Multi-Objective Optimization of a Composite Material Spring Design Using an Evolutionary Algorithm

Frédéric Ratle¹, Benoît Lecarpentier¹, Richard Labib², François Trochu¹

¹ Centre de Recherche Appliquée Sur les Polymères, Département de Génie Mécanique, École Polytechnique de Montréal, C.P. 6079, Succ. Centre-ville, Montréal, Québec, Canada H3C 3A7

{Frederic.Ratle, Benoit.Lecarpentier, Trochu}@polymtl.ca

² Département de Mathématiques et de Génie Industriel, École Polytechnique de Montréal, C.P. 6079, Succ. Centre-ville, Montréal, Québec, Canada H3C 3A7
Richard.Labib@polymtl.ca

Abstract. A multi-objective evolutionary algorithm is applied to optimize the design of a helical spring made out of a composite material. The criteria considered are the minimization of the mass along with the maximization of the stiffness of the spring. Considering the computation time required for finite element analyses, the optimization is performed using approximate relations between design parameters. Dual kriging interpolation allows improving the accuracy of the classical model of spring stiffness by estimating the error between the model and the results of finite element analyses. This error is taken into account by adding a correction function to the stiffness function. The NSGA-II algorithm is applied and shows satisfactory results, while using the correction function induces a displacement of the Pareto front.

1 Introduction

The problem addressed here consists of optimizing mechanical properties, e.g. stiffness and mass, of springs made out of composite materials given a set of design parameters, which constitutes the design space of the problem. Rather than considering an aggregation of the objectives, a multi-objective evolutionary algorithm, which can deal with two or more contradictory objectives, is used in this investigation. For instance, in the case of spring design, maximizing stiffness for a given geometry results in increasing the mass, whereas the latter has to be minimized. The maximum admissible load is also a contradictory objective with respect to the mass: a lighter spring admits a lower load before failure, while it is interesting in applications to maximize this load. However, only mass and stiffness will be considered in this paper.

1.1 Helical springs and composite materials

With the development of new composite manufacturing technology based on resin injection through fibrous reinforcements, it is nowadays possible to replace the classical metal wire helical springs by a lightweight composite version particularly well

adapted for torsional loads and efficient at storing elastic deformation energy. This change of material brings about the possibility of substantially reducing spring mass while the other desired mechanical properties, such as stiffness and maximum deflection, could remain reasonably unchanged compared to metallic springs.

At the design stage, optimal geometric and material-related parameters are required to get the best compromise among many possible spring configurations. Parameters can be divided into two categories, geometric and material-related. Unlike metal springs, it is possible to adapt the structure of a composite spring to the specific requirements of an application. In our case, the material-related variable is the braid angle, assuming that each ply has the same fiber orientation. All the design parameters ($D, d_{int}, e, p, N, \theta$) are described in Fig. 1.

Fig. 1. Geometric and material-related design parameters

These design variables lie within the boundary values specified in Table 1. Note that the composite material is a laminate made of fiberglass and epoxy resin.

Table 1. Design space of the problem

Parameter	Description	Minimum value	Maximum value	Units
D	coil diameter	60	200	mm
d_{int}	inner diameter	8	16	mm
e	thickness	2	6	mm
p	helix pitch	50	70	mm
N	number of active coils	2	7	-
θ	braid angle	30	55	degrees

It is convenient for the sake of brevity to introduce the following notations: the helix angle α (1), the wire outer diameter d_{ext} (2), the spring index C (3), and the thickness ratio ξ (4), given by the following relations:

$$\alpha = \arctan\left(\frac{p}{2D}\right), \quad (1)$$

$$d_{ext} = d_{int} + 2e \quad , \quad (2)$$

$$C = \frac{D}{d_{ext}} \quad , \quad (3)$$

$$\xi = \frac{e}{d_{ext}} \quad . \quad (4)$$

1.2 Classical formula for helical spring design

The following formula is used in classical spring theory to evaluate the stiffness k of a helical tubular spring with an empty core:

$$k = \frac{G_{12}(d_{ext}^4 - d_{int}^4)}{8ND^3} \quad . \quad (5)$$

where G_{12} is the in-plane material shear modulus. However, this formula results from an approximation, i.e., only the torsion component of the applied load matters. In fact, this component is usually dominant over the flexural one, but in the case of small spring indices ($C \leq 5$) or important helix angles ($\alpha \geq 15^\circ$) a gap exists with the theoretical formula. Since flexural stress can reach half the value of the torsional stress, it cannot be neglected. Furthermore, this formula has been established for isotropic materials and cannot be straightforwardly applied to orthotropic materials such as braided composites. To solve this problem, a correction function was devised for the output values of the classical stiffness formula. The approach proposed in this investigation is described in section 3.

1.3 Previous work in spring design optimization

Very few studies have been conducted on the optimization of spring design. Yokota et al. [1] have worked with a single-objective genetic algorithm to solve the constrained nonlinear integer programming problem of minimizing the weight of a helical metallic spring. Also, Gobbi and Mastinu [2] have applied multi-objective programming, a branch of Operations Research, to optimize simultaneously the mass and the stiffness of a spring. To improve the classical stiffness formula, they have introduced an analytical formula based on the moment of inertia of thin-walled tubular composite materials.

The present work aims at applying multi-objective evolutionary algorithms to this type of problem. However, the problem settings in [1, 2] being different from that of this paper (particularly the thin-walled material assumption), comparison is difficult. Nevertheless, the results obtained with the correction function are compared to a direct application of NSGA-II on the classical formulae of stiffness and mass.

2 Multi-Objective Evolutionary Optimization

Evolutionary algorithms have emerged in the recent years as one of the most efficient approaches to solve multi-objective optimization problems. Their intrinsically parallel nature allows finding the best trade-off solutions among a pool of potential designs.

When dealing with multiple objectives, finding a set of optimal solutions rather than a single optimum is in many ways more convenient. On one hand, from a computational point of view, it is much more straightforward to optimize a single-objective function. On the other hand, this presupposes that the designer is able to aggregate the objectives in a single function expressing the relative importance of each criterion. This *a priori* knowledge is usually hard to translate into quantitative information, and the choice of the weighting coefficients is somewhat arbitrary.

During the last decade, *Pareto-based* algorithms have been studied extensively [3, 4, 5]. These algorithms allow the decision-maker to choose among many optimal solutions and do not require quantitative *a priori* information. As a result, the decision concerning the optimal design can be made *a posteriori*.

2.1 Pareto dominance

The main difference between single- and multi-objective evolutionary algorithms is the way of performing the selection of individuals. The original concept of fitness is no longer significant in multi-objective optimization. It is rather the concept *Pareto-dominance* that is relevant to rank the individuals in a given generation. If x and y are two individuals, x is said to be *Pareto-dominant* over y if x is not worse than y with respect to all objectives and is strictly better than y for at least one objective. This way of comparing individuals enables the algorithm to pick the best solutions of compromise by considering all criteria at once.

2.2 The NSGA-II Algorithm

The original non-dominated sorting genetic algorithm (NSGA) is presented by Srinivas and Deb [3], while NSGA-II, the faster version of the algorithm, is described by Deb et al. [4]. The algorithm is based on the concept of domination rank. Individuals of a certain generation are selected (originally with the deterministic tournament method [3]) on the basis of their domination rank in the population. When two individuals are compared, the one with the smallest domination rank wins. Selection is based on this rank, while crossover and mutation remain unchanged.

If the domination ranks of two individuals are equal, the crowding distance will guide the selection process. This distance is a measure of the local sparseness and aims to promote diversity in the population. The individual with the largest crowding distance will be favored. Thus, individuals lying in a region with a high density of potential solutions will be penalized to prevent premature convergence.

The most valuable advantages of NSGA-II over NSGA are the introduction of elitism, the less computationally expensive sorting of individuals with respect to their domination rank, and the concept of crowding distance, which does not require a user-

specified niche radius parameter, as would require a classical fitness sharing method. Some practical applications of NSGA-II have been studied in [6, 7, 8] and have demonstrated the efficiency of this approach.

3 Correction Function

As stated in section 1, the problem of composite springs has been addressed with approximate relations rather than numerical analyses because of the computational cost involved. However, stiffness calculation is inaccurate for composite springs. A first approach that could be considered consists, as described by Jin [9], of updating the fitness function during the evolution process. Yet, by looking at the differences between the classical model and the results of finite element analysis, it is clear that some regions of the search space are very well approximated, while some other regions show a non negligible error. Adaptive fitness function approximation would then result in evaluating a great amount of points which are already well approximated by the classical formula, while uncertain regions could remain unexplored.

An interesting alternative to the latter approach is to build a function representing the approximation error as a function of the design parameters using dual kriging. This error is the difference between the output of the analytical formula and numerical results. The principles underlying kriging can be found in Cressie [10] or Trochu [11].

The first step of the correction approach consists of performing a uniform experimental design in the search space. Different spring configurations are evaluated by the analytical formula and the finite element analysis. The points evaluated then constitute the training set to build the kriging model, which output is the approximation error. The objective function can then be expressed as a combination of the analytical formula and the correction function. The main drawback of this approach is the time required to build a substantial training set, the search space being multi-dimensional and the numerical simulation computationally expensive. Consequently, two steps are undertaken to reduce both dimensionality and the number of function evaluations.

Firstly, a sensitivity analysis shows that parameters responsible for the gap with finite element calculations are the spring index C , the helix angle α , the thickness ratio ξ , the number of active coils N and the material used, namely here the braid angle θ . The experimental design is thus established with these five parameters only. The stiffness can hence be expressed as

$$k = k_o [1 + L(C, \alpha, \xi, N, \theta)] , \quad (6)$$

where k_o is the stiffness obtained from (5). Function L has the following form:

$$L(\mathbf{x}) = a(\mathbf{x}) + W(\mathbf{x}) , \quad (7)$$

where $\mathbf{x}$ is the input vector, $a(\mathbf{x})$ is the drift, a known function representing the global trend of the phenomenon, and $W(\mathbf{x})$, a random function representing the local deviation from the drift. In the present case, the following functions have been used:

$$a(\mathbf{x}) = a_0 + a_1 \mathbf{x} , \quad (8)$$

$$\text{cov}(W(\mathbf{x}^i), W(\mathbf{x}^j)) = \mathbf{K}(c(\mathbf{x}^i, \mathbf{x}^j)), \quad (9)$$

where $\mathbf{K}$ is the correlation matrix, and c the correlation function between any two of the N training samples. In this study, function c is assumed to be linear:

$$c(\mathbf{x}^i, \mathbf{x}^j) = \sum_{k=1}^n b_j |x_k^i - x_k^j|. \quad (10)$$

where the b_j are the correlation parameters, and n the dimension of the design space.

Secondly, an approach based on the query learning paradigm [12, 13] is used to reduce the number of function calls required to build the correction function. From the initial experimental design, different independent datasets are picked using the bootstrap method, i.e., sampling with replacement from the original data, and used to construct a kriging estimator. The point which splits most evenly the predictors is added to the original training set. This procedure is repeated for a fixed number of iterations and the improvement is measured by a cross-validation estimation of the error on the current data. The general idea of this approach takes root in the “Query by Bagging” algorithm presented by Abe and Mamitsuka [13].

The aim of this method is to decrease more rapidly the generalization error than by a passive learning approach, where training data is selected at random. Knowing that the amount of information provided by a randomly selected point decreases as points are added to the training set, such an approach will hopefully provide pertinent information to the learner and allow using a smaller dataset for the construction of the correction function. For the sake of clarity, the correction function is plotted in Fig. 2. Only the influence of C and α is showed here (ξ , N and θ are kept constant).

Fig. 2. Correction function

From the previous figure, it can be seen that the springs presenting a large index C are well approximated by the analytical formula, especially for small helix angles. As

C decreases, a gap appears. For small angles, there is a positive error, which means that stiffness is underestimated by the classical theory. This is due to the flexural component of the stress. A higher load has to be applied to attain the same deflection. The stiffness of the spring thus appears higher. At high angles, the error is negative. This means that the stiffness is overestimated. At such angles, the resulting stiffness of the spring is reduced since the compression load contributes to increase D , the spring diameter.

Therefore, the optimization problem can finally be stated as follows:

$$\text{Max} \quad k = \frac{G_{12}(d_{ext}^4 - d_{int}^4)}{8ND^3} [1 + L(C, \alpha, \xi, N, \theta)] , \quad (11)$$

$$\text{Min} \quad mass = \frac{\rho\pi ND(d_{ext}^2 - d_{int}^2)}{4\cos\alpha} , \quad (12)$$

$$\text{s.t.} \quad p - d_{ext} > 0 . \quad (13)$$

The above constraint ensures that coils do not touch each other. It is handled by assigning a penalty to unfeasible individuals. The parameter ρ is the density of the material. Of course, mass calculation does not require a correction function. It is computed directly from the geometric and material-related parameters of the spring.

4 Experimental Results

4.1 Experimental Conditions

The NSGA-II is built on the basis of the Evolving Objects library, with a real representation. Variation relies solely on Gaussian mutation. The mutation is continuous for all the design parameters except the number of active coils N and the braid angle θ , for which the Gaussian law is approximated by a binomial law, the latter variables being discrete. For all the experiments, the population size is 200, the stopping criterion is the maximum number of generations, which is set to 200, and the mutation rate is $1/n$, where n is the space dimension. The deterministic tournament size is set to 2. These parameters have been statically determined using a sensitivity analysis. All experiments are performed with a Pentium IV processor running at 2.8GHz under Windows XP.

4.2 Results and Discussion

Table 2 shows the average CPU time for the NSGA-II algorithm with and without the correction function. As an indication, NSGA, the former version of the algorithm, has been tested on the same problem. Twenty runs of each algorithm were conducted.

Table 2. CPU time

Algorithm	Average CPU time (sec.)
NSGA-II with correction function	20.218
NSGA-II	13.130
NSGA with correction function	179.197
NSGA	152.566

As we can see, NSGA-II is much faster than its former version. It is known [4] that NSGA is of complexity $O(mN^3)$ while NSGA-II is $O(mN^2)$, m being the number of objectives, and N the population size. Looking at Table 2, NSGA-II seems to be more than 10 times faster than NSGA for this particular problem. The ratio is different when the correction function is considered because of the time required by the kriging calculations.

The Pareto front obtained with the NSGA-II algorithm and the correction function is compared in Fig. 3 with the one derived with the classical stiffness formula only.

Fig. 3. Pareto fronts for NSGA-II with and without the correction function

According to Fig. 3, the classical spring model overestimates the stiffness for a given mass. Indeed, as mass and stiffness increase, a minute displacement of the Pareto front is noticeable. This displacement reaches a maximum of approximately 4% of the stiffness value. To ensure that this value is statistically significant, the maximal displacement of the front was measured for all the runs of the algorithm and showed an average of 3.93% and a standard deviation of 0.05%. Such a difference in the value of stiffness is physically significant and justifies using the correction function.

Another important point is the presence of an inflection point, which is zoomed in Fig. 4.

Fig. 4. Zoom on the Pareto front in the neighborhood of the inflection point

This bilinear behaviour has been noted in every run of the algorithm and is due to the boundary limits of the different parameters. As the springs are getting larger, d_{int} increases and reaches its maximum value around $m = 40$. Afterwards, mass grows proportionally to the thickness e , which begins to increase from this point. This shift in the parameters responsible for mass growth explains the slope change, because their effect on the variation of stiffness is different.

5 Conclusion

The aim of this work was to optimize simultaneously the mass and the stiffness of a composite-made spring. Good results were obtained with the NSGA-II algorithm for this particular problem. The use of dual kriging for estimating the approximation error of the classical theory induced a statistically significant, but minute change in the position of the Pareto front.

Future study could focus on extending this work to three or more objectives which are important in spring design optimization. For instance, the maximum admissible load, maximum deflection and elastic energy could also be considered in the optimization process.

It would also be interesting to compare the results with the direct use of numerical simulations in order to evaluate the accuracy of the fitness model. However, this would require problem-level approximations. For instance, Parmee [14] reports the use of mesh refinement techniques during the evolution process, or coevolution with meshes of different precisions.

Acknowledgements

Authors wish to thank NSERC, FQRNT, the Auto21 Network of Centres of Excellence as well as Composites Atlantic Ltd. for their financial support.

References

1. Yokota, T., Taguchi, T., Gen, M.: A solution method for optimal weight design problem of helical spring using genetic algorithms. In: *Computers Ind. Engng.*, Vol. 33, Nos 1-2 (1997) 71-76
2. Gobbi, M., Mastinu, G.: On the optimal design of composite material tubular helical springs. In: *Meccanica*, Vol. 36. Kluwer Academic Publishers (2002) 525-553
3. Srinivas, D., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. In: *Journal of Evolutionary Computation*, Vol. 2, No. 3 (1995) 221-248
4. Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: M. Schoenauer et al. (eds): *Parallel Problem Solving from Nature*, Vol. 6 (2000) 849-858
5. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. In: *Evolutionary Computation*, Vol. 8, No. 2 (2000) 173-195
6. Hamda, H., Roudenko, O., Schoenauer, M.: Application of a multi-objective evolutionary algorithm to topological optimum design. In: I. Parmee (ed): *Fifth International Conference on Adaptive Computing in Design and Manufacture* (2002)
7. Lahanas, M., Baltas, D. and Zamboglou, N.: A hybrid evolutionary multiobjective algorithm for anatomy based dose optimization algorithm in HDR brachytherapy. In: *Physics in Medicine and Biology*, Vol. 48 (2003) 399-415
8. Wu, J.-L., Agogino, A.M.: Automating keyphrase extraction with multi-objective genetic algorithms. In: *Proceedings of the 37th Hawaii International Conference on Systems Sciences* (2004)
9. Jin, Y.: Fitness approximation in evolutionary computation: a survey. In: *Soft Computing Journal*, Vol. 4 (2003), In press
10. Cressie, N.: *Statistics for Spatial Data*. Wiley (1993)
11. Trochu, F.: A contouring program based on dual kriging interpolation. In: *Engineering with Computers*, Vol. 9 (1993) 160-177
12. Cohn, D.A., Ghahramani, Z., Jordan, M.I.: Active learning with statistical models. In: *Journal of Artificial Intelligence Research*, Vol. 4 (1996) 129-145
13. Abe, N., Mamitsuka, H.: Query learning strategies using boosting and bagging. In: *Proceedings of The Fifteenth International Conference on Machine Learning* (1998) 1-9
14. Parmee, I.: *Evolutionary and Adaptive Computing in Engineering Design*. Springer-Verlag, 2001