

Optimization of truss structures by a stochastic method

Dominique Chamoret^{1,a}, Kepeng Qiu² and Matthieu Domaszewski¹

¹M3M, Université de Technologie de Belfort Montbéliard, 90000 Belfort, France

²The Key Laboratory of Contemporary Design & Integrated Manufacturing Technology, Northwestern, Polytechnical University, Xi'an, China

Received 08 July 2008, Accepted 12 October 2008

Abstract – This paper presents an original stochastic method of optimization (PGSL) for optimum design of truss structures. PGSL (Probabilistic Global Search Lausanne, developed at the Swiss Federal Institute of Technology at Lausanne) has been interfaced with a finite element code, then applied and tested on several size optimization problems of plane truss structures. This method uses a probability distribution function (PDF) to sample the search space and select the best structure. A PDF represents the probability of finding a good solution at any given point in the search domain. The better solutions are more likely to be found in the neighbourhood of good solutions; hence, probabilities are increased in regions where good solutions are found. The main advantage of this method is to avoid the sensitivity analysis and to converge to the global optimum.

Key words: Structural Optimization, Truss Structures, Stochastic Method

1 Introduction

Structural optimization is the subject of a great interest in design of mechanical systems in the industry such as automotive, aeronautics, and civil engineering industry. The aim is not only to improve some mechanical properties but also to obtain a global optimization of weight, rigidity, resistance and cost. It can also concern aesthetics aspects, if the latter can be quantified.

To optimize appears like to find the best solution to a certain design problem. The mathematical representation of optimization problem is the minimization, or maximization of a scalar-valued objective function with respect to a vector of design parameters. A large number of optimization methods have been developed last years.

The purpose of this paper is to use an original search direct method to perform structural optimization size. Sections 2 and 3 present the structural optimization problem and a brief review of methods for solving the optimization problem. Section 4 introduces the method PGSL (Probabilistic Global Search Lausanne [1]) used to solve the structural optimization problem. In section 5, PGSL algorithm is applied for size optimization of truss structures.

2 Structural optimization

Structural optimization techniques have to take into account various aspects. These techniques depend on the kind of problems what they are used for. Common type of structural optimization can be classified as follows: size optimization, shape optimization and topology optimization.

2.1 Size optimization

This is the simplest problem that can be defined in structural optimization. The aim is usually to minimize the weight of the structure subject to certain behavioral constraints on stress and displacements. None modifications of the geometric model are allowed. Design variables influence neither the topology nor the geometry of the structure which are considered to be fixed. It concerns a modification of cross sections or thickness of the structural element. In a finite element context, the mesh does not change during the optimization process.

2.2 Shape optimization

Shape optimization is usually performed on solid models, although there are many applications for surface models (shells). It concerns the changes of the geometric dimensions of the initial design to satisfy a large variety of objective functions (stress, weight, displacements...). It is fundamental to understand that the design variables are linked to the geometric representation of the boundaries of the structures. So only shape changes which do not affect the topology of the structures are accepted. This means that it is not possible to create holes, or to modify the connectivity between different constitutive elements of the optimized structure. Usually a finite element model can be used as starting point. The shape of this initial model is changed at each step of the optimization process. It needs to update the finite element mesh during the evolution of the calculus. The aim is to understand how the structure is affected by the

^a Corresponding author: Dominique.chamoret@utbmfr

evolution of its shape. The evolution of the finite element mesh is the main difficulty in design sensitivity analysis.

2.3 Topology optimization

Topology optimization [2] allows the control of not only geometry but also of topology of a structure without any restriction on the number or the nature of the structural members. It involves the determination of the type of structural members (location, number, shape of holes) and their connectivity within the design space available when the loads and the boundary conditions are supposed known. Topology optimization appears much more interesting than size and shape optimization. Indeed, its flexibility leads to original and innovative solutions to engineering problems. Topology optimization really increases the performance of structures. For classical problems, the gain in the objective functions could reach between 40 percents and 100 percents compared to size and shape optimization [3].

3 Structural optimization problem

The aim of any structural optimization problem can be defined in the following way: to determine the best design for a given problem subject to certain restrictions. The design variables can be defined starting from the geometry, the topology or material properties of the structures. A set of derived parameters related to mechanical behavior can also be obtained: strains, stresses, deflections, natural frequencies and loads... The cost (or objective) function is given by the proper choice of the design criterion. This function can be either minimized or maximized.

From a mathematical point of view, this problem can be formulated in the following way:

$$\begin{cases} \min : F(X) \text{ with } X = (x_1, x_2, \dots, x_n) \\ \text{subject to: } g_i(X) \leq 0 \text{ with } i = 1, \dots, p \end{cases} \quad (1)$$

The nature of the mathematical programming problem depends on the functional form of F and g_i .

Gradient based optimization methods (conjugate gradient, sequential quadratic programming, etc) are often used to solve the minimization problem given by relation (1). These approaches have been employed successfully for several years. Nevertheless, some drawbacks may limit their application to practical situations:

- The cost function can be not smooth.
- These approaches require substantial gradient information. The gradient can be evaluated either analytically or numerically. Unfortunately, the exact gradient may be difficult to reach even sometimes impossible, and its numerical approximation can really increase the numerical cost.
- The cost function may be non convex and so can present multiple local minima and maxima. For practical problems where no hypothesis can be made a priori, deterministic methods often converge towards one of local minima of the cost function.

It appears that these methods may find the global minimum of the function under some hypotheses like convexity, differentiability, etc. In other words, if the objective function fulfils these hypotheses in a local region containing the

desired minimum, and if the initial configuration is somewhere inside this region, these methods converge very fast toward this solution.

The above mentioned drawbacks explain the growing interest in the development of a class of optimization strategies that do not require gradient information. These approaches can be divided into two broad categories: deterministic and stochastic strategies.

- Deterministic strategies
 - Direct search methods
 - Response surface methodology
- Stochastic strategies (Meta-heuristics)
 - Adaptive random search technique
 - Genetic algorithms [4]
 - Simulated annealing
 - Tabu search
 - Particle Swarm Optimization
 - Ant Colony Optimization

The power of metaheuristics comes from the fact that they are robust and can deal successfully with a wide range of problem areas, and especially in structural optimization [5-9]. Their main drawback is the large number of evaluations of the objective function. So, these approaches may result in a high computational cost. Today, many researchers are focused on techniques to decrease computational time. Performance of these algorithms can be improved with the use of parallel computing. Hybrid optimization algorithms have been proposed by combining simulated annealing and genetic algorithms with a local optimization method to develop more efficient methods with relatively faster convergence than the pure metaheuristics. Some attempts have been made to improve genetic algorithms

4 PGSL algorithm

In this paper, we have chosen to use a direct stochastic algorithm for global search developed by Raphael and Smith [10-13]: PGSL "Probabilistic Global Search Lausanne". It is based on the assumption that sets of better values are more likely to be found in the neighbourhood of sets of "good values" and, therefore, intensifies search in regions that contain sets of "good solutions". Gradients are not required in the search. This algorithm uses four embedded loops: the iteration cycle, the focusing cycle, the probability updating cycle and the sampling cycle (Table 1).

4.1 Sampling cycle

A given number n_s of points are generated randomly (Monte Carlo technique) according to the current PDF (probability density function). For each point, the specified objective function is evaluated and the point having the best cost is selected as a^* .

4.2 Probability updating cycle

During this cycle, the probabilities of domains containing "good solutions" are increased and the others are decreased. After each iteration of the sampling cycle (n_{puc} iterations required), the probability of the interval containing the best point a^* is increased by multiplication by some factor.

<p>Set the complete search space as the current subdomain</p> <p>❶ SUBDOMAIN CYCLE (n_{sdc} loops)</p> <p>❷ FOCUSING CYCLE (n_{fc} loops)</p> <ul style="list-style-type: none"> • Initialization of c^* <p>❸ PROBABILITY UPDATING CYCLE (n_{puc} loops)</p> <ul style="list-style-type: none"> • Initialization of b^* <p>❹ SAMPLING CYCLE (n_s loops)</p> <ul style="list-style-type: none"> • Evaluate n_s sample • Evaluate the cost function • Select the point having the best cost as a^* <p>❺ END OF THE LOOP</p> <ul style="list-style-type: none"> • If $f(a^*) < f(b^*)$ then update b^* • Increment the probability of the interval containing c^* <p>❻ END OF THE LOOP</p> <ul style="list-style-type: none"> • If $f(b^*) < f(c^*)$ then update c^* • Subdivide the interval containing c^* and redefine probability values <p>❼ END OF THE LOOP</p> <ul style="list-style-type: none"> • Choose a new subdomain centred around the best current solution <p>❶ END OF THE LOOP</p>
--

Table 1. PGSL algorithm

4.3 Focusing cycle

After each iteration of the probability updating cycle (n_{fc} iterations), the search becomes more intense in the interval containing the current best point. This is made by dividing the interval containing the best point on which the process is repeated.

4.4 Subdomain cycle

After each focusing cycle (n_{sdc} iterations), the search space is modified. For first iteration, the total search space is used. For the next iterations, the search is made on a new subdomain. The size of this sub domain decreases gradually during the process to reach the convergence to a “good point”. The size of the sub domains is modified by changing the maximum and the minimum of each variable.

4.5 Parameters

The parameters to be determined for PGSL algorithm are the number of iterations for each one of the four loops described in Table 1: n_{sdc} , n_{fc} , n_{puc} and n_s . In practice, the following values are used: $n_s=2$ and $n_{puc}=1$.

5 Application: size optimization of truss structures

In this section, two truss structures commonly used in literature are selected as benchmark to test efficiency of PGSL approach. The PGSL method has been interfaced with a finite element code.

5.1 Truss structures

Size optimization of truss structures deals with the determination of optimum values for member cross-section

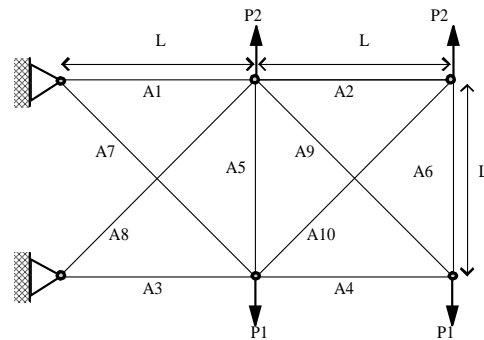
areas A_i that minimize the structural weight W . Thus, this size optimization problem with stress constraints may be formulated as follows:

$$\begin{cases} \min W(\mathbf{A}) = \sum_{i=1}^{i=n} \rho_i A_i l_i \text{ with } \mathbf{A} = (A_1, \dots, A_n) \\ \text{subject to } \begin{cases} A_{\min} \leq A_i \leq A_{\max} & \text{for } i = 1, \dots, n \\ \sigma_{\min} \leq \sigma_i \leq \sigma_{\max} & \text{for } i = 1, \dots, n \\ u_{\min} \leq u_j \leq u_{\max} & \text{for } j = 1, \dots, k \end{cases} \end{cases} \quad (2)$$

where n is the number of elements of the structure; k is the number of nodes; l_i is the length of the i^{th} element; ρ_i is density, and A_i is cross-section area of the i^{th} element. σ_i is the axial stress in the i^{th} bar and u_j is the nodal vertical displacement of node j .

5.1 Ten-bar truss

The first example considers the well-known problem corresponding to a ten-bar truss optimization. It has been previously studied using various optimization methods [13,7,14,15]. Fig. 1 shows the geometrical properties, loading and boundary conditions. The design variables are the cross-section areas of the 10 bars. Each member is subjected to stress limitations of ± 25 ksi (172.4 MPa) and a minimum area of 0.1 in^2 (64.5 mm^2) is assumed. The vertical displacement of all nodes is limited to ± 2.0 in (50.8mm).



$$E = 10^4 \text{ ksi } (6.89 \times 10^4 \text{ MPa})$$

$$\rho = 0.1 \text{ lb/in}^3 \text{ } (2.8 \times 10^3 \text{ kg/m}^3)$$

Case 1

$$P_1 = 100 \text{ kips } (444.8 \text{ kN})$$

$$P_2 = 0 \text{ kips}$$

Case 2

$$P_1 = 150 \text{ kips } (667 \text{ kN})$$

$$P_2 = 50 \text{ kips } (222.4 \text{ kN})$$

Fig. 1. Ten-bar truss structure

PGSL has been applied for both load cases. The maximum number of evaluations is fixed to 30000. It means 75 subdomain loops ($n_{sdc}=75$) and 400 evaluations in each subdomain loop ($n_s=2$; $n_{puc}=1$; $n_{fc}=200$).

In case 1, the best solution vector given by PGSL is

$$\mathbf{A} = (29.35, 0.1, 23.785, 15.113, 0.1, 0.513, 7.367, 21.037, 22.212, 0.1) \text{ in}^2$$

which corresponds to $W(\mathbf{A})=5064.2$ lb.

In case 2, the best solution vector given by PGSL is

$$\mathbf{A} = (23.761, 0.1, 25.184, 14.315, 0.1, 1.971, 12.522, 13.111, 19.889, 0.1) \text{ in}^2$$

which corresponds to $W(\mathbf{A})= 4678$ lb.

The optimization techniques are linked to the finite element method. In this context, the evaluation of the cost function for given values of the design variables requires a finite element analysis. This work can be very cpu time consuming especially when the finite element models are large and have a considerable number of design parameters. That is the reason why it is very important to control and to limit the number of evaluations of the cost function. The evolution of the cost function is shown in Fig. 2

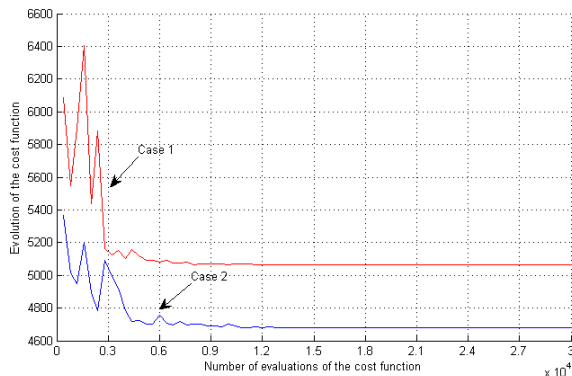


Fig. 2. Ten-bar truss, evolution of the cost function

Table 2. 10-bar truss: optimal design comparison for case 1

	HPSO	PSO	HS	PSACO	PGSL
A ₁	30.704	33.500	30.150	30.068	29.351
A ₂	0.100	0.100	0.102	0.100	0.100
A ₃	23.167	22.766	22.710	23.207	23.785
A ₄	15.183	14.417	15.270	15.168	15.113
A ₅	0.100	0.100	0.102	0.100	0.100
A ₆	0.551	0.100	0.544	0.536	0.513
A ₇	7.460	7.534	7.541	7.462	7.367
A ₈	20.978	20.467	21.560	21.228	21.037
A ₉	21.508	20.392	21.450	21.630	22.212
A ₁₀	0.100	0.100	0.100	0.100	0.100
W	5060.92	5024.21	5057.88	5057.36	5064.2

To conclude our study, we have compared our optimal design results, with other results from the literature: HPSO, heuristic particle swarm optimizer [14]; PSO, particle swarm optimization algorithm [15]; HS, harmony search meta-heuristic [7]; and PSACO, Particle Swarm Ant Colony Optimization [10]. The comparison is shown in Table 2 and Table 3 .

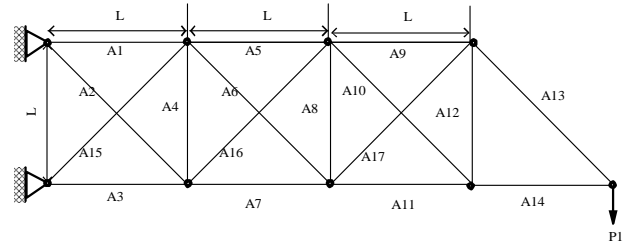
Table 3. 10-bar truss: optimal design comparison for case 2

	HS	HPSO	PSACO	PGSL
A ₁	23.25	23.353	23.052	23.7614
A ₂	0.102	0.100	0.100	0.10
A ₃	25.73	25.502	25.601	25.1842

A ₄	14.51	14.250	15.139	14.315
A ₅	0.100	0.100	0.100	0.100
A ₆	1.977	1.972	1.969	1.97127
A ₇	12.210	12.363	12.206	12.522
A ₈	12.610	12.894	12.568	13.1114
A ₉	20.36	20.356	20.330	19.8898
A ₁₀	0.100	0.101	0.100	0.100
W	4668.81	4677.29	4676.050	4678.0

5.3 Seventeen-bar truss

The second example deals with a seventeen-bar truss optimization problem. Fig. 3 shows the geometrical properties, loading and boundary conditions. The design variables are the cross-section areas of the 17 bars. Each member is subject to stress limitations of ± 50 ksi (344.7 MPa) and a minimum area of 0.1 in² (64.5mm²) is assumed. The vertical displacement of all nodes is limited to ± 2.0 in (50.8mm).



$$L = 100\text{in.}(2.5 \text{ m})$$

$$P_1 = 100 \text{ kips}(444\text{kN})$$

$$E = 3 \times 10^4 \text{ ksi}(2.1 \times 10^5 \text{ MPa})$$

$$\rho = 0.268 \text{ lb/in}^3 (7.4 \times 10^3 \text{ kg/m}^3)$$

Fig. 3. Seventeen-bar truss structure

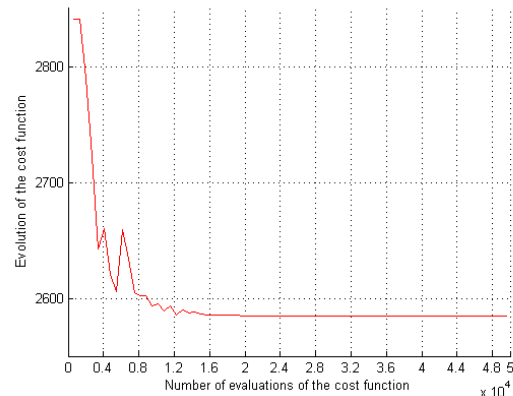


Fig. 4. 17-bar truss, evolution of the cost function

PGSL has been applied with a maximum number of evaluations fixed to 50000. It means 73 subdomain loops ($n_{\text{sdc}}=73$), 680 evaluations in each subdomain loop ($n_{\text{se}}=2$; $n_{\text{pic}}=1$; $n_{\text{r}}=340$). The best solution vector given by PGSL is

$A = (15.141, 0.151, 12.192, 0.1, 7.993, 5.760, 11.926, 0.104, 7.930, 0.103, 4.198, 0.103, 6.001, 3.917, 5.504, 0.116, 5.585) \text{in}^2$

which corresponds to

$$W(\mathbf{A}) = 2584.5 \text{ lb}$$

The evolution of the cost function is shown in Fig. 4.

To conclude our study, we have compared our optimal design, with other results from the literature: HPSO, heuristic particle swarm optimizer [14]; HS, harmony search meta-heuristic [7]. The comparison is shown in

Table 4.

Table 4. 17-bar truss: optimal design comparison			
	HS	HPSO	PGSL
A ₁	15.821	15.896	15.141
A ₂	0.108	0.103	0.148
A ₃	11.996	12.092	12.233
A ₄	0.100	0.100	0.100
A ₅	8.150	8.063	7.997
A ₆	5.507	5.591	5.747
A ₇	11.829	11.915	11.921
A ₈	0.100	0.100	0.100
A ₉	7.934	7.965	7.930
A ₁₀	0.100	0.100	0.100
A ₁₁	4.093	4.076	4.201
A ₁₂	0.100	0.100	0.103
A ₁₃	5.660	5.670	5.997
A ₁₄	4.061	3.998	3.921
A ₁₅	5.656	5.548	5.503
A ₁₆	0.100	0.103	0.119
A ₁₇	5.582	5.537	5.570
W	580.81	581.94	2584.5

6 Conclusion

Practical engineering problems require finite element simulations for the evaluation of the objective function. This step can be computationally very expensive. In this context, it seems interesting to explore alternative optimization algorithms, which can converge faster with least number of function evaluations and thereby improve the computational performances.

That is the reason why, we have used, in this paper, a global optimization method. PGSL method has been tested on simple structural optimization problems. Our results have been compared with the results obtained by others metaheuristics. Future work will be concerned with the continuation of the validation (for more complex problems) and with comparing these methods with the genetic algorithms and others new metaheuristic algorithms in order to choose the best optimization strategy for global design of structures.

References

1. B. Raphael and I.F.C. Smith, *Global search through sam-*

- pling using a PDF*, Computer Science, **2827**, 71-82, (2003).
2. M.P. Bendsøe and O. Sigmund, *Topology Optimization: Theory, Methods and Application*, Berlin: Springer Verlag, (2004).
3. P. Duysinx, *Optimisation Topologique: du Milieu Continu à la Structure Elastique*, PhD thesis, Université de Liège, (1996).
4. D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Pub. Co (ISBN: 0201157675), (1989).
5. S. Christodoulou, *In Optimal truss design using ant colony optimization*. In 5th GRACM International Congress on Computational Mechanics, (2005).
6. M. Kripka, *Discrete optimization of trusses by simulated annealing*, Journal of the Brazilian Society of Mechanical Sciences and Engineering, **26(2)**, (2004).
7. Kang Lee and Zong Woo Geem, *A new structural optimization method based on the harmony search algorithm*, Computers & Structures, **82**, 781-798, (2004).
8. M. Papadrakakis and N. D. Lagaros, *Soft computing methodologies for structural optimization*, Applied Soft Computing, **3(3)**, 283--300, (2003).
9. B. Sid, *Optimisation topologique des structures par algorithmes génétiques*. PhD thesis, UTBM/Université de Franche Comté, (2006).
10. B. Adam and I.F.C. Smith, *Tensegrity Active Control: Multiobjective Approach*, Journal of Computing in Civil Engineering, **21(1)**, 3-10, (2007).
11. B. Domer, B. Raphael, K. Shea and I.F.C. Smith, *A study of two stochastic search methods for structural control*, Journal of Computing in Civil Engineering, **17(3)**, 132-141, (2003).
12. B. Raphael and I.F.C. Smith, *A direct stochastic algorithm for global search*, Journal of Applied Mathematics and Computation, **146(2-3)**, 729-758, (2003).
13. U. Kirsch, *Structural Optimization. Fundamentals and applications*, Berlin, Springer Verlag, (1993).
14. L.J. Li and Z.B. Huang and F. Liu and Q.H. Wu, *A heuristic particle swarm optimizer for optimization of pin connected structures*, Computers & Structures, **85**, 340-349, (2007).
15. R.E. Perez and K. Behdinan, *Particle swarm approach for structural design optimization*, Computers & Structures, **85**, 1579-1588, (2007).
16. A. Kaveh and S. Talat Ahari, *Particle swarm and ant colony optimization for design of truss structures*, Asian Journal of Civil Engineering, **9(4)**, 329-348, (2008).