mine. Therefore an approximate model is used.

In this work, we consider probability distributions that can be factored as:

$$P(x) = \prod_{k=1}^N P_k(x_k), \quad x \in D \quad (1)$$

where p_k is a multinomial probability over the alphabet \mathcal{A} . In other words, we assume that the variables $X_k, 1 \leq k \leq N$ are statistically independent, which allows us to estimate the probabilities one variable at a time.

The method proposed in the present paper can be viewed as a stochastic approximation of a gradient ascent applied to the maximization of the expectation $E_P(f)$ in the space of probability distributions, as shown by Mühlenbein in [7].

Algorithm

Estimation and updating of the probabilities

The general idea of statistical optimization is to represent the distribution of good designs by a statistical model (the probability distribution $P(x)$). That model is used as a memory, which stores the desirable features of good individuals. It is used to guide the search toward regions of the design space that have a high probability of containing the optimum. The new observations made at each generation are used to refine the statistical model.

As in standard evolutionary algorithms, design points are represented by strings of finite length (chromosomes). Each digit (gene) of the string can take on a finite set of values (alleles). Our goal is to estimate the probability for each gene of each allele being the best.

Figure 1 shows the general principle of statistical optimization. We start with a uniform probability distribution and approach the optimum distribution $P^*(x)$ gradually by selection learning. At each generation, a population of μ individuals is created by sampling from the current probability distribution. The fitness function f of all the individuals is calculated and $\lambda < \mu$ promising designs are selected. To update the probability distribution, a weighted average rule is adopted. The new distribution is obtained by moving the previous distribution toward the selected population of good designs:

$$P_k^{t+1} = \frac{1}{\alpha + 1} \left(\alpha P_k^t + \hat{P}_k \right) \quad (2)$$

where P_k^t designates the probability distribution of the k^{th} gene at generation t , \hat{P}_k designates its frequency in the selected population and the parameter α determines

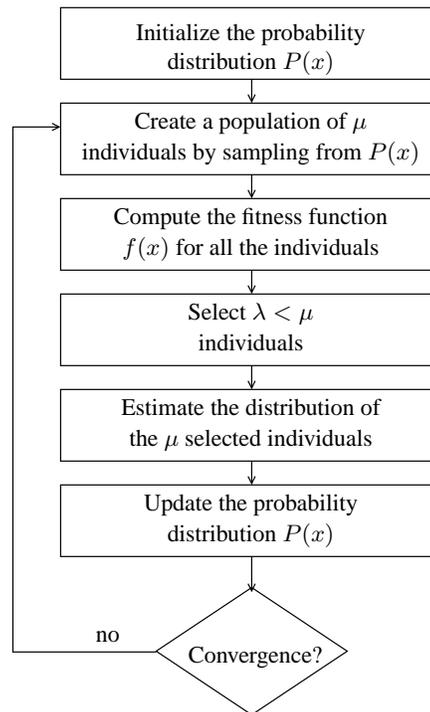


Figure 1: General flowchart of statistical optimization

the amount of memory in the model. When α is large, information obtained in previous generations is preserved, but convergence can be slow, when α is close to 0, the negative influence of observations made during the first generations is eliminated but there is a risk of premature convergence.

Selection method

Three different selection methods are compared, corresponding to a varying selection pressure toward the best individuals:

truncation selection keeps the λ best individuals out of the μ potential individuals. In our study, we use a truncation ratio of two and the frequency of the i^{th} allele of the k^{th} gene is obtained by:

$$P_k(x_i) = \frac{n_k^i}{N/2}$$

where n_k^i is the number of occurrences of allele i of gene k in the top half of the population.

In the remainder of the paper this selection method will be referred to as “top-half”.

rank proportionate selection is slightly different from the general selection scheme outlined in the previous section in that it considers the whole population. A new population is created by copying each design $\mu - r$ times, where r is the rank of the individual in the population. That frequencies used to update the probability distribution are then assessed based on that population. For $\mu = \lambda/2$, the frequencies are given by:

$$P_k(x_i) = \frac{1}{N(N-1)/2} \sum_{r=1}^N \delta_{ir}^k (N-r)$$

This scheme has the advantage that it discriminates between individuals, unlike truncation selection, which considers all best $\mu/2$ individuals equally good. We sometimes refer to this method as “rank”.

hybrid selection is a combination of the two previous schemes, which discriminates between individuals according to their rank, but discards very poor designs. For $\mu = \lambda/2$, the frequencies are given by:

$$P_k(x_i) = \frac{1}{N_h(N_h-1)/2} \sum_{r \leq N_h} \delta_{ir}^k (N_h - r)$$

where $N_h = N/2$. This scheme corrects the shortcomings of the rank proportionate scheme, which

tends to overestimate the goodness of alleles present in poor individuals. This selection method is also referred to as “half-rank” in this study.

Note that no method is a priori better than the other ones. The best scheme has to be determined experimentally on a specific problem.

Introduction of coupling via a change of variables

As pointed earlier, the univariate statistical model does not represent interactions between variables. In order to account for coupling, we introduce auxiliary variables:

$$V_i = \phi_i(\theta_1, \theta_2, \dots, \theta_N), \quad i = 1, \dots, M$$

that are functions of all the variables of the problem. Imposing the probability density $p_V(V_1, V_2, \dots, V_M)$ amounts to constraining the integral of the probability density over the contours $\phi = \text{constant}$. For example, for two design variables and one auxiliary variable, we have the relation:

$$p_V(v) = \int_{\mathcal{C}} p(\theta_1, \theta_2) ds$$

where \mathcal{C} is the set of points such that $\phi(\theta_1, \theta_2) = v$. It is clear that this type of relationship imposes limitations on the design variable probability densities. In the particular case where $M = N$ and ϕ is a bijective mapping between the θ 's and the V 's, the probability density of the θ 's is fully determined by the probability $p_V(V)$.

We have seen that the probability distribution of the auxiliary variables can be used to complement the distribution of the design variables because it summarizes some coupling effects. The method we propose to combine this information is to use the auxiliary variable distribution as a filter that rejects designs that are unlikely to have a high evaluation of the fitness function. The algorithm is the following:

1. Initialize the θ and V probability distributions $p_\theta(\theta)$ and $p_V(V)$
2. Create a pool of ν potential designs by sampling from the distribution $p_\theta(\theta)$
3. Carry out the transformation $V = \phi(\theta)$ and assess the selection probability p_V of each candidate design
4. Select $\mu \leq \nu$ designs based on p_V
5. Compute the fitness function $f(\theta)$ of the selected population

6. Select $\lambda < \mu$ designs based on the fitness function
7. Evaluate the θ -distribution and the V -distribution of the λ selected designs
8. Update the θ and V probability distributions
9. Go to 2 if the convergence criterion is not satisfied

The auxiliary variable distribution is used to pre-select designs from a pool of candidate designs generated from the design variable distribution. The gain expected is a saving of useless analyses of designs that have little chance of performing well.

Application to composite laminates optimization

Test problem

Composite laminates are constructed by stacking a certain number of plies made up of fibers running in one direction embedded in a polymer resin (the matrix). The goal of composite laminate optimization is to determine the optimal number of plies, the material of each ply, and the fiber orientations to maximize some criterion, subject to a set of constraints. In this paper, we will only consider the problem of determining the ply optimal angles. The manufacturing techniques currently in use in industry to fabricate such structures allow only a limited set of discrete values for the fiber orientations (typically $0^\circ, \pm 45^\circ, 90^\circ$), which leads to a combinatorial optimization problem.

The problem considered is the maximization of the longitudinal stiffness A_{11} of a laminate $[\pm\theta_1 / \pm\theta_2 / \dots / \pm\theta_{10}]_s$ subject to constraints on the transverse and shear stiffnesses $A_{22} \geq A_{22}^{min}$ and $A_{66} \geq A_{66}^{min}$. The angles θ_i can take values from $\{0, 15, 30, 45, 60, 75, 90\}$. The material properties are summarized in Table 1.

Table 1: Material properties of graphite-epoxy

Longitudinal modulus, E_1	128 MPa
Transverse modulus, E_2	13 MPa
Shear modulus, G_{12}	6.4 MPa
Poisson's ratio, ν_{12}	0.3
Thickness, t	0.05 mm

The stiffnesses are obtained by $A_{11} = hA_{11}^*$, $A_{22} = hA_{22}^*$ and $A_{66} = hA_{66}^*$ where h is the thickness of the

laminate and

$$\begin{Bmatrix} A_{11}^* \\ A_{22}^* \\ A_{66}^* \end{Bmatrix} = \begin{bmatrix} U_1 & V_1^* & V_3^* \\ U_1 & -V_1^* & V_3^* \\ U_5 & 0 & -V_3^* \end{bmatrix} \begin{Bmatrix} 1 \\ U_2 \\ U_3 \end{Bmatrix}$$

The material invariants are expressed in terms of the stiffness matrix Q as:

$$U_1 = \frac{1}{8}(3Q_{11} + 3Q_{22} + 2Q_{12} + 4Q_{66})$$

$$U_2 = \frac{1}{2}(Q_{11} - Q_{22})$$

$$U_3 = \frac{1}{8}(Q_{11} + Q_{22} - 2Q_{12} - 4Q_{66})$$

$$U_5 = \frac{1}{8}(Q_{11} + Q_{22} - 2Q_{12} + 4Q_{66})$$

and the lamination parameters are obtained by integrating angle dependent terms over the laminate thickness:

$$V_1^* = \sum_{k=1}^N \nu_k \cos 2\theta_k \quad (3)$$

$$V_3^* = \sum_{k=1}^N \nu_k \cos 4\theta_k \quad (4)$$

Since the statistical optimizer does not handle constraints, we use a penalty function. The fitness function is defined as:

$$F(x) = \begin{cases} A_{11}(1 + 0.05 \min(g_1, g_2)) & \text{if } \min(g_1, g_2) \geq 0 \\ 0.8 A_{11} + p \min(g_1, g_2) & \text{if } \min(g_1, g_2) < 0 \end{cases} \quad (5)$$

where p is the penalty parameter. The factor 0.8 guarantees that no infeasible design can have a fitness function higher than a feasible design. Designs that satisfy the constraints with a margin are rewarded proportionally to the margin of safety. The constraints g_1 and g_2 are defined as follows:

$$g_1 = \frac{A_{22}}{A_{22}^{min}} - 1 \geq 0$$

$$g_2 = \frac{A_{66}}{A_{66}^{min}} - 1 \geq 0$$

The particularity of the problem is that the amount of coupling between variables can be calibrated by modifying the value of the constraints A_{22}^{min} and A_{66}^{min} , thereby permitting the investigation of the algorithm behavior under various conditions of complexity. Indeed, when these quantities are zero (no constraints), the optimum design is achieved when all variables are zero and the fitness of any given laminate can be improved by changing any

gene to 0 regardless of all other genes. In that case we say that the variables are independent because the effect of one variable does not depend on the context. In contrast, when the values of A_{22}^{min} and A_{66}^{min} are raised we are dealing with a coupled problem because the effect of changing one gene to 0 will depend upon the value of the other genes: if we are close to the boundary of the feasible domain, decreasing the value of that gene can cause to laminate to be non-feasible and the fitness to decrease¹.

Implementation

The statistical algorithms described in the previous section were applied to the laminate optimization problem. The designs are represented by strings of 10 digits, one for each ply-stack. Each variable can take integer values between 1 and 7, corresponding to the seven possible angles. The goal of the optimization is therefore to determine for each variable the probability of each of the seven values being the best.

The lamination parameters V_1^* and V_3^* , introduced in the previous section, are used as intermediate variables for the double distribution statistical algorithm. These variables have proven their value in composite laminate optimization (Nagendra et al. [9]). Because they capture global stiffness properties of a laminate, they are a natural choice for representing coupling between ply orientation variables. Because lamination parameters are continuous in nature, their probability distribution is modeled by a continuous density, namely a Gaussian distribution. The five parameters of the distribution (the means \bar{V}_1^* and \bar{V}_3^* and the covariance matrix Σ) are estimated at each generation based on the selected population. In the current implementation, no memory mechanism is provided. The efficiency of the algorithms is compared against a standard genetic algorithm with the following characteristics: natural coding, elitist strategy, one-point crossover with probability $p_c = 1.0$, one-gene mutation with probability $p_m = 2.0$ (two gene are mutated with certainty to an adjacent value) and population size 10. These parameters were adjusted experimentally so as to maximize the efficiency of the algorithm.

¹Note that since the stacking sequence has no influence on the problem, it could be solved easily by solving directly for the number of plies of each orientation. However, our purpose is only to devise a test problem that enables us to calibrate the amount of coupling in order to assess the efficiency of our algorithms under specific conditions.

Results

The algorithms are first applied to the decoupled problem ($A_{22}^{min} = 0$ and $A_{66}^{min} = 0$) for which the maximum fitness $A_{11} = 2.58 \times 10^5$ N/m is reached for $[0_{20}]_s$. For these tests, the population size is 10, no memory ($m = 0$) and a mutation rate of 2.0 are found to yield the best results. The reliability of the three variants of the statistical algorithm is compared to that of a standard GA in Figure 2. The reliability of an algorithm designates its probability of finding a practical optimum after a given number of analyses. In this study, reliabilities are estimated over 300 independent runs.

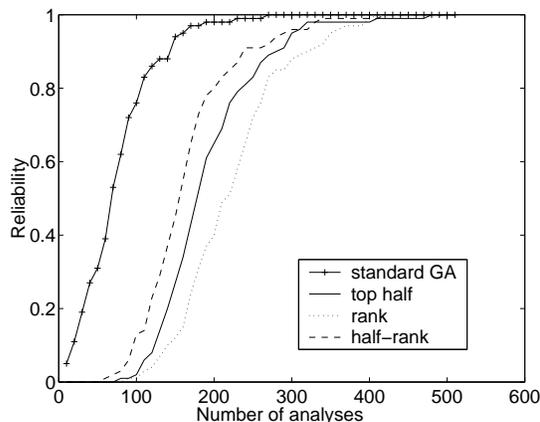


Figure 2: Comparison of the reliability of three variants of the statistical algorithm to that of a standard GA on the decoupled problem. All three algorithms outperform the standard GA. The most successful variant is the “half-rank”, which is also the most selective scheme.

The three versions of the statistical algorithm show a substantially higher reliability than the standard GA. This is hardly surprising since the problem was engineered to demonstrate the superiority of the former when the statistical model and the structure of the problem coincide.

The robustness of the statistical algorithms is then tested by comparing their reliability to that of the standard GA on a strongly coupled problem ($A_{22}^{min} = 1.10^5$ and $A_{66}^{min} = 3.10^4$). In this case, the optimum laminate is $[\pm 15_4 / \pm 30_2 / 90_6]_s$ and the maximum longitudinal stiffness is $A_{11} = 1.47 \times 10^5$ N/m. The two constraints are satisfied: $A_{22} = 1.01 \times 10^5$ N/m and $A_{66} = 3.04 \times 10^4$ N/m. Figure 3 compares the reliability of the three variants to that of the simple GA. The reliability of univariate statistical algorithms is compara-

ble to that of the simple GA: the “top-half” scheme is less reliable than the GA but the “half-rank” scheme still outperforms the GA.

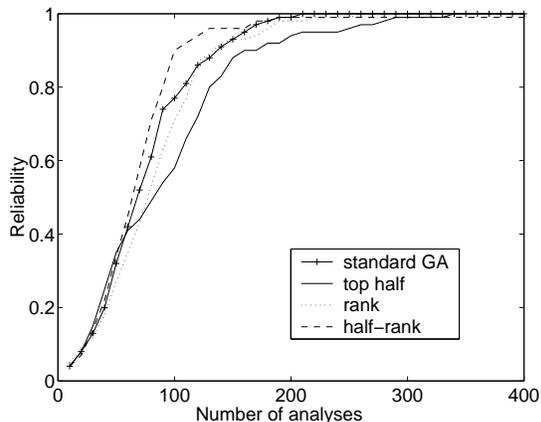


Figure 3: Comparison of the reliability of three variants of the statistical algorithm to that of a standard GA on the coupled problem. The univariate statistical algorithms compare well with the standard GA.

By introducing information on the lamination parameters distribution, we are expecting to improve upon these results by filtering out potential designs that are unlikely to be optimal. As a preliminary study, we implement the double distribution scheme in combination with the simple “top-half” algorithm and apply it to the coupled optimization problem. The information on coupling provided by the lamination parameters contributes to improving the reliability of the algorithm, as shown in Figure 4.

Side effect of the change of variables: drift in the V -space

Observation: drift effect

The change of variables used to account for some interactions between variables has side effects that deserve some attention. The non-linearity of the transformation $\theta \rightarrow V$ alters the topology. Hence, when the selection is based on the distance to a point in the V -space (mean of the normal distribution), some regions of the θ -space are favored. Since the next generation of individuals is created by sampling from the θ -distribution, some bias is introduced that leads to a drift of the distributions.

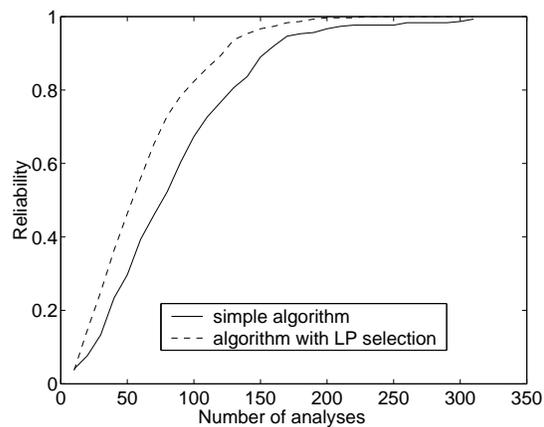


Figure 4: The use of the lamination parameter distribution improves the reliability of the basic “top-half” statistical algorithm.

This phenomenon can be observed by running the algorithm with no selection based on the fitness function. Figure 5 shows the evolution of the mean of the population on the V -space for various random starting points. The graph shows a clear tendency of the distribution to move away from the center, toward the top corners of the design space.

Theoretical analysis

The drift effect can be analyzed theoretically, using the method presented by Rudolph [4]. The algorithm is simplified as follows to facilitate the convergence analysis:

- only one θ variable is considered, and the variable transformation is $V = \phi(\theta) = \cos 2\theta$;
- the variable θ is continuous and its probability distribution is $\mathcal{N}(\mu_\theta, \sigma_\theta^0)$, where the standard deviation σ_θ^0 is fixed while the mean μ_θ is variable. Note that the normal distribution allows values of θ that are not in the interval $[0, 90]$;
- the selection in the V -distribution is performed by chopping off the tails of the distribution $V < \bar{V} - \alpha\sigma_V$ and $V > \bar{V} + \alpha\sigma_V$, where α is a parameter that determines the selection pressure.

The algorithm is the following: the θ probability density $(\mu_\theta, \sigma_\theta^0)$ is initialized; a population of points is created by sampling from that distribution; the transformation $\theta \rightarrow V$ is applied and the mean and the standard deviation of the population are calculated; points that are

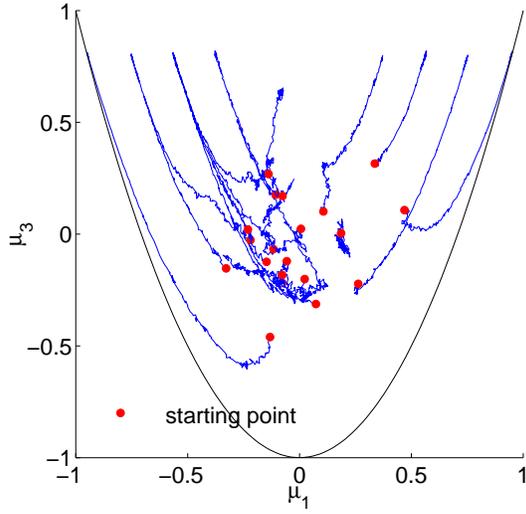


Figure 5: Drift of the mean of the distribution when the algorithm is run with no fitness selection. The distribution moves toward the top corners of the design space due to the non-linear change of variables.

outside the interval $[\bar{V} - \alpha\sigma_V, \bar{V} + \alpha\sigma_V]$ are eliminated; the mean μ_θ of the remaining points is computed and used as new value for the θ -distribution. The process is iterated until convergence is reached.

For the analysis, we consider that the parameters of the θ -distribution are μ_θ^t and σ_θ^0 at a certain generation. Our goal is to find the expectation of the parameter μ_θ^{t+1} at the next generation. The probability distribution in the θ -space is defined as follows:

$$p(\theta) = \frac{1}{\sqrt{2\pi}\sigma_\theta^0} e^{-\frac{1}{2}\left(\frac{\theta - \mu_\theta^t}{\sigma_\theta^0}\right)^2} \quad (6)$$

The transformation $\theta \rightarrow V$ is decreasing over $[0, 90]$, therefore the probability density in the V -space is obtained by:

$$p_\theta(\theta)d\theta = -p_V(V)dV \quad (7)$$

Hence

$$p_V(V) = -\frac{p_\theta(\theta)}{dV/d\theta} \quad (8)$$

$$= \frac{1}{2\sqrt{1-V^2}\sqrt{2\pi}\sigma_\theta^0} e^{-\frac{1}{2}\left(\frac{1/2 \cos^{-1} V - \mu_\theta}{\sigma_\theta^0}\right)^2} \quad (9)$$

The expectation of the mean in the V -space is given by:

$$\mu_V = \frac{\int_{-1}^1 V p_V(V) dV}{\int_{-1}^1 p_V(V) dV} \quad (10)$$

$$= \frac{\int_{-1}^1 \frac{V}{2\sqrt{1-V^2}\sqrt{2\pi}\sigma_\theta^0} e^{-\frac{1}{2}\left(\frac{1/2 \cos^{-1} V - \mu_\theta}{\sigma_\theta^0}\right)^2} dV}{\int_{-1}^1 p_V(V) dV} \quad (11)$$

The expectation of the standard deviation in the V -space is given by:

$$\sigma_V^2 = \frac{\int_{-1}^1 (V - \mu_V)^2 p_V(V) dV}{\int_{-1}^1 p_V(V) dV} \quad (12)$$

$$= \frac{\int_{-1}^1 \frac{(V - \mu_V)^2}{2\sqrt{1-V^2}\sqrt{2\pi}\sigma_\theta^0} e^{-\frac{1}{2}\left(\frac{1/2 \cos^{-1} V - \mu_\theta}{\sigma_\theta^0}\right)^2} dV}{\int_{-1}^1 p_V(V) dV} \quad (13)$$

Finally, the expectation of the new center of the θ -probability distribution is obtained by applying the truncation selection:

$$E[\mu_\theta^{t+1}] = \frac{\int_{\phi^{-1}(\mu_V - \alpha\sigma_V)}^{\phi^{-1}(\mu_V + \alpha\sigma_V)} \theta p_\theta(\theta) d\theta}{\int_{\phi^{-1}(\mu_V - \alpha\sigma_V)}^{\phi^{-1}(\mu_V + \alpha\sigma_V)} p_\theta(\theta) d\theta} \quad (14)$$

We calculate $E[\mu_\theta^{t+1} | \mu_\theta^t]$ for two different selection pressures: $\alpha = 0.5$ (strong selection) and $\alpha = 2.0$ (moderate selection). The integrals are computed numerically. The results are shown in Figure 6. Surprisingly, it appears that the direction of the drift depends on the selection pressure: for a strong selection pressure, the distribution moves toward 45 degrees, for a moderate selection pressure, it is attracted to 0 and 90 degrees. Note that it is not possible to obtain the full curve, as numerical difficulties are encountered when the selection interval was not contained entirely in $[-1, 1]$ ($\phi^{-1}(V)$ not defined).

These results are verified experimental by running the simplified algorithm described previously. To guaranty the number of selected individuals while maintaining the idea of variable selection pressure, the selection procedure is slightly modified: out of 1000 candidate individuals created, a certain number are selected based on the lamination parameters: 500 individuals for a strong selection pressure, 950 for weak selection pressure. The drift of the the θ -distribution for an initial distribution $\mathcal{N}(10, 5)$ confirms the theoretical predictions: the distribution is attracted to 45 degrees for strong selection, to 0 degree for weak selection.

Though the drift effect caused by the non-linear change of variables is now clearly established, its influence on the optimization is yet to be characterized.

Concluding remarks

In this paper, we have shown how a statistical algorithm based on a simple model can outperform a standard Genetic Algorithm on a decoupled composite laminate optimization problem, and compares very well with it on a coupled problem. We have compared three selection methods and proposed a hybrid schemes which is a combination of truncation and rank proportionate schemes. In the two examples treated, this schemes yielded the best results. We also propose a new method for accounting for coupling between variables, which is not based on higher order statistical models, but on an auxiliary probability distribution of intermediate variables — lamination parameters in the case of composite optimization. This double distribution method shows promising results on the coupled problem when used in conjunction with truncation selection.

Future work will extend the analysis of double distribution distribution evolution algorithms. In particular, it will check whether the promising results obtained in the present paper can be reproduced when the algorithm is applied to more complex and more realistic problems. In addition, many improvements can be made to the algorithm, in particular some distributions may be more appropriate than the normal distribution, which may unnecessarily constrain exploration of new regions.

Acknowledgments

The authors gratefully acknowledge the financial support of Visteon Corporation and the technical advice of Dr Naveen Rastogi, our project sponsor at Visteon.

References

- [1] Shumeet Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Tech. Rep. No. CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, PA, 1994.
- [2] Shumeet Baluja and Scott Davies. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In *Proc.*

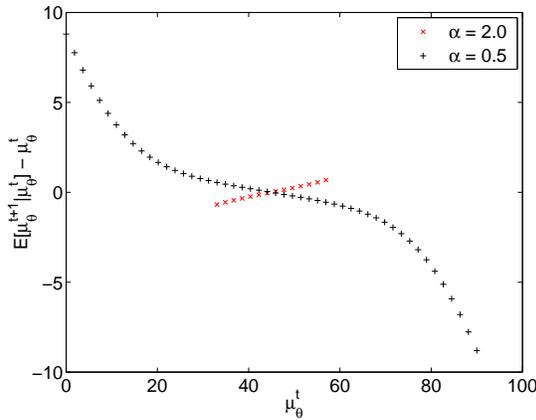


Figure 6: Theoretical expectation of the position of the center of the θ distribution in terms of its current position ($E[\mu_{\theta}^{t+1}|\mu_{\theta}^t]$) for strong (black +’s) or moderate (red \times ’s) selection pressure.

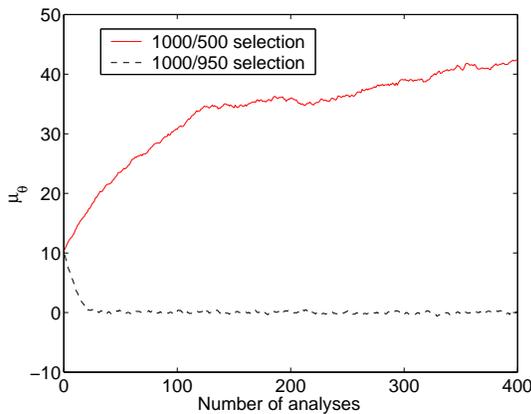


Figure 7: Drift of the θ -distribution for a strong (red solid line) or weak (black dashed line) selection pressure. The experimental results confirm our theoretical predictions.

14th International Conference on Machine Learning, pages 30–38. Morgan Kaufmann, 1997.

- [3] A. Berny. Extending Selection Learning toward Fixed-Length d -ary Strings. In *5^{ème} conférence internationale sur l'Évolution Artificielle*, 2001.
- [4] M. Böhfeld and G. Rudolph. Towards a theory of population-based incremental learning. In *Proceedings of the 4th IEEE Conference on Evolutionary Computation, Piscataway, NJ*, pages pp. 1–5. IEEE Press, 1997.
- [5] Jeremy S. De Bonet, Charles L. Isbell, and Paul Viola. MIMIC: Finding optima by estimating probability densities. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in neural information processing systems*, volume 9, page 424. The MIT Press, Cambridge, 1997.
- [6] Hillol Kargupta. The gene expression messy genetic algorithm. In *Proceedings of 1996 IEEE International Conference on Evolutionary Computation*, pages 814–819, Piscataway, NJ, 1996. IEEE Press.
- [7] Tilo Mahnig and Heinz Mülenbein. Mathematical analysis of optimization methods using search distributions. In *Proceedings of the Third International Symposium on Adaptive Systems*, pages 166–185, Havana, 2001.
- [8] Heinz Mühlenbein and Thilo Mahnig. Evolutionary Algorithms: From Recombination to Search Distributions. *Theoretical Aspects of Evolutionary Computing*, pages 137–176, 2000.
- [9] S. Nagendra, R.T. Haftka, Z. Gürdal, and L.T. Watson. Derivative based approximation for predicting the effect of changes in laminate stacking sequence. *Structural Optimization*, 11:235–243, 1996.
- [10] Martin Pelikan, David E. Goldberg, and Erick Cantú-Paz. BOA: The Bayesian optimization algorithm. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, volume I, pages 525–532, Orlando, FL, 1999. Morgan Kaufmann Publishers, San Fransisco, CA. Also IlliGAL Report No. 99003.
- [11] Martin Pelikan and Heinz Mühlenbein. The bivariate marginal distribution algorithm. In R. Roy,

T. Furuhashi, and P. K. Chawdhry, editors, *Advances in Soft Computing - Engineering Design and Manufacturing*, pages 521–535, London, 1999. Springer-Verlag.