Received: 10 February 2017 | Revised: 16 August 2017 | Accepted: 7 October 2017
DOI: 10.1002/tal.1449

WILEY

RESEARCH ARTICLE

Constrained mean-variance mapping optimization for truss optimization problems

Mohamad Aslani[1] | **Parnian Ghasemi**[2] | **Amir H. Gandomi**[3]

[1]Department of Aerospace and Mechanical Engineering, Iowa State University, Ames, Iowa, USA
[2]Department of Civil Engineering, Iowa State University, Ames, Iowa, USA
[3]School of Business, Stevens Institute of Technology, Hoboken, New Jersey, USA

Correspondence
Amir H. Gandomi, School of Business, Stevens Institute of Technology, Hoboken, New Jersey 07030, USA.
Email: a.h.gandomi@stevens.edu

Summary

Truss optimization is a complex structural problem that involves geometric and mechanical constraints. In the present study, constrained mean-variance mapping optimization (MVMO) algorithms have been introduced for solving truss optimization problems. Single-solution and population-based variants of MVMO are coupled with an adaptive exterior penalty scheme to handle geometric and mechanical constraints. These tools are explained and tuned for weight minimization of trusses with 10 to 200 members and up to 1,200 nonlinear constraints. The results are compared with those obtained from the literature and classical genetic algorithm. The results show that a MVMO algorithm has a rapid rate of convergence and its final solution can obviously outperform those of other algorithms described in the literature. The observed results suggest that a constrained MVMO is an attractive tool for engineering-based optimization, particularly for computationally expensive problems in which the rate of convergence and global convergence are important.

KEYWORDS

exterior penalty, mean-variance mapping optimization, stochastic search, truss weight

1 | INTRODUCTION

Weight minimization of skeletal structures is historically used as a benchmark for heuristic and evolutionary optimization algorithms. The objective (representing the weight of the structure) and constraint(s) (i.e., mechanical properties of the structure such as nodal displacement, member stress,[1, 2] or vibration[3, 4]) lead to nonconvex systems with local optima. The set of design variables differs for each problem, and it includes (but is not limited to) the structure topology,[5-7] its shape,[8, 9] its size,[10, 11] and combinations of these factors.[12] To solve such a problem reliably and efficiently, researchers have developed specialized optimization algorithms,[10, 13] as well as hybridized and other current tuned techniques.[14-16]

Population-based evolutionary search algorithms such as genetic algorithm (GA),[17] particle swarm,[18] ant colony,[19] bee colony,[20] big-bang crunch,[21] krill-herd optimization,[22] firefly,[23] and cuckoo search[24] (see Ishibuchi et al. and Gandomi et al.[25, 26] for a review) have been employed in recent years to solve complex real-world problems. These algorithms utilize a variety of tools to explore favorable regions in the search domain and exploit them further to locate global minima. Change in an algorithm's behavior is sometimes accomplished manually (i.e., changing the parameters) or more rigorously through an adaptive strategy. Even though all these techniques are listed in the literature, nonconvex global optimization is still challenging. Due to the irregularities in either objectives, constraints or both that narrow down the margins of success for a specific algorithm; if an algorithm works correctly for one problem, it may not be suitable for another problem, so there has always been a desire for designing and implementing new ideas in this field.

Borrowing ideas of selection, mutation, and crossover from the concept of evolutionary algorithms and employing a mapping method, Elrich et al.[27] designed a new optimization algorithm, namely, mean-variance mapping optimization (MVMO). MVMO shares philosophical properties with population-based stochastic algorithms. For instance, elitism is intrinsic in an evolutionary algorithm, and MVMO similarly preserves an archive of the best points that implicitly resemble elitism. Moreover, the archive of *n*-best solutions provides guidelines for development of a mapping algorithm based on the mean and variance of the archive that the mapping function and upon which the new offspring (child) will essentially depend. Given the mean, variance, and a parameter called a shape variable, a transformation can be constructed and, as explained later, this transformation automatically switches the behavior of the algorithm, that is, it will explore and exploit the search domain. Finally, the elitism criteria dictate that the parent is the first-ranked solution in the archive and a new generation is born following the constructed mapping function. One of the important features of MVMO is its rapid rate of convergence while at the same time carrying out a global search. This feature is especially important for problems with expensive function evaluations where a moderately good solution is assumed to be acceptable.[28, 29]

MVMO has been initially designed to work with a single solution on a normalized domain (or $x_i \in [0, 1]$). However, if multiple sessions of MVMO run simultaneously, a population-based approach is achieved.[30] With extra information available from unfavorable regions, a multiparent crossover can occur to support further exploration of these regions. This extra effort has been shown to make the algorithm robust and reliable in the sense of a general optimization algorithm.[30-32]

MVMO is an algorithm for unconstrained problems, but the performance of MVMO has never been tested in a real-world application where multiple nonlinear constraints might exist, and the convergence rate is important. For the truss structure considered in this paper, these constraints are nodal displacements and member stresses, and if a solution point violates any of them, a penalty is applied through an adaptive quadratic function. Therefore, in this paper, performances of constrained single-solution and population-based MVMO are evaluated to find the global weight minimization of truss structures. Parameters of the algorithm (MVMO and penalty function) are tuned, and the results are presented for a variety of population sizes (linearly proportional to the number of variables).

The remainder of this paper is organized as follows: In Section 2, two variants of MVMO (single-solution and population-based) are presented, followed by an adaptive penalty function to handle constraints. In Section 3, the weight minimization problem of truss structures is formally defined, and finally, Section 4 describes the results for a variety of cases.

2 | MEAN-VARIANCE MAPPING OPTIMIZATION

2.1 | Single-solution MVMO

In contrast to many evolutionary algorithms that utilize a population of points, MVMO is originally introduced as a single-solution algorithm. The internal search range of all variables is restricted to $x_i \in [0, 1]$. Therefore, the desired range for a general problem is scaled to the unity length.

The distinctive property of MVMO is the ability to carry out a global search using the best-so-far solutions. Figure 1a depicts the properties of such a search for two variables in which red circles indicate the newly generated points. For a conservative search, most of the points will concentrate around the mean value. However, several points will try a broader search region. For an exploration type of search, fewer points are concentrated around the mean as shown in Figure 1b. These types of search are controlled by the values chosen for shape function.

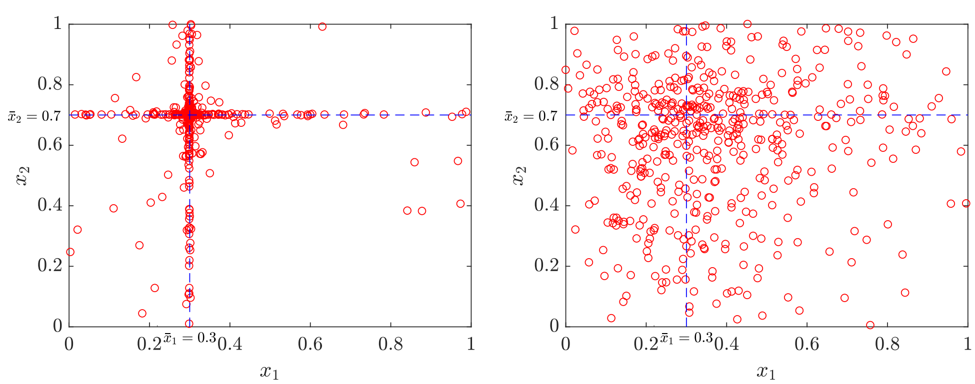
A particular form of a mapping function, which depends on mean value and shape variables, is central to understanding the mechanism of MVMO. The mapping function is used to transfer a random point in [0, 1] to a point x_i or $M : x^{rand} \mapsto x_i$. This mapping occurs in a fashion that the new point is located in the vicinity of the mean value with specific criteria (see Figure 2a). The mean value $(\bar{x}_i)$ and shape variable (s_i) are calculated using the archive of the best points with the size n:

$$\bar{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_i^{(j)}, \qquad s_i = -f_s \ln(v_i), \tag{1}$$

where v_i is the variance of data in the set (for each variable) or

$$v_i = \frac{1}{n}\sum_{j=1}^{n}\left(x_i^{(j)} - \bar{x}_i\right)^2, \tag{2}$$

where $f_s > 0$ is a coefficient and (j) is an indicator of points in the archive. The s_i is used to find the vector of shape variable $\mathbf{s}_i = (s_{i1}, s_{i2})$ for the *ith* variable in the problem. The algorithm to find $\mathbf{s}_i$ vector will be explained later. We also note that because v_i is less than one, s_i is always positive.



(a) $s_1 = s_2 = 20$ (b) $s_1 = s_2 = 5$

FIGURE 1 Mapped points for (a) a conservative search with a large value for shape variables and (b) an exploration search with small value for shape variables

The inputs of mapping function are x^{rand} and a function, H, which depends on the mean value and shape variables or:

$$\begin{aligned} x_i &= M(x^{rand}, H) \\ &= H(\bar{x}_i, \mathbf{s}_i, x^{rand}) + (1 - H_1 + H_0)\, x^{rand} - H_0, \end{aligned} \tag{3}$$

where function H depends on the mean value, shape variables, and x:

$$H(\bar{x}_i, \mathbf{s}_i, x) = \bar{x}_i(1 - e^{-xs_{1i}}) + (1 - \bar{x}_i)e^{-(1-x)s_{2i}}, \tag{4}$$

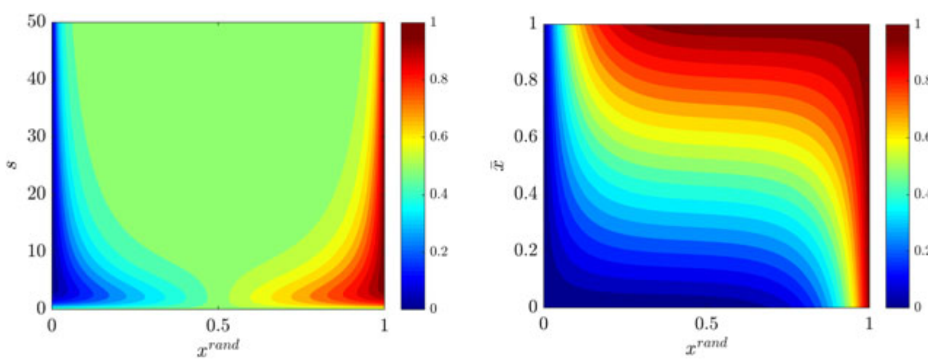
along with the definitions for $H_0 \equiv H(\bar{x}_i, \mathbf{s}_i, 0)$ and $H_1 \equiv H(\bar{x}_i, \mathbf{s}_i, 1)$.

Contour plot of Figure 2a shows the mapping function for a constant mean value of 0.5. As it is shown, if the random value is around $\bar{x}$, then the mapped variable remains constant, independent of values for s. However, the mapped variables change drastically for small values of s when the random point is away from the mean value. Also, Figure 2b shows the result of the mapping function for constant shape variables.

The shape variable vector $\mathbf{s}_i$ consists of two shape variables, that is, $\mathbf{s}_i = (s_{1i}, s_{2i})$. One of the shape variables is always set to s_i, which is given in Equation 1, and the other one is a search distance d_i, chosen upon the value of s_i. If s_i is greater than this distance, then d_i is reduced. Otherwise, it will be increased. Initially, this distance is set to a big number and $s_{i1} = s_{i2} = s_i$. Numerical experiments have shown that the range of $d_i^{initial} \in [1 - 5]$ is big enough to result in an acceptable range for variables. The algorithm to find $\mathbf{s}_i = (s_{1i}, s_{2i})$ is summarized as

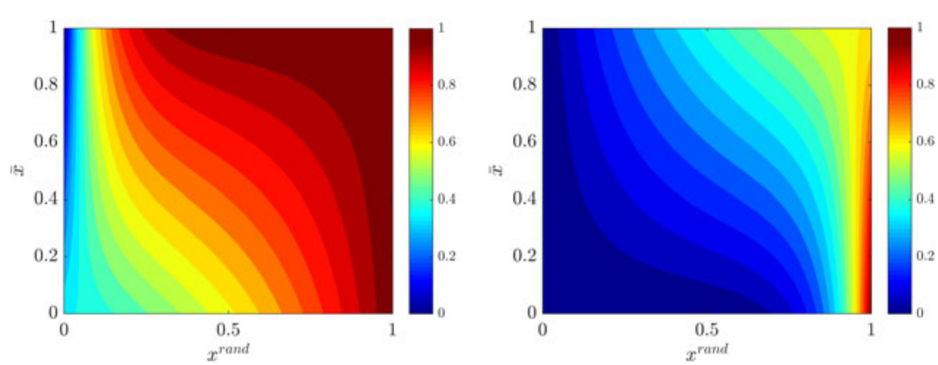
IF $(s_i > d_i)$ **THEN**$d_i = d_i \times \Delta d$
ELSE$d_i = d_i / \Delta d$
END IF
IF (***rand*** ≥ 0.5) **THEN**$\mathbf{s}_i = (s_i, d_i)$
ELSE$\mathbf{s}_i = (d_i, s_i)$
END IF.

If shape variables are equal, then the mapping function is symmetric. This is shown in Figure 2b. Asymmetric mapping is achieved once shape variables are not equal as illustrated by the contours in Figure 3a,b. Also, as $\mathbf{s}_i \to (\infty, \infty)$, the new point converges to the mean value $\bar{x}_i$. The flowchart of the single-solution MVMO is summarized in Figure 4.



(a) $\bar{x} = 0.5$ (b) $s_1 = s_2 = 10$

FIGURE 2 Mapping function for (a) constant mean and (b) symmetric mapping when shape variables are equal



(a) $s_1 = 10, s_2 = 1$ (b) $s_1 = 1, s_2 = 10$

FIGURE 3 Asymmetric mapping when shape variables are not equal

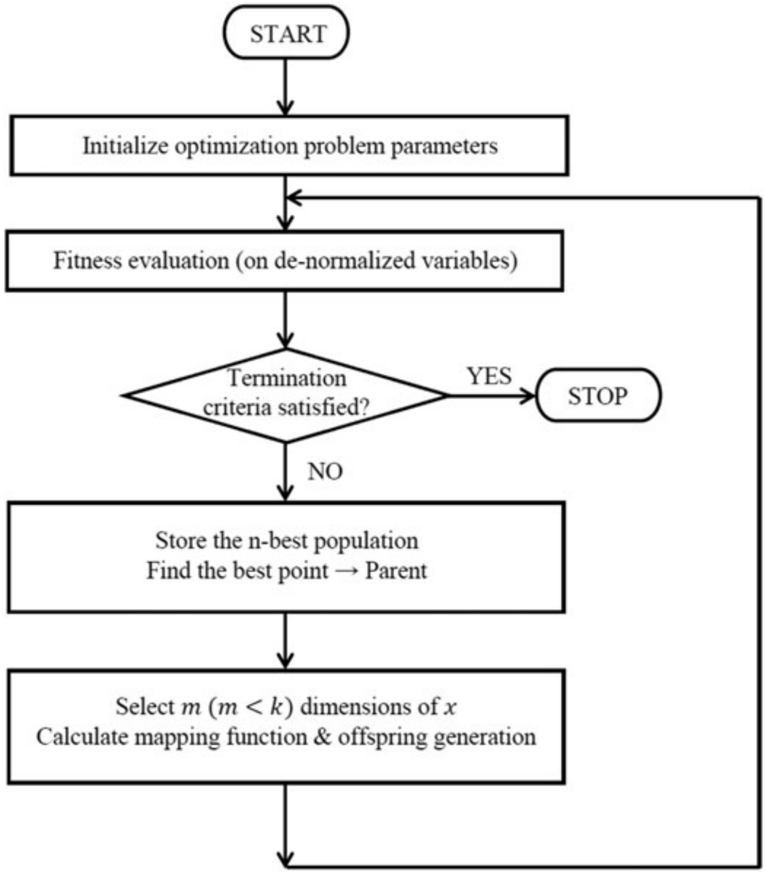


FIGURE 4 Flowchart for the single-solution mean-variance mapping optimization algorithm

Once mean, variance, and shape variables are constructed, a number of variables (number of mapped variables, *m*) are chosen randomly to be varied based on the mapping function. This number introduces another parameter into MVMO algorithm. Thus, the user needs to tune the algorithm based on the problem in hand with the following parameters:

- Size of archive (n)
- Number of mapped variables (m)
- Factor (f_s)
- Increment (Δd_i)
- Number of iterations (N_{tot})

Appropriate assignment of these parameters mimics both exploration and exploitation of an evolutionary algorithm with implicit elitism. Numerical experiments show that an archive size of $4 \leq n \leq 6$ is appropriate for the problems of interest in this study. Generally, the larger archive size corresponds to a more conservative search, and as a result, it requires more computation. Also, factor f_s in Equation 1 will make a conservative search (when accuracy is needed) if $f_s > 1$ and opposite (global search) if $f_s < 1$. Ideally, an adaptive strategy is required, which is the topic of next section.

2.2 | Adaptive strategies for f_s and Δd_i

Appropriate values for f_s and Δd_i are not straightforward to set. Both of them determine the properties of shape variables that are used in the mapping algorithm. Ideally, it is required that $f_s < 1$ initially and gradually becomes greater than one as the algorithm evolves. For this purpose, the following strategy is adopted:

$$f_s = f_s^{min} + \mu^2 (f_s^{max} - f_s^{min}), \tag{5}$$

where $f_s^{min} = 0.8$, $f_s^{max} = 2$ and $\mu = \#iteration/N_{tot}$. The above functionality allows a global search initially and a quadratic increase in f_s. In the original algorithm, a random number in the range of [0, 1] was used. However, numerical experiments showed that a progressive increase of f_s as in Equation 5 gives decent results as well.