

Design Optimization Average-Based Algorithm

João Barradas Cardoso and Artur Barreiros

Department of Mechanical Engineering, Instituto Superior Técnico, University of Lisbon

Av Rovisco Pais 1, 1049-001 Lisbon, Portugal

E-mail: barreiros@tecnico.ulisboa.pt, <https://tecnico.ulisboa.pt/en/>

Keywords: heuristic, randomness, population-based, optimization

Received: October 6, 2017

This article introduces a metaheuristic algorithm to solve engineering design optimization problems. The algorithm is based on the concept of diversity and independence that is aggregated in the average design of a population of designs containing information dispersed through a variety of points, and on the concept of intensification represented by the best design. The algorithm is population-based, where the population individual designs are randomly generated. The population can be normally or uniformly generated. The algorithm may start either with points randomly generated or with a designer preferred trial guess. The algorithm is validated using standard classical unconstrained and constrained engineering optimum design test problems reported in the literature. The results presented indicate that the proposed algorithm is a very simple alternative to solve this kind of problems. They compare well with the analytical solutions and/or the best results achieved so far. Two constrained problem analytical solutions not found in the literature are presented in annex.

Povzetek: Članek uvaja metahevristični algoritem za reševanje problemov optimizacije. Algoritem temelji na populaciji, kjer se naključno generirajo posamezni modeli posameznika in končno agregirajo v primerne rešitve.

1 Introduction

With the advent of fast, cheap and reliable computing power over the last decades, in addition to the application of classic optimization to larger and larger size problems, new alternative algorithms operating in a different fashion have been developed. The classical optimization algorithms have shortcomings and are not suitable for all optimization problems. The new alternative algorithms allow attacking optimization problems either too costly or not applicable to classical algorithms.

The purpose of heuristic algorithms applied to optimization problems is to search a solution to them by trial-and-error in a satisfying amount of computing time. The optimum solution is not guaranteed but a near optimum solution is accepted as a good solution. Metaheuristics refers to higher-level algorithms combining lower-level techniques and tactics for exploration and exploitation of the design space. That is, these algorithms, on one hand, must be able to generate a range of points in the whole design space including potentially optimum ones; on the other hand, they intensify the search around the neighborhood of an optimum or near optimum points (Yang, 2008).

Exploration and exploitation are the two important components of the metaheuristic algorithms. They are also called diversification and intensification (Glover, Laguna, 1997). A good balance of these components is required. Too much weight in diversification risks slow convergence with the solutions jumping around the potentially optimum ones; too much weight in intensification restricts the design space to a local region and risks the convergence to a local optimum (Blum,

Roli, 2003). The heuristic algorithms start typically either with guess solutions randomly generated or with a designer preferred trial solution. The diversification is gradually reduced as the algorithm proceeds; simultaneously, the intensification is increased.

One of the roles of injected randomness in stochastic search is to allow for movements to unexplored areas of the search space that may contain an unexpectedly good design. This is especially relevant when the search is stalled near a local solution. Injected randomness may also be used for the creation of simple random quantities that act like their deterministic counterparts, but which are much easier to obtain and more efficient to compute.

Metaheuristics algorithms are either population-based or trajectory-based. Examples of the former ones are the genetic algorithms (Holland, 1975) or the particle swarm optimization (Cagnina, Esquivel, Coello, 2008; Kayham, Ceylan, Ayvaz, Gurarslan, 2010) the last ones include the simulated annealing optimization (Kirkpatrick, Gelatt, Vecchi, 1983) or the harmonic search (Geem, Kim, Loganathan, 2001).

If a sufficient numerous and diversified group of people is asked to decide on subjects of general interest, the decisions of the group are better than the decisions that an isolated individual would take (Surowiecki, 2005). Well informed and sophisticated an expert is, his or her advice and predictions should be pooled with those of others to get the most out of him or her. Practical examples, simple and complex, are described in (Surowiecki, 2005), remarking the principle of group think, and the concept that the masses are better problem

solvers, forecasters, and decision makers than any one individual. A classic example of group intelligence is the jelly-beans-in-the-jar experiment, in which invariably the group's average estimate for the number of the jelly beans in the jar is superior to the vast majority of the individual guesses.

The theory that groups are remarkably intelligent and often smarter than the smartest people in them, demonstrates the significant impact on how businesses operate, how knowledge is increased, how economies are structured, and how people live their daily lives (Williams, 2006). The necessary conditions for a crowd to be wise include diversity, independence, and a specific type of decentralization. These conditions are essential to making good decisions which are the result of disagreement and contest rather than consensus or compromise.

Diversity means individuals have some private information or their own interpretation of known facts. Diversity helps because it actually adds perspectives that would otherwise be absent and because it takes away, or at least weakens, some of the destructive characteristics of group decision-making. Homogeneous groups are great at doing what they do well, but they become progressively less able to investigate alternatives. Independence means freedom from the influence of others. It keeps the individual mistakes from becoming correlated. Diversity is essential to preserving this independence. In the jelly-beans-in-the-jar experiment, most group members are not talking to each other or solving problems together.

Decentralization means people draw on local knowledge. It encourages individuals to make important decisions, not just in one location based only on one specific type of information, but dispersed through a variety of locations from where local knowledge is drawn and shared. The information coming out of a decentralized group must be aggregated throughout the system, to maintain a balance between local and global counterparts. Aggregation needs a mechanism that turns individual judgments into a collective decision. For instance, in a free market, the aggregating mechanism is price; in the jelly-beans-in-the-jar experiment, individual guesses were aggregated and then averaged, i.e., the aggregating mechanism is the average guess.

The aim of the present article is the proposal of an algorithm based on the concept just described. The algorithm considered is stochastic in the sense that it relies on random numbers and that different results may be obtained upon running the algorithm repeatedly. The algorithm is population-based, where the population individual designs are randomly generated. These individuals are diversified and independent since their design variables values are chosen stochastically without any correlation.

The individual designs are also decentralized since the design variables are chosen all over the entire design space. Finally, the different values of the design variables are aggregated as the plain or weighted average of those values. However, generating a diverse set of possible solutions isn't enough. The designer, as the body of

people, also has to be able to distinguish the good solutions for the bad. So, at each of the iterations, the present algorithm selects two designs: the best design, the one with the best objective so far; and the averaged design, the one which design variable values are the mean of those variable values for the iteration.

The best design represents the intensification component of the algorithm; the averaged design represents the diversity part. In the current article, the optimization problem is understood as a minimization problem, where the function of merit to be assessed is an extended cost function that takes in account penalization due to violated constraints. A reference design is considered as a linear combination of both the best and the averaged design variables. A simple recurrence formula, centered on the reference design, is used to actualize the design for the next randomly generated population. The population can be normally or uniformly generated.

In optimization, there is traditionally a concern with developing a good stopping criterion. Unfortunately, the quest for an automatic means of stopping an algorithm with a guaranteed level of accuracy seems doomed to failure in general stochastic search problems. The fundamental reason is that in nontrivial problems, there will always be a significant region within the design space that will remain unexplored in any finite number of iterations, and there is always the possibility the optimum could lie in this unexplored region.

A danger arises in making broad claims about the performance of an algorithm based on the results of numerical studies. Performance can vary tremendously under even small changes in the form of the functions involved or the coefficient settings within the algorithms themselves. Outstanding performance on some types of functions is consistent with poor performance on other types of functions. This is a manifestation of no free lunch theorems (Spall, 2003; Wolpert, Macready, 1997).

The present algorithm is applied to several constrained and unconstrained test functions as well as to typical engineering design problems.

2 The optimal design problem

The optimal design problem may be formulated in a generalized fashion as

$$\begin{aligned} & \min_{\mathbf{b}} \Psi_0(\mathbf{b}) \\ & \text{s.t.} \\ & \Psi_j(\mathbf{b}) \leq 0; \quad j = 1, 2, \dots, m \\ & b_{il} \leq b_i \leq b_{iu}; \quad i = 1, 2, \dots, n \end{aligned} \quad (1)$$

where Ψ_0 is the cost or objective function, Ψ_j are the constraints in a dimensionless form, $\mathbf{b} \equiv (b_1, b_2, \dots, b_n)$ is the design vector, b_{il} and b_{iu} are respectively the lower and upper bounds of the design variable b_i .

In order of applying the present algorithm, the formulation of the Eq. (1) is replaced by the following one:

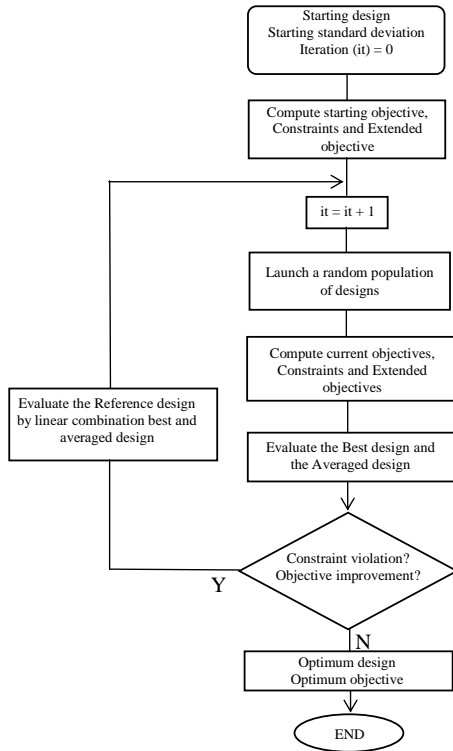


Figure 1: Flowchart of the algorithm.

$$\begin{aligned} \min_b \bar{\Psi}_0 &\equiv \Psi_0 + |\Psi_0| P \\ \text{s.t.} \quad & b_i \leq b_i \leq b_{iu}; \quad i = 1, 2, \dots, n \end{aligned} \quad (2)$$

In Eq. (2), P stands for the penalty factor which value depends on the violation of the constraints as

$$P = \begin{cases} 0, & \text{for satisfied constraints} \\ \sum_{v=1}^{m'} \alpha \Psi_v / m', & \text{for violated constraints} \end{cases} \quad (3)$$

where $\alpha > 0$ and m' is the number of violated constraints. In practical terms, the constraint Ψ_j can be considered violated if $\Psi_j > \varepsilon$, with ε a very small number. Note that the absolute value of the objective is considered in the Eq. (2) in order to accommodate problems with negative objective functions.

θ	Starting Point			
	(-2.5, -2.5)	(-2.5, 2.5)	(2.5, -2.5)	(2.5, 2.5)
1.00	(0.08986,-0.71283) -1.0316283 at iteration 38	(0.09140,-0.71266) -1.0316191 at iteration 39	(-0.09162,0.71276) -1.0316163 at iteration 36	(0.08887,-0.71247) -1.0316248 at iteration 31
0.85	(0.08979,-0.71267) -1.0316285 at iteration 15	(0.08997,-0.71265) -1.0316285 at iteration 21	(-0.08991,0.71265) -1.0316285 at iteration 21	(-0.08976,0.71272) -1.0316285 at iteration 15
0.25	(0.08973,-0.71262) -1.0316285 at iteration 36	(-0.08992,-0.71259) -1.0316285 at iteration 29	(0.08974,-0.71262) -1.0316285 at iteration 26	(-0.08973,0.71262) -1.0316285 at iteration 37
0.00	(0.09002,-0.71401) -1.0316136 at iteration 989	(-0.09020,0.71196) -1.0316237 at iteration 617	(0.09020,-0.71196) -1.0316237 at iteration 617	(-0.09002,0.71401) -1.0316136 at iteration 989

Table 1: Optimum points and function values for different θ s and starting points.

3 The average concept algorithm

The present algorithm may be described by the flowchart of Fig. 1 as well as in the following manner:

1. Start with a preference design guess \mathbf{b} and an imposed standard deviation, estimated as $s_i = b_{iu} - b_{il}$ ($i = 1, 2, \dots, n$). Compute the starting values of Ψ_0 , Ψ_j ($j = 1, 2, \dots, m$) and $\bar{\Psi}_0$, and establish the starting reference design \mathbf{b}^R as the preference guess, i.e. $\mathbf{b}^R \equiv \mathbf{b}$.

Start iterates as

2. Launch a normally or uniformly random population of N designs as

$$b_i^k = b_i^R + s_i r_i^k \quad (4)$$

where $i = 1, 2, \dots, n$, $k = 1, 2, \dots, N$ and r_i^k is the k -th random number (with mean value 0 and standard deviation 1) related to the design variable b_i .

Do $b_i = b_{il}$ if $b_i < b_{il}$ and $b_i = b_{iu}$ if $b_i > b_{iu}$.

3. Evaluate Ψ_0 , Ψ_j ($j = 1, 2, \dots, m$) and $\bar{\Psi}_0$ for the entire population of N designs.
4. Evaluate the best design \mathbf{b}^B corresponding to the minimum value $\bar{\Psi}_0^{\min}$ of the extended function.

Evaluate the averaged design \mathbf{b}^A for the distribution of designs as

$$\mathbf{b}^A = \sum_{k=1}^N \mathbf{b}^k p_k / \sum_{k=1}^N p_k \quad (5)$$

where \mathbf{b}^k is an arbitrary design vector in the population and p_k is the weight accounted for design \mathbf{b}^k on the average. The weights are selected by the designer. If a plain average is chosen, then $p_k = 1$ for every design.

Another choice used in this work is to compare the

values of the extended function $\bar{\Psi}_0$ and of the best extended function $\bar{\Psi}_0^{\min}$ in the previous iteration; and then assign a weight $p_k = 2$ if the first value is smaller than the second one, and a weight $p_k = 1$ if it is equal or larger.

5. If there are no constraint violations and there are no improvements of the objective function within a prescribed number of iterations, go to 7.
6. Evaluate a reference design as the linear combination

$$\mathbf{b}^R = \theta \mathbf{b}^B + (1 - \theta) \mathbf{b}^A \tag{6}$$

with $0 \leq \theta \leq 1$ and assume the distribution of designs \mathbf{b}^k centered at \mathbf{b}^R with a standard deviation vector evaluated as $s = 2|\mathbf{b}^A - \mathbf{b}^B|$.

Go to 2.

End the iterates.

7. Stop.

In order of handling tabular discrete value design variables, the Eq. (4) is rewritten in these cases as

$$b_i^k = \text{int} \left(\frac{b_i^R + s_i r_i^k}{\Delta} \right) \Delta \tag{7}$$

where Δ is the difference value between two consecutive design variables.

4 Numerical applications

In this Section one is going to solve unconstrained as well as constrained optimization problems by applying the formulation and the algorithm presented on the previous sections. With respect to the unconstrained problems, the applications are the minimizations of the benchmark following test functions: Six-Hump Camelback function, Rosenbrock and Michalewicz's functions.

Concerning to the constrained problems, one solves three well-known engineering design optimization test problems: the welded beam design, the pressure vessel design and the tension-compression spring design. For all the applications normally distributed populations were used. A total of 30 independent runs were performed per problem. In the Sections 4.1 to 4.6 the runs of the algorithm were performed with a particular initial seed for different population sizes and different weights on the calculation of the average and reference designs.

The results are compared with analytical solutions and/or heuristic and nonlinear programming algorithms.

4.1 Six-Hump Camelback function

This function is one of the typical test functions in unconstrained global optimization (Dixon, Szego, 1975; Lee, Geem, 2005). It is mathematically expressed as

$$\Psi_0 = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4 \tag{8}$$

Within the bounded region, this function has six local minima. Two of them are global minima located at either $(-0.08984, 0.71266)$ or $(0.08984, -0.71266)$, each with the corresponding function value equal to $\Psi_0^* = -1.0316285$.

For the algorithm presented here, a population of size $N = 20$ and a plain average in the step 4 were used within the design space $-2.5 \leq x_1, x_2 \leq 2.5$. For different values of the θ -factor and two starting points, the Table 1 gives the corresponding achieved solutions and the number of iterations needed for convergence.

We should note that the best and fastest solutions are obtained with $\theta = 0.85$, i.e., giving respectively at each iteration, the weights 0.85 and 0.15 to the best and average points of the prior iteration in the formation of the current reference design.

With the starting point $(2.5, 2.5)$, corresponding to the cost value $\Psi_0 = 161.8489685$, and with $\theta = 0.85$, the present algorithm achieves the minimum cost value $\Psi_0^* = -1.0316285$ at the point $(-0.08976, 0.71272)$ after 15 iterations. However, after 9 iterations, the algorithm gets the cost value $\Psi_0 = -1.0316285$ at the point $(-0.08799, 0.71382)$, not far off the analytical solution.

We should also note that the worst results exist for $\theta = 0$, i.e., only considering the average design as the design of reference. For $\theta = 1$, respecting to the selection at each iteration of the best design as the reference one, the solutions are also not so good

For $N = 20$, $\theta = 0.85$, starting point $(2.5, 2.5)$ and the weighted average (second) option on step 4 described in Chapter 3 instead of the plain average, the global optimum $\Psi_0^* = -1.0316285$ is obtained at the point $(-0.08988, 0.71266)$ after 19 iterations.

Better results can be expectedly obtained by increasing the population size. For $N = 100$, the best and fastest solutions are obtained with $\theta = 0.85$ after 10 iterations at the points $(-0.08982, 0.71264)$ and $(0.08983, -0.71264)$ respectively starting from the points $(-2.5, 2.5)$ and $(2.5, -2.5)$, corresponding both to the minimum function value $\Psi_0^* = -1.0316285$. For

N	IP	Iterate	x_1	x_2	Ψ_0
1000	0	4	-0.0897507	0.7126449	-1.0316285
	1	5	-0.0898749	0.7127146	-1.0316285
100	0	13	-0.0897433	0.7126887	-1.0316285
	1	11	-0.0898545	0.7126102	-1.0316285
20	0	15	-0.0897623	0.7127175	-1.0316285
	1	19	-0.0898786	0.7126648	-1.0316285

Table 2: Camelback's optimal solutions for $\theta = 0.85$ and starting at $(2.5, 2.5)$.

$N=1000$ and $\theta=0.85$, and starting from the point $(-2.5, -2.5)$, the optimum solution $\mathbf{x}^* = (0.08975, -0.71264)$, $\Psi_0^* = -1.0316285$ has been achieved after 4 iterations.

Table 2 shows the optimal solutions achieved for different N population sizes and different average choices on Eq. (5), $IP=0$ standing for plain average and $IP=1$ meaning weighted average as written on step 4 of the algorithm described on Chapter 3, using $\theta=0.85$ and with the starting point located at $(2.5, 2.5)$.

We may note that is the population size $N=20$ the one that needs the smallest number of function evaluations (20×15) .

4.2 Rosenbrock’s function

Another classical test function in unconstrained optimization is the Rosenbrock’s function, which two-dimensional form (Moré, Garbow, Hillstrom, 1981; Rosenbrock, 1960; Yang, 2008) is

$$\Psi_0 = (1 - x_1)^2 + 100(x_2 - x_1^2)^2 \quad (9)$$

This function also referred to as the Valley or Banana function due to the shape of its contour lines, is a popular test problem for gradient-based optimization algorithms. The function turns out to be quite challenging to find its minimum point by numerical methods.

Its global optimum point is $\mathbf{x}^* = (1.0, 1.0)$ that gives the optimum cost of $\Psi_0^* = 0.0$. The function is unimodal and its analytical solution can be obtained straightforwardly by partial differentiation. The numerical solution, however, poses a particular challenge. The solution lies inside a very deep, narrow, banana shaped valley. The valley causes a lot of troubling for nonlinear programming search algorithms.

Using a population of 1000 samples within the design space $-10.0 \leq x_1, x_2 \leq 10.0$, a plain average, the factor $\theta=0.85$ and a starting point $\mathbf{x} = (0.0, 0.0)$ corresponding to $\Psi_0 = 1.0000000$, the present algorithm achieves the optimum point $\mathbf{x}^* = (1.00000, 1.00000)$ and the corresponding $\Psi_0^* = 0.0000000$ after 18 iterations. However, the point $\mathbf{x} = (1.00051, 1.00087)$ corresponding to $\Psi_0^* = 0.0000029$ was obtained after 9 iterations. If $N=100$, the same optimum point above is obtained after 38 iterations. For $N=20$, the algorithm converges towards the same optimum point after 2604 iterations. One may say that is the population of 100 samples that gives the shortest number of function evaluations (100×38) .

4.3 Michalewicz’s function

The third unconstrained optimization problem uses the Michalewicz’s function in its two-dimensional form

$$\Psi_0 = -\sum_{i=1}^2 \sin(x_i) \left[\sin(i x_i^2 / \pi) \right]^{20} \quad (10)$$

with $0 \leq x_1, x_2 \leq \pi$.

The function is tricky; it has several local minimum values and several flat areas which make the one global minimum value hard to find numerically. Its global minimum is $\Psi_0^* = -1.8012980$ at the point $\mathbf{x}^* = (2.20319, 1.57049)$.

Using a population with size $N=20$ samples within the design space $0.0 \leq x_1, x_2 \leq 4.0$, a plain average, the factor $\theta=0.85$, and a starting point $\mathbf{x} = (2.0, 2.0)$ corresponding to $\Psi_0 = -0.3701513$, the present algorithm achieves the optimum point $\mathbf{x}^* = (2.20281, 1.57079)$ and the corresponding cost value $\Psi_0^* = -1.8013037$ after 24 iterations. However, after 13 iterations the algorithm achieves the point $\mathbf{x} = (2.20258, 1.56925)$ corresponding to the objective value $\Psi_0 = -1.8012055$. Changing the population size to $N=100$, the global minimum $\Psi_0^* = -1.8013039$ is achieved after 10 iterations at the point $\mathbf{x}^* = (2.20289, 1.57091)$. Changing now the population size to $N=1000$, the global minimum $\Psi_0^* = -1.8013039$ is achieved after 8 iterations at the point $\mathbf{x}^* = (2.20292, 1.57076)$. The smallest number of function evaluations (20×24) is obtained for $N=20$.

4.4 Welded beam design

The welded beam design problem is well studied in the context of single-objective optimization. A beam A needs to be welded on another beam B and must carry a certain load P as shown in Fig. 2. The welded beam is designed for minimum fabrication cost subject to constraints on

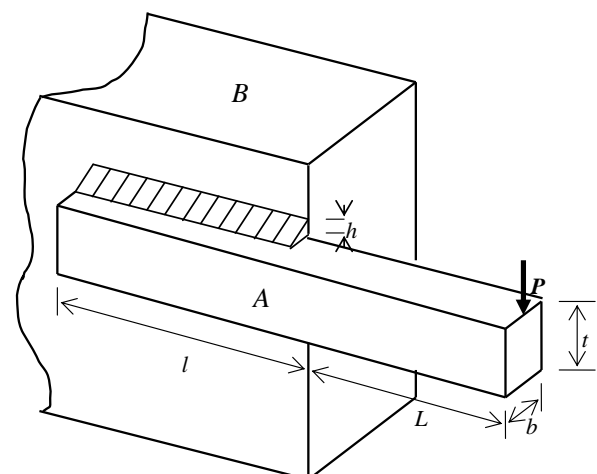


Figure 2: Welded beam structure.

shear stress τ , bending stress in the beam σ , buckling load on the bar P_c , end deflection of the beam δ , cost of the weld and beam A materials, and side constraints (Ragsdell, Phillips, 1975; Rao, 1996). One wants to find four design parameters: thickness of the beam b , width of the beam t , length of the weld l , and thickness of the weld h .

In order of formulating the problem in a standard form, let $\mathbf{x} \equiv (x_1, x_2, x_3, x_4) = (h, l, t, b)$ be the design vector. Then, our problem may be described as

$$\begin{aligned} \min \Psi_0 &\equiv (C_1 + C_2)x_1^2 x_2 + C_3 x_3 x_4 (L + x_2) \\ \text{s.t.} \\ \Psi_1 &\equiv \tau / \tau_{\max} - 1 \leq 0 \\ \Psi_2 &\equiv \sigma / \sigma_{\max} - 1 \leq 0 \\ \Psi_3 &\equiv x_1 / x_4 - 1 \leq 0 \\ \Psi_4 &\equiv [C_1 x_1^2 + C_3 x_3 x_4 (L + x_2)] / 5 - 1 \leq 0 \quad (11) \\ \Psi_5 &\equiv \delta / \delta_{\max} - 1 \leq 0 \\ \Psi_6 &\equiv 1 - P_c / P \leq 0 \\ 0.125 &\leq x_1 \leq 2 \\ 0.1 &\leq x_4 \leq 2 \\ 0.1 &\leq x_1, x_2 \leq 10 \end{aligned}$$

where $C_1 = 0.10471 \text{ \$/in}^3$ is the cost per unit volume of

the weld material, $C_2 = 1 \text{ \$/in}^3$ is the labor cost per unit weld volume, $C_3 = 0.04811 \text{ \$/in}^3$ is the cost per unit volume of the beam B , $\tau_{\max} = 13,600 \text{ psi}$, $\sigma_{\max} = 30,000 \text{ psi}$ and $\delta_{\max} = 0.25 \text{ in}$. The other parameters are defined as:

$$\begin{aligned} \tau &= \sqrt{\tau_1^2 + \tau_1 \tau_2 x_2 / R + \tau_2^2}, \quad \tau_1 = P / (\sqrt{2} x_1 x_2), \\ \tau_2 &= MR / J, \quad M = P (L + x_2 / 2), \\ R &= 0.5 \sqrt{x_2^2 + (x_1 + x_3)^2}, \quad \sigma = 6PL / (x_3^3 x_4), \\ \delta &= 4PL^3 / (E x_3^3 x_4), \quad J = (\sqrt{2} / 6) x_1 x_2 [x_2^2 + 3(x_1 + x_3)^2], \\ P_c &= 4.013 (E / L^2) (x_3 x_4^3 / 6) [1 - (0.5 x_3 / L) \sqrt{0.25 E / G}], \\ E &= 30 \times 10^6 \text{ psi}, \quad G = 12 \times 10^6 \text{ psi}, \quad P = 6,000 \text{ lb}, \\ L &= 14 \text{ in} \end{aligned}$$

By using mathematical programming, (Rao, 1996) presents the optimum cost function $\Psi_0^* = 2.3810$ corresponding to the design point $\mathbf{x}^* = (0.2444, 6.2177, 8.2915, 0.2444)$. A better nonlinear programming solution has been achieved in (Andrei, 2013) by adding GAMS created nonlinear model: $\Psi_0^* = 1.72485$, $\mathbf{x}^* = (0.206, 3.470, 9.037, 0.206)$.

The lowest optimal solution known so far by the

N	IP	Iterate	x_1	x_2	x_3	x_4	Ψ_0
5000	0	16645	0.205729	3.470519	9.036630	0.205730	1.724858
		268	0.205468	3.476687	9.038588	0.205723	1.725573
		32	0.205436	3.477702	9.034963	0.205964	1.726863
		16	0.204191	3.530001	9.042473	0.205769	1.731815
	1	11144	0.205726	3.470586	9.036639	0.205730	1.724864
		303	0.204957	3.488440	9.035478	0.205797	1.726386
		57	0.205409	3.475931	9.042488	0.205808	1.726697
		19	0.204431	3.523721	9.041501	0.205707	1.730704
1000	0	9434	0.205727	3.470567	9.036656	0.205730	1.724866
		27	0.204332	3.495317	9.051999	0.205779	1.729056
		19	0.204359	3.503654	9.084193	0.205596	1.734414
	1	10419	0.205730	3.470530	9.036634	0.205730	1.724866
		1335	0.205707	3.470647	9.038382	0.205760	1.725373
		522	0.205451	3.477778	9.038218	0.205742	1.725777
		37	0.205635	3.479632	9.039730	0.205804	1.727052
		21	0.206227	3.492924	9.013144	0.206816	1.732874
100	0	15080	0.205727	3.470617	9.036695	0.205729	1.724877
		1596	0.205693	3.470650	9.039433	0.205731	1.725315
		205	0.205585	3.477111	9.031481	0.206261	1.728674
		177	0.206150	3.503281	9.041299	0.206170	1.734152
	1	27689	0.205724	3.470646	9.036589	0.205731	1.724869
		672	0.204555	3.495600	9.037673	0.205735	1.726630
		300	0.205584	3.488843	9.039643	0.205969	1.729467
		141	0.202780	3.547630	9.027128	0.206246	1.732925
20	0	11462	0.205731	3.470480	9.036694	0.205735	1.724909
		2265	0.205546	3.475157	9.036914	0.205765	1.725515
		714	0.207129	3.459350	8.988936	0.208278	1.736547
	1	22011	0.205726	3.470606	9.036623	0.205730	1.724868
		2788	0.205607	3.474374	9.036784	0.205731	1.725223
		115	0.202686	3.587231	9.037897	0.205726	1.736020

Table 3: Welded beam optimization solutions.

N	IP	Iterate	x_1	x_2	x_3	x_4	Ψ_0
1000	0	4143	0.81250	0.43750	42.09843	176.63712	6059.7231
	1	13441	0.81250	0.43750	42.09843	176.63678	6059.7158
500	0	8285	0.81250	0.43750	42.09843	176.63701	6059.7212
	1	11909	0.81250	0.43750	42.09844	176.63673	6059.7168
200	0	4173	0.81250	0.43750	42.09838	176.63742	6059.7227
	1	5077	0.81250	0.43750	42.09843	176.63704	6059.7222

Table 4: Pressure vessel optimal solutions.

authors is given in (Kayham, Ceylan, Ayvaz, Gurarslan, 2010): $\mathbf{x}^* = (0.205830, 3.468338, 9.036624, 0.205730)$ and $\Psi_0^* = 1.724717$.

The optimization results achieved with the present algorithm are shown in Table 3 for different population sizes N and different average choices on Eq. (5). The first row for each choice combination of N and IP represents the optimum at the convergence of the algorithm. The following rows are the results at intermediary iterations. For all these design points all the constraints are satisfied.

The best minimum cost value obtained in the present article is $\Psi_0^* = 1.724858$, corresponding to the point $\mathbf{x}^* = (0.205729, 3.470519, 9.036630, 0.205730)$.

One may observe that the algorithm solutions compare very well with the best solution presented above, even for earlier iterates of the algorithm. We may also observe that convergence is faster when the plain average is used in Eq. (5).

4.5 Pressure vessel design

The pressure vessel design problem has been proposed in (Kannan, Kramer, 1994). It is one of the most used test problems for validating optimization algorithms. The problem is to find the optimal design of a compressed air storage tank (Fig. 3) with a working pressure of 1000 psi and a minimum capacity volume of $V_{\min} = 1,296,000 \text{ in}^3$.

The pressure vessel is composed of a cylindrical shell capped at both ends by hemispherical heads.

Let the design variables be $x_1 \equiv T_s$ the thickness of the shell, $x_2 \equiv T_h$ the thickness of the heads, $x_3 \equiv R$ the inner radius and $x_4 \equiv L$ the length of the cylindrical shell. The variables x_1 and x_2 should be integer multiples of 0.0625 in. The objective is to minimize the manufacturing cost (material, welding and forming costs) of the pressure vessel (Sandgren, 1990), subjected to constraints on volume capacity and in accordance with

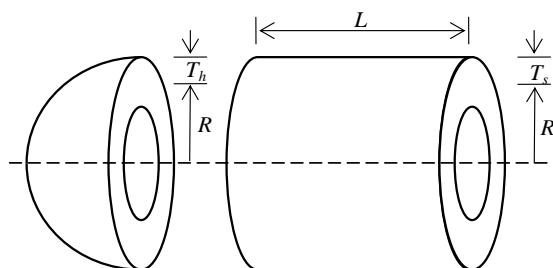


Figure 3: Pressure vessel.

respective ASME codes. The mathematical model of the problem is:

$$\min \Psi_0 \equiv 0.6224 x_1 x_3 x_4 + 1.7781 x_2 x_3^2 + 3.1661 x_1^2 x_4 + 19.84 x_1^2 x_3$$

s.t.

$$\Psi_1 \equiv 0.0193 x_3 / x_1 - 1 \leq 0$$

$$\Psi_2 \equiv 0.00954 x_3 / x_2 - 1 \leq 0 \tag{12}$$

$$\Psi_3 \equiv V_{\min} / V - 1 \leq 0$$

$$V = \pi (x_3^2 x_4 + 4 x_3^3 / 3)$$

$$0.0625 \leq x_1, x_2 \leq 99 \times 0.0625$$

$$10 \leq x_3, x_4 \leq 200$$

The analytical optimum for this problem is calculated in the Annex A as $\Psi_0^* = 6059.714335$ at $\mathbf{x}^* = (0.8125, 0.4375, 42.0984456, 176.6365958)$ with the first and third constraints being active.

The optimal results achieved with the present algorithm are shown in Table 4 for different population sizes N and different average choices on Eq. (5). For all these solutions there is no violation of the constraints. The first constraint is nearly active at the optimal point for all the different population sizes; it takes values in the interval $-0.3666 \times 10^{-7} \leq \Psi_1^* \leq -0.1215 \times 10^{-5}$.

The results for the present algorithm compare well with the analytical solution. Again, the plain average of the Eq. (5) gives origin to faster convergence.

4.6 Tension/Compression spring design

The tension/compression spring design optimization problem is described in (Arora, 1989). The goal is to minimize the weight of a tension/compression spring (Fig. 4) subject to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter and side constraints. The design variables to be considered are the wire diameter d , the mean coil diameter D and the number n of active coils.

Let us set up the vector of design variables as $\mathbf{x} \equiv (x_1, x_2, x_3) \equiv (d, D, n)$.

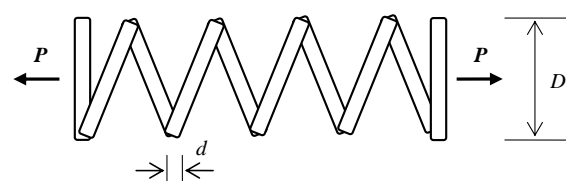


Figure 4: Tension-compression spring.

The problem may be formulated as

$$\begin{aligned}
 \min \Psi_0 &\equiv (x_3 + 2)x_1^2 x_2 \\
 \text{s.t.} \\
 \Psi_1 &\equiv 1 - x_2^3 x_3 / (71785 x_1^4) \leq 0 \\
 \Psi_2 &\equiv (4x_2^2 - x_1 x_2) / [12566(x_2 x_1^3 - x_1^4)] + \\
 &\quad 1 / (5108 x_1^2) - 1 \leq 0 \\
 \Psi_3 &\equiv 1 - 140.45 x_1 / (x_2^2 x_3) \leq 0 \\
 \Psi_4 &\equiv (x_1 + x_2) / 1.5 - 1 \leq 0 \\
 0.05 &\leq x_1 \leq 2 \\
 0.25 &\leq x_2 \leq 1.3 \\
 2 &\leq x_3 \leq 15
 \end{aligned} \tag{13}$$

The analytical solution for this problem is presented in the Annex B. The minimum weight of the spring is achieved as $\Psi_0^* = 0.012665232$ at the point $\mathbf{x}^* = (0.051690, 0.356740, 11.287642)$. At the optimum, the constraints Ψ_1 and Ψ_2 are active, and $\Psi_3^* = -4.054$, $\Psi_4^* = -0.7277$.

The minimum objective function value obtained in (Arora, 1989) by nonlinear programming is $\Psi_0^* = 0.012679$ corresponding to the optimum point $\mathbf{x}^* = (0.051699, 0.35695, 11.289)$. The best result known so far by the authors is given in (Cagnina, Esquivel, Coello, 2008) as $\Psi_0^* = 0.012665$, with $\mathbf{x}^* = (0.051583, 0.354190, 11.438675)$.

The optimum results achieved with the present algorithm are shown in Table 5, for $IP=0$, different sizes N of the population and plain average selection on Eq. (5). For all these solutions there is no violation of the constraints, being practically active the first two constraints Ψ_1 and Ψ_2 . The last two constraints have values within the intervals $-4.08 \leq \Psi_3^* \leq -4.04$ and $-0.734 \leq \Psi_4^* \leq -0.720$.

Again, the present algorithm results compare well with the analytical solution.

5 Concluding remarks

This article presents an average concept algorithm to solve various optimization problems which include typical benchmark functions unconstrained problems and structural engineering test design constrained problems.

N	1000	500	200	100
Ψ_0	0.012665	0.012666	0.012668	0.012671
x_1	0.051718	0.051775	0.051308	0.052127
x_2	0.357409	0.358778	0.347629	0.367321
x_3	11.248693	11.169669	11.842695	10.694988
Iterate	10063	11690	24818	13899

Table 5: Tension-compression spring optimal solutions.

To evaluate the performance of the present algorithm, numerical applications are conducted and the results are compared to the results obtained analytically and/or to the best ones achieved by other optimization methods. The analytical solutions for two of the constrained problems, namely the pressure vessel design and the tension-compression spring design problems, are determined in the present article. The solutions found by the proposed algorithm compare well with those results.

We may conclude that the algorithm finds the global solution or a near-global solution in each problem tested. The Six-Hump Camelback function, for example, has 6 local minimal points; however, the algorithm converges to the global minima.

Characterizing the principal advantage of the algorithm one should emphasize the good balance between the accuracy of the solutions it achieves and its rare simplicity.

References

- [1] Andrei, N. (2013) Nonlinear Optimization Applications Using the GAMS Technology. Springer Science & Business Media. <https://doi.org/10.1007/978-1-4614-6797-7>
- [2] Arora, J. S. (1989) Introduction to Optimum Design. McGraw-Hill, New York. <https://doi.org/10.1016/C2013-0-15344-5>
- [3] Blum, C., Roli, A. (2003) Metaheuristics in Combinatorial Optimization: overview and conceptual comparison. ACM Computing Surveys. 35, 268-308. <https://dl.acm.org/doi/pdf/10.1145/937503.937505>
- [4] Cagnina, L.C., Esquivel, S.C., Coello, C.A.C. (2008) Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer, Informatica. 32, 319-326.
- [5] Dixon, L.C.W., Szego, G.P. (1975) Towards Global Optimization. North Holland.
- [6] Geem, Z.W., Kim, G.H., Loganathan, G.V. (2001) A New Heuristic Optimization Algorithm: Harmonic Search. Simulation. 76, 60-68. <https://doi.org/10.1177/003754970107600201>
- [7] Glover, F., Laguna, M. (1997) Tabu Search. Kluwer Academic Publishers. https://doi.org/10.1007/978-1-4613-0303-9_33
- [8] Holland, J.H. (1975) Adaptation in Natural and Artificial Systems. The University of Michigan Press: Ann Arbor.
- [9] Kannan, B.K., Kramer, S.N. (1994) An Augmented Lagrange Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and its Applications to Mechanical Design. Journal of Mechanical Design. 116, 318-320. <https://doi.org/10.1115/1.2919393>
- [10] Kayham, A.H., Ceylan, H., Ayvaz, M.T., Gurarlan, G. (2010) PSOLVER: A New Hybrid Particle Swarm Optimization Algorithm for Solving Continuous Optimization Problems. Expert Systems with Applications. 37, 6798-6808. <https://doi.org/10.1016/j.eswa.2010.03.046>

[11] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. (1983) Optimization by Simulated Annealing. *Science*. 220, 671-680.
<https://doi.org/10.1126/science.220.4598.671>

[12] Lee, K.S., Geem, Z.W. (2005) A New Meta-Heuristic Algorithm for Continuous Engineering Optimization: Harmony Search Theory and Practice. *Computer Methods in Applied Mechanical Engineering*. 194, 3902–3933.
<https://doi.org/10.1016/j.cma.2004.09.007>

[13] Luenberger, D.G. (1984) *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, Massachusetts.
<https://doi.org/10.1007/978-0-387-74503-9>

[14] Moré, J.J., Garbow, B.S., Hillstom, K.E. (1981) Testing Unconstrained Optimization Software. *ACM Transactions on Mathematical Software*. 7, (1), 17- 41.
<https://doi.org/10.1145/355934.355936>

[15] Papalambros, P.Y., Wilde, D.J. (1988) *Principles of Optimal Design: modeling and computation*. Cambridge University Press, New York.
<https://doi.org/10.1017/9781316451038>

[16] Ragsdell, K. M., Phillips, D.T. (1975) Optimal Design of a Class of Welded Structures Using Geometric Programming. *ASME J. Eng. Ind. Ser. B*. 98 (3), 1021–1025.
<https://doi.org/10.1115/1.3438995>

[17] Rao, S.S. (1996) *Engineering Optimization Theory: and Practice*. John Wiley & Sons.
<https://doi.org/10.1002/9780470549124>

[18] Rosenbrock, H.H. (1960) An Automatic Method for Finding the Greatest or Least Value of a Function. *Computer Journal*. 3, 175-184.
<https://doi.org/10.1093/comjnl/3.3.175>

[19] Sandgren, E. (1990) Nonlinear Integer and Discrete Programming in Mechanical Design Optimization. *Journal of Mechanical Design*. 112 (2), 223–229.
<https://doi.org/10.1115/1.2912596>

[20] Spall, J.C. (2003) *Introduction to Stochastic Search and Optimization: estimation, simulation and control*. John Wiley&Sons.
<https://doi.org/10.1002/0471722138>

[21] Surowiecki, J. (2005) *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*. Anchor Books.

[22] Williams, S. (2006) *The Wisdom of Crowds: why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations*. *Business Book Review™*. 21 (43).

[23] Wolpert, D.H., Macready, W.G. (1997) No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*. 1, 67-82.
<https://doi.org/10.1109/4235.585893>

[24] Yang, X.S. (2008) *Nature-inspired Metaheuristic Algorithms*. Luniver Press.

Annex A: Pressure Vessel Classical Design Optimization

Let us assume initially the variables x_1 and x_2 are continuous and later on make the convenient correction to table (multiple of 0.0675) values.

The cost function Ψ_0 decreases monotonically with x_1 or x_2 decrease. The constraint Ψ_1 is the only constraint that increases as x_1 decreases, and the constraint Ψ_2 is the only constraint that increases as x_2 decreases. Then, Ψ_1 and Ψ_2 provide respectively lower bounds for x_1 and x_2 , and these variables may be minimized out as

$$\begin{cases} x_1 = 0.0193x_3 \\ x_2 = 0.00954x_3 \end{cases} \quad (A1)$$

Substituting these relationships into the original problem, we have

$$\begin{aligned} \min \bar{\Psi}_0 &\equiv 0.01319166x_3^2x_4 + 0.024353275x_3^3 \\ \text{s.t.} \\ \bar{\Psi}_3 &\equiv 1296000 - \pi x_3^2x_4 - (4/3)\pi x_3^3 \leq 0 \\ 10 &\leq x_3, x_4 \leq 200 \end{aligned} \quad (A2)$$

where the upper bars on the cost and constraint symbols mean we are determining by now the solution for all variables continuous. The Lagrangian function for this problem is

$$L = \bar{\Psi}_0 + \lambda_3 \bar{\Psi}_3 - \mu_3^-(x_3 - 10) + \mu_3^+(x_3 - 200) - \mu_4^-(x_4 - 10) + \mu_4^+(x_4 - 200) \quad (A3)$$

where λ_3 , μ_3^- , μ_3^+ , μ_4^- , μ_4^+ are the Lagrange multipliers for the constraint $\bar{\Psi}_3$ and side constraints. The necessary Karush-Kuhn-Tucker conditions (Arora, 1989) for the problem (A.2) may now be set as

$$\begin{aligned} \partial L / \partial x_3 &\equiv 2 \times 0.01319166x_3x_4 + 3 \times 0.024353275x_3^2 \\ &\quad - 2\pi\lambda_3(x_3x_4 + 2x_3^2) - \mu_3^- + \mu_3^+ = 0 \\ \partial L / \partial x_4 &\equiv 0.01319166x_3^2 - \pi\lambda_3x_3^2 - \mu_4^- + \mu_4^+ = 0 \\ \lambda_3 \bar{\Psi}_3 &\equiv \lambda_3 [1296000 - \pi x_3^2x_4 - (4/3)\pi x_3^3] = 0 \\ \mu_k^-(x_k - 10) &= 0, \quad \mu_k^+(x_k - 200) = 0, \quad k = 3, 4 \\ \lambda_3, \mu_3^-, \mu_3^+, \mu_4^-, \mu_4^+ &\geq 0 \\ \bar{\Psi}_3 &\equiv 1296000 - \pi x_3^2x_4 - (4/3)\pi x_3^3 \leq 0 \\ 10 &\leq x_3, x_4 \leq 200 \end{aligned} \quad (A4)$$

Let us now search the different combinations of Lagrange multipliers:

1. $\mu_3^- > 0, \mu_4^- > 0 \Rightarrow$
 $x_3 = x_4 = 10, \Psi_3 = 1288669.6$ (violated)
 $\mu_3^- > 0, \mu_4^+ > 0 \Rightarrow$
 $x_3 = 10, x_4 = 200, \Psi_3 = 1228979.4$ (violated)

Then, whatever the value of λ_3 , must have $\mu_3^- = 0$

2. $\lambda_3 = 0$
 - 2.1 $\mu_3^- = 0 \Rightarrow \mu_3^+ < 0$ (from the 1st condition, then violating the 5th one)
 - 2.2 $\mu_4^- = 0 \Rightarrow \mu_4^+ < 0$ (from the 2nd condition, then violating the 5th one)

3. $\lambda_3 > 0 \Rightarrow \Psi_3 = 0 \therefore x_4 = 1296000 / (\pi x_3^2) - (4/3)x_3$
 - 3.1 $\mu_3^+ = \mu_4^- = \mu_4^+ = 0 \Rightarrow \lambda_3 = 0.01319166 / \pi$
 (from the 2nd condition) and $x_3 = 0$ (after substituting x_4 and λ_4 into the 1st one)
 - 3.2 $\mu_3^+ = \mu_4^- = 0, \mu_4^+ > 0 \Rightarrow x_4 = 200$ (from the 4th condition),

$$1296000 - \pi x_3^2 x_4 - (4/3)\pi x_3^3 = 0 \Rightarrow$$

$$x_3 = 40.3196187244$$

$$\lambda_3 = \frac{2 \times 0.01319166 x_3 x_4 + 3 \times 0.024353275 x_3^2}{2\pi(x_3 x_4 + 2x_3^2)}$$

$$= 0.004663057579 > 0$$

$$\mu_4^+ = \pi x_3^2 \lambda_3 - 0.01319166 x_3^2$$

$$= 2.369851195 > 0$$

All the optimality necessary conditions are satisfied, then

$$\begin{cases} \bar{x}_1 = 0.0193x_3 = 0.778168646 \\ \bar{x}_2 = 0.00954x_3 = 0.384649165 \\ \bar{x}_3 = 40.3196187244 \\ \bar{x}_4 = 200 \end{cases}$$

is candidate local optimum point for the assumed continuous variables.

- 3.3 $\mu_3^+ = \mu_4^+ = 0, \mu_4^- > 0 \Rightarrow x_4 = 10$ (from the 4th condition),

$$1296000 - \pi x_3^2 x_4 - (4/3)\pi x_3^3 = 0 \Rightarrow$$

$$x_3 = 65.2252326139$$

$$\lambda_3 = \frac{2 \times 0.01319166 x_3 x_4 + 3 \times 0.024353275 x_3^2}{2\pi(x_3 x_4 + 2x_3^2)}$$

$$= 0.005698937505 > 0$$

$$\mu_4^- = 0.01319166 x_3^2 - \pi x_3^2 \lambda_3$$

$$= -20.04674872 < 0$$

(violates 5th condition)

$$3.4 \mu_4^- = \mu_4^+ = 0, \mu_3^+ > 0 \Rightarrow x_3 = 200 \Rightarrow$$

$$x_4 = -256.3534264 < 0$$

$$3.5 \mu_4^- = 0, \mu_3^+ > 0, \mu_4^+ > 0 \Rightarrow x_3 = x_4 = 200 \Rightarrow$$

$$\Psi_3 = -57347062.87 \neq 0$$

(contradicts point 3: $\Psi_3 = 0$)

$$3.6 \mu_4^+ = 0, \mu_3^+ > 0, \mu_4^- > 0 \Rightarrow x_3 = 200, x_4 = 10$$

$$\Rightarrow \Psi_3 = -33470958.7 \neq 0$$

(contradicts point 3: $\Psi_3 = 0$)

Testing now the second-order sufficient conditions for the only point $\bar{x} \equiv (\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4)$, determined in 3.3, satisfying the necessary conditions, one may use the so-called bordered Hessian (Luenberger, 1984) calculated at that point:

$$\mathbf{B}(\bar{x}_3, \bar{x}_4) \equiv \begin{bmatrix} 0 & \partial\Psi_3/\partial x_3 & \partial\Psi_3/\partial x_4 \\ \partial\Psi_3/\partial x_3 & \partial^2 L/\partial x_3^2 & \partial^2 L/\partial x_3 \partial x_4 \\ \partial\Psi_3/\partial x_4 & \partial^2 L/\partial x_3 \partial x_4 & \partial^2 L/\partial x_4^2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -71095.91969 & -5107.198124 \\ -71095.91969 & 232.2341915 & -0.117553254 \\ -5107.198124 & -0.117553254 & 0 \end{bmatrix}$$

As $n - m$ for the problem (A.2) is $2 - 1 = 1$ one has to calculate the last principal minor $\det(\mathbf{B})$. Since its value is negative, its sign is coincident with $sign(-1)^m = sign(-1)$. Hence, the Hessian of L is positive-definite and the point \bar{x} is a minimum point.

Now, rounding up the values of \bar{x}_1 and \bar{x}_2 to the table values, we have

$$\begin{cases} x_1^* = 0.8125 \\ x_2^* = 0.4375 \end{cases}$$

then determine the other two variables as

$$x_3^* = \min\{0.8125/0.0193, 0.4375/0.00954\}$$

$$= 42.0984456$$

From the two first constraints $x_3 \leq 0.8125/0.0193$, $x_3 \leq 0.4375/0.00954$, i.e., the constraint Ψ_1 is active at the optimum, and

$$x_4^* = 176.6365958$$

from the condition of $\Psi_3 = 0 \therefore$

$$x_4 = 1296000 / (\pi x_3^2) - (4/3)x_3$$

Therefore, the analytical optimum point for the pressure vessel design problem is

$$\mathbf{x}^* = (0.8125, 0.4375, 42.0984456, 176.6365958)$$

giving the minimum optimum cost $\Psi_0^* = 6059.714335$. At the optimum, the constraints have $\Psi_1^* = \Psi_3^* = 0$ and $\Psi_2^* = -0.082013323$.

Annex B: Tension-Compression Spring Classical Design Optimization

Again, let us firstly to analyze the monotonicity of the problem (Papalambros, Wilde, 1988). One should observe the constraint Ψ_1 is critical with respect to the design variable x_3 . The cost function Ψ_0 increases monotonically in the variable x_3 , and there is exactly one constraint, the constraint Ψ_1 , whose monotonicity with respect to x_3 is opposite from that of the objective. Then, Ψ_1 is active and bounds x_3 from below:

$$x_3 = 71785 x_1^4 / x_2^3 \tag{B1}$$

Substituting the relationship (B.1) into the objective function and into the constraint Ψ_3 we get

$$\begin{cases} \Psi_0 = 2 x_1^2 x_2 + 71785 x_1^6 / x_2^2 \\ \Psi_3 \equiv 1 - 0.001956536881 x_2 / x_1^3 \leq 0 \end{cases} \tag{B2}$$

Substituting the lower and upper bounds of x_2 into the constraint Ψ_3 we have that $0.25 x_1 \leq 0.078790891$, $1.3 x_1 \leq 0.136503503$; then the range of the design variable x_1 can be set up as

$$0.05 \leq x_1 \leq 0.136503503$$

If one uses now the upper bounds of x_1 and x_2 into the constraint Ψ_4 , $0.136503503 + 1.3 \leq 1.5$, it is obvious that this constraint is always inactive, not playing any role into the optimization problem.

Studying now the monotonicity of the objective expressed in (B.2) with respect to the design variable x_2 , the minimum of Ψ_0 is given as

$$\begin{aligned} \partial \Psi_0 / \partial x_2 &\equiv 2 x_1^2 - 2 \times 71785 x_1^6 / x_2^3 = 0 \quad \therefore \\ x_2 &= \sqrt[3]{71785 x_1^4} \end{aligned}$$

since $\partial^2 \Psi_0 / \partial x_2^2 \equiv 6 \times 71785 x_1^6 / x_2^4 > 0$. Thus, Ψ_0 decreases monotonically in x_2 increase for $0.25 \leq x_2 \leq \sqrt[3]{71785 x_1^4}$ and increases monotonically in x_2 increase for $\sqrt[3]{71785 x_1^4} \leq x_2 \leq 1.3$. For example, within the range $0.25 \leq x_2 \leq \sqrt[3]{71785 x_1^4}$, the function Ψ_0 decreases in the variable x_2 increase, achieving the minimum at $x_2 = 0.765545910$ for a prescribed $x_1 = 0.05$, and increases the value of the minimum point at x_2 as x_1 increases. For $x_1 \geq 0.074378786$ the function

Ψ_0 is a decreasing function all along the feasible domain of x_2 , $0.25 \leq x_2 \leq 1.3$.

The constraint Ψ_2 may be expressed as

$$\begin{aligned} 4 x_2^2 + (C - 1) x_1 x_2 - C x_1 &\leq 0, \\ C &= 2.460062647 - 12566 x_1^2 \end{aligned}$$

This constraint increases monotonically in x_2 increase; then its monotonicity with respect to x_2 is opposite from that of Ψ_0 for $0.25 \leq x_2 \leq \sqrt[3]{71785 x_1^4}$ and the constraint Ψ_2 is critical providing an upper bound for x_2 :

$$x_2 = (x_1 / 8) \left(1 - C + \sqrt{C^2 + 14C + 1} \right)$$

The reduced problem may now be expressed as

$$\begin{aligned} \min \Psi_0 &\equiv 2 x_1^2 x_2 + 71785 x_1^6 / x_2^2 \\ \text{s.t.} & \\ x_2 &= (x_1 / 8) \left(1 - C + \sqrt{C^2 + 14C + 1} \right) \\ 0.05 \leq x_1 &\leq 0.136503503, \quad 0.25 \leq x_2 \leq 1.3 \end{aligned} \tag{B3}$$

The optimum point can be determined easily by using a unidimensional search in x_1 , with the active constraint Ψ_2 determining x_2 . The variable x_3 is calculated by using the relationship (B.1) after solving the problem (B.3).

The optimum value of the objective function is obtained as $\Psi_0^* = 0.012665232$ at the point $x^* \equiv (0.051690, 0.356740328, 11.28764160)$. At the optimum, the constraints have the values $\Psi_1^* = 0$, $\Psi_2^* = 0$, $\Psi_3^* = -4.05383024$, $\Psi_4^* = -0.727713114$.

