26 Mechanical engineering problem optimization by SOMA

Ivan Zelinka and Jouni Lampinen

26.1 Mechanical engineering problem optimization by SOMA

To discover the effectiveness of the techniques just proposed in Chapter 7, three numerical examples were optimized using SOMA (Table 26.1). These non-linear, engineering design optimization problems with discrete, integer and continuous variables were first investigated by Eric Sandgren [1] and subsequently by many other researchers [2], [3], [4], [5], [6], [7], [8], [9], [10], [11] and [12] who applied a variety of optimization techniques (Table 26.2). These problems represent optimization situations involving discrete, integer and continuous variables that are similar to those encountered in everyday mechanical engineering design tasks. Because the problems are clearly defined and relatively easy to understand, they form a suitable basis for comparing alternative optimization methods

Table 26.1. Test problems

Summary of Test Problems					
Example	Description	Number of variables			
		Total	Discrete	Integer	Continuous
1	Gear Train	4	0	4	0
2	Pressure Vessel	4	2	0	2
3	Coil Spring	3	1	1	1

G. C. Onwubolu et al., *New Optimization Techniques in Engineering*

Table 26.2. Alternative methods used to solve the test problems.

Compared Methods		
Reported by	Solution technique	Reference
Sandgren	Branch & Bound using Sequential Quadratic Programming	[1]
Fu, Fenton & Gleghorn	Integer-Discrete-Continuous Non-Linear Programming	[2]
Loh & Papalambros	Sequential Linearization Algorithm	[3], [4]
Zhang & Wang	Simulated Annealing	[5]
Chen & Tsao	Genetic Algorithm	[6]
Li & Chow	Non-Linear Mixed-Discrete Programming	[7]
Wu & Chow	Meta-Genetic Algorithm	[8]
Lin, Zhang & Wang	Modified Genetic Algorithm	[9]
Thierauf & Cai	Two-level Parallel Evolution Strategy	[10]
Cao & Wu	Evolutionary Programming	[11]
Lampinen & Zelinka	Differential Evolution	[12]
Zelinka &Lampinen	SOMA	This article

26.1.1 Designing a gear train

In the first example the problem is to optimize the gear ratio for the compound gear train arrangement shown in Fig. 26.1. The gear ratio for a reduction gear train is defined as the ratio of the angular velocity of the output shaft to that of the input shaft. In order to produce the desired overall gear ratio, the compound gear train is constructed out of two pairs of gearwheels, *d-a* and *b-f*. The overall gear ratio, i_{tot}, between the input and output shafts can be expressed as:

$$i_{tot} = \frac{\omega_o}{\omega_i} = \frac{z_d z_b}{z_a z_f} \tag{26.1}$$

Variables ω_o and ω_i are the angular velocities of the output and input shafts, respectively, and z denotes the number of teeth on each gearwheel.

The optimization problem is to find the number of teeth for gearwheels *d, a, b* and *f* in order to produce a gear ratio, i_{tot}, as close as possible to the target ratio: i_{trg} = 1/6.931 (= 0.1443). For each gear, the minimum number of teeth is 12 and the maximum is 60.

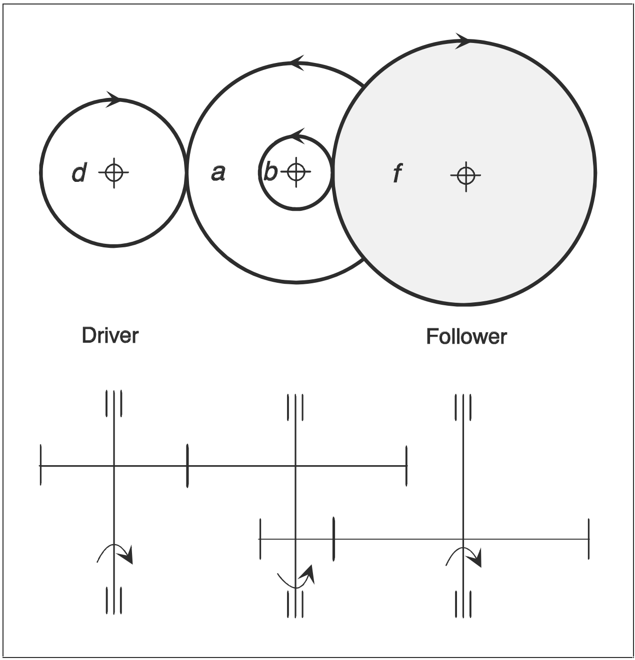

Fig. 26.1. Compound gear train for Example 1.

The problem is formulated as follows:

$$
\begin{aligned}
&\text{Find} \\
&X = (x_1, x_2, x_3, x_4) = (z_d, z_b, z_a, z_f), \quad x \in \{12, 13, \ldots, 60\} \\
&\text{to minimize} \\
&f(X) = (i_{trg} - i_{tot})^2 = \left(\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4} \right)^2 \\
&\text{subject to} \\
&12 \le x_i \le 60 \;, \qquad i = 1,2,3,4.
\end{aligned}
\tag{26.2}
$$

Thus, the goal is to find optimum values for four integer variables that will minimize the squared difference between the desired gear ratio, i_{trg}, and the current gear ratio, i_{tot}. For this problem, each variable is subject only to upper and lower boundary constraints.

The gear train problem was solved using the SOMA (see Table 26.4), i.e. by AllToOne and All ToAll strategies. The integer techniques described in Chapter 7 were invoked to handle boundary constraints. Because no constraint functions were involved, the objective (cost) function was simply defined to be the squared error, i.e., *f(X)*:

$$f_{cost}(X) = f(X) \qquad (26.3)$$

Table 26.3 lists the various gear train solutions and compares DE's result with those reported in [1], [2], [4], [5], [9], [8] and [11]. In Table 26.4 are the results of the SOMA algorithm for mutual comparison.

Table 26.3. Optimal solutions for the gear train problem.

Item	Optimum solution				Type
	Sandgren [1]	Fu et al. [2]	Loh and Papalambros [4]	Zhang and Wang [5]	
x_1 (z_d)	18	14	19	30	integer
x_2 (z_b)	22	29	16	15	integer
x_3 (z_a)	45	47	42	52	integer
x_4 (z_f)	60	59	50	60	integer
f(x)	5.7×10^{-6}	4.5×10^{-6}	0.233×10^{-6}	2.36×10^{-9}	
Gear Ratio	0.146666	0.146411	0.144762	0.144231	
Item	Optimum solution				Type
	Lin et al. [9]	Wu and Chow [8]	Cao & Wu [11]	Lampinen & Zelinka [12] *	
x_1 (z_d)	19	19	30	16	integer
x_2 (z_b)	16	16	15	19	integer
x_3 (z_a)	49	43	52	43	integer
x_4 (z_f)	43	49	60	49	integer
f(x)	$\underline{2.7\times10^{-12}}$	$\underline{2.7\times10^{-12}}$	2.36×10^{-9}	2.7×10^{-12}	
Gear Ratio	0.144281	0.144281	0.144231	0.144281	

* Also alternative solutions with an equal target function value were obtained from run to run.

Table 26.4. Optimal solutions for the gear train problem by SOMA.

Item	Optimum solution by SOMA*				Type
	AllToOne PathLength=3, Step=0.3, PopSize =100, PRT=0.61, Migrations=20, MinDiv = negative Average cost value = 2.05×10^{-10}		AllToAll PathLength=3, Step=0.3, Pop-Size=100, PRT=0.61, Migra-tions=5, MinDiv = negative Average cost value = 2.7×10^{-12}		
	the worst case	the best case	the worst case	the best case	
x_1 (z_d)	24	19	16	19	integer
x_2 (z_b)	13	16	19	16	integer
x_3 (z_a)	47	43	49	43	integer
x_4 (z_f)	46	49	43	49	integer
f(x)	$9.9\text{x}10^{-10}$	2.7×10^{-12}	2.7×10^{-12}	2.7×10^{-12}	
Gear Ratio	0.14311	0.144281	0.144231	0.144281	

* Also alternative solutions with an equal target function value were obtained from run to run as in the case of DE, see [12].

Solutions obtained by SOMA in Table 26.6 were obtained after 100 simulations for each SOMA version. From Table 26.4 it is visible that the solution found by SOMA was equally as good as the best solution in the literature or in the DE solutions. In addition, it should be noted that SOMA as well as DE provided different results from run to run with the same objective function value (Table 26.5).

Table 26.5. Alternative solutions for the gear train problem found by SOMA.

Alternative solutions for gear train problem by SOMA					
Version	Solution	z_d	z_b	z_a	z_f
AllToOne	1	24	13	47	46
	2	13	24	46	47
AllToAll	1	19	16	43	49
	2	16	19	43	49

Trough close inspection Eq. (26.2), it is obvious that there are four global optima. Because SOMA as well as any evolutionary algorithm in general can work with a population of solutions rather than just a single solution, it is capable of finding multiple global optima for this problem. By using a sufficiently large population, it is possible to obtain all four alternative solutions in a single run. Despite that, only one solution was extracted from the population of the last generation because the other three solutions can be found in a trivial way based on one solution and the symmetry of Eq. (26.2). In practice, however, there exist optimization tasks with multiple global optima that cannot be detected so simply. One may wonder why finding a single, globally optimal solution is not always sufficient when multiple global optima exist. One reason is that the sensitivity of the objective function to small changes in its variables may be different at the alternative optimal points. In practice, it is often important to select the most robust solution, i.e., the global optima with the least sensitivity to noise. For example, if a machine design is subject to optimization, it is possible that the optimized design variables cannot effectively be manufactured. Alternatively it may be that the design parameters change during the lifetime of the machine due, for example, to normal wear of its components. In such cases, robust global optima are to be preferred over those that exhibit a high sensitivity to design implementation errors.

Results for SOMA were repeated 100 times (see Fig. 26.2) and are based on parameters of the SOMA algorithm (Table 26.4). All runs yielded the reported value of *f(X)*. As mentioned, different solutions were obtained from run to run because of the existence of multiple global optima (Table 26.5).

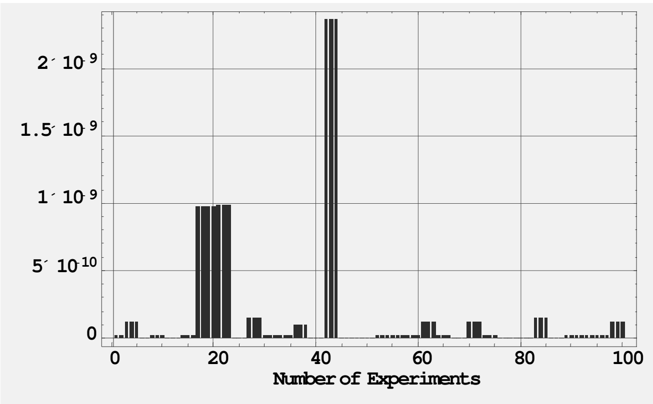

Fig. 26.3. Typical result of repeated optimization by SOMA (AllToOne)

26.1.2 Designing a pressure vessel

The second test problem is to design a compressed air storage tank with a working pressure of 3,000 psi and a minimum volume of 750 ft^3. As Fig. 26.3 shows, the cylindrical pressure vessel is capped at both ends by hemispherical heads. Using rolled steel plate, the shell is to be made in two halves that are joined by two longitudinal welds to form a cylinder. Each head is forged and then welded to the shell.

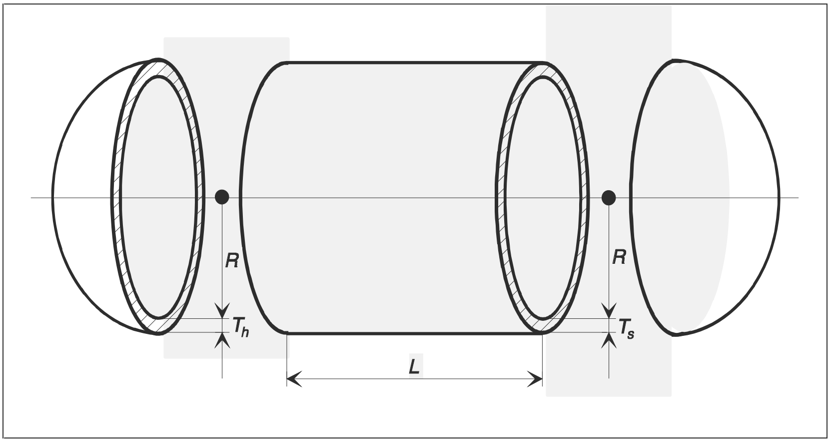

Fig. 26.3. Pressure vessel.

The objective is to minimize the manufacturing cost of the pressure vessel. The cost is a combination of the material cost, welding cost and forming cost. Refer to [1] for more details.

The design variables are shown in Fig 26.3. Variables L and R are continuous while T_s and T_h are discrete. The thickness of the shell, T_s, and the head, T_h, are both required to be standard sizes. For this example, steel plates were available in multiples of 0.0625 inch.

The problem can be formulated as follows:

$$\begin{aligned}
&\text{Find}\\
&X=(x_1,x_2,x_3,x_4)=(T_s,T_h,R,L)\\
&\text{to minimize}\\
&f(X)=0.6224x_1x_3x_4+1.7781x_2x_3^2+3.1611x_1^2x_4+19.84x_1^2x_3\\
&\text{subject to}\\
&g_1(X)=0.0193x_3-x_1\le 0\\
&g_2(X)=0.00954x_3-x_2\le 0\\
&g_3(X)=750.0\times 1728.0-\pi x_3^2x_4-\frac{4}{3}\pi x_3^3\le 0\\
&g_4(X)=x_4-240.0\le 0\\
&g_5(X)=1.1-x_1\le 0\\
&g_6(X)=0.6-x_2\le 0
\end{aligned} \tag{26.4}$$

The objective function, $f(X)$, represents the total manufacturing cost of the pressure vessel as a function of the design variables. The constraints, $g_1,\ldots,g_6$, quantify the restrictions to which the pressure vessel design must adhere. These limits arise from a variety of sources. For example, the minimal wall thickness of the shell, T_s (g_1), and heads, T_h (g_2), with respect to the shell radius are limited by the pressure vessel design code. The volume of the vessel must be at least the specified 750 ft^3 (g_3). Available rolling equipment limits the length of the shell, L, to no more than 20 feet (g_4). According to the pressure vessel design code, the thickness of the shell, T_s, is not to be less than 1.1 inches (g_5) and the thickness of the head, T_h, is not to be less than 0.6 inches (g_6).

The SOMA control variables used to solve the pressure vessel design problem are described in Table 26.8 (Case A), Table 26.10 (Case B) and Table 26.12 (Case C). The problem statements do not define the boundaries for the design variables, but the constraints, g_4, g_5 and g_6 are pure boundary constraints, so they were handled as lower boundary constraints for x_1 and x_2, and as an upper boundary constraint for x_4, respectively. The lower boundaries for x_3 and x_4 can be set to zero, since common sense demands that they must be non-negative values. The upper boundaries for x_1, x_2 and $x3$, however, must still be specified in order to define the search space. Consequently, these bounds were arbitrarily set high enough to make it highly probably that the global optimum lies inside of the defined search space. Since the possibility existed that the global optimum was outside of the initially defined search space, these estimated bounds were used only for initializing the population. SOMA was then allowed to extend the search beyond these boundaries. The possibility of using this kind of "loose" boundary constraint for variables are one of the advantages of modern evolutionary algorithms such as is for example DE. In practical engineering design work, it is not unusual for one or more boundaries to be unknown so that the distance to the optimum cannot be reliably estimated. The boundary constraints used for each variable are shown in Table 5. The other constraints, g_1, g_2 and g_3 were handled as constraint functions.

Table 26.6. Boundary constraints used for the pressure vessel example.

Boundary constraints for pressure vessel example		
Lower limitation	Constraint	Upper limitation
Constraint g_5	$1.1 \le x_1 \le 12.5$	Roughly guessed *
Constraint g_6	$0.6 \le x_2 \le 12.5$	Roughly guessed *
non-negative value of x_3	$0.0 \le x_3 \le 240.0$	Roughly guessed *
non-negative value of x_4	$0.0 \le x_4 \le 240.0$	Constraint g_4

* The value of this boundary is not given among the problem statements. Thus, the value is estimated roughly and used only for initialization of population. SOMA was allowed to extend the search beyond this limit.

The cost function for optimization was formulated as follows:

$$f_{\text{cost}}(X) = f(X) \cdot \prod_{i=1}^{3} c_i^2 \tag{26.5}$$

where

$$c_i = \begin{cases} 1.0 + s_i \cdot g_i(X) & \text{if} \quad g_i(X) > 0 \\ 1 & \text{otherwise} \end{cases}$$

$$s_1 = 1.0 \cdot 10^{10}, \ s_2 = s_3 = 1.0$$

Notice that it is not necessary to evaluate the constraint functions, g_4, g_5 and g_6, because they were handled as boundary constraints, and thus there was no reason to generate a candidate vector that violated any of them.

In researching this problem, at least three different formulations were found in the literature. To enable a more comprehensive comparison with the other methods, all three cases were solved using SOMA. Case A is an exception to the problem statements above, since all of the variables are treated as continuous. Table 26.8 contains results for SOMA (Case A) and can be compared with Table 26.7. Case B is formulated according to the original problem statements and results are in Table 26.10. Case C, reported in Table 26.12, also differs from the original problem statements. For some unknown reason, [7], [10] and [11] have used a slightly reformulated constraint-function, g_5:

$$g_5(X) = 1.0 - x_1 \le 0 \tag{26.6}$$

This modification extends the region of feasible solutions and also makes it possible to obtain a significantly lower objective function value. All of the solutions of [7], [10] and [11] lie in the extended region of the search space. Because of that, the results of [7], [10] and [11] cannot be fairly compared with the results obtained using Sandgren's original problem statements [1]. Thus, Cases A, B and C are not comparable because they represent differently formulated problems.

As Table 26.8, Table 26.10 and Table 26.12, show, SOMA, usually similar to DE, found a better solution for each pressure vessel design problem than the best solution found in the literature. Computations were based on parameters as mentioned in these tables and were repeated 100 times. In addition, all solutions were within the feasible design domain

Table 26.7. Optimal solutions for the pressure vessel problem. Case A: all variables are treated as continuous.

Item	Optimum solution, Case A			Type
	Sandgren [1]	Fu et al. [2]	Lampinen & Zelinka [12]	
x_1 (T_s) [inch]	1.1	1.100001	1.100000	continuous
x_2 (T_h) [inch]	0.6	0.600016	0.600000	continuous
x_3 (R) [inch]	47.7008	48.35145	56.99482	continuous
x_4 (L) [inch]	117.701	111.9893	51.00125	continuous
f(X) [$]	7867.0	7790.588	<u>7019.031</u>	

Table 26.8. Optimal solutions for the pressure vessel problem by SOMA. Case A: all variables are treated as continuous.

Item	Optimum solution by SOMA – Case A				Type
	AllToOne PathLength=3, Step=0.11, PopSize=20, PRT=0.1, Migrations=100, MinDiv = negative Average cost value = 7057.77		AllToAll PathLength=3, Step=0.11, PopSize=20, PRT=0.9, Migrations=20, MinDiv = negative Average cost value = 7030.4		
	the worst case	the best case	the worst case	the best case	
x_1 (T_s)	1.10019	1.10015	1.1791	1.1	cont.
x_2 (T_h)	0.600052	0.600001	0.6	0.6	cont.
x_3 (R)	54.5505	57.0024	61.0933	56.9948	cont.
x_4 (L)	65.923	50.9591	29.069	51.0012	cont.
f(x)	7199.75	7019.32	7098.15	7019.03	

Table 26.9. Optimal solutions for the pressure vessel problem. Case B: solved according to Sandgren's original problem statements [1].

Item	Optimum solution, Case B *					Type
	Sandgren [1]	Fu et al. [2]	Loh and Papalambros [4]	Wu and Chow [8]	Lampinen & Zelinka [12]	
X_1 (T_s)	1.125	1.125	1.125	1.125	1.125	discrete
X_2 (T_h)	0.625	0.625	0.625	0.625	0.625	discrete
X_3 (R)	48.97	48.38070	58.2901 **	58.1978	58.29016	continuous
X_4 (L)	106.72	111.7449	43.693	44.2930	43.69266	continuous
f(x)	7982.5	8048.619	7197.734 **	7207.497	<u>7197.729</u>	

* In [5] and [9] it is reported that the value of f(X) = 7197.7 was reached. Because neither a more accurate result, nor details of the result were provided, they are not included in this comparison.
** No values for constraint functions were reported in [4]. Also, the optimum solution was too inaccurately reported to reconstruct the results properly. The reported value, x_3 = 58.290, causes a violation of constraint g_3. Because of that, 58.2901 was used here for reconstructing the constraint functions and target function values. In [4], a value of f(X) = 7197.7 was originally reported.

Table 26.10. Optimal solutions for the pressure vessel problem. Case B: solved according to Sandgren's original problem statements [1].

Item	Optimum solution by SOMA – Case B				Type
	AllToOne PathLength=3, Step=0.11, PopSize=20, PRT=0.5, Migrations=20, MinDiv = negative Average cost value = 7256.71		AllToAll PathLength=3, Step=0.11, PopSize =20, PRT=0.5, Migrations=20, MinDiv = negative Average cost value = 7197.73		
	the worst case	the best case	the worst case	the best case	
x_1 (T_s)	1.125	1.125	1.125	1.125	discr.
x_2 (T_h)	0.625	0.625	0.625	0.625	discr.
x_3 (R)	55.8592	55.8592	58.2902	58.2902	cont.
x_4 (L)	57.7315	57.7315	43.6927	43.6927	cont.
f(x)	7359.2	7197.73	7197.73	7197.73	

Table 26.11. Optimal solutions for the pressure vessel problem. Case C: different formulation of constraint-function, g_5, with respect to Sandgren's original problem statements [1].

Item	Optimum solution, Case C				Type
	Li and Chou [7]	Cao & Wu [11]	Thierauf and Cai [10]	Lampinen & Zelinka [12]	
x_1 (T_s)	1.000	1.000	1.000	1.000	discrete
x_2 (T_h)	0.625	0.625	0.625	0.625	discrete
x_3 (R)	51.250	51.1958	51.812	51.81347	continuous
x_4 (L)	90.991	90.7821	84.591	84.57853	Continuous
f(x)	7127.3	7108.6160	7006.9	7006.358	

Table 26.12. Optimal solutions for the pressure vessel problem. Case C: different formulation of constraint-function, g_5, with respect to Sandgren's original problem statements [1].

Item	Optimum solution by SOMA – Case C				Type
	AllToOne PathLength=3, Step=0.11, PopSize =20, PRT=0.5, Migrations=20, MinDiv = negative Average cost value = 7050.45		AllToAll PathLength=3, Step=0.11, PopSize =20, PRT=0.5, Migrations=20, MinDiv = negative Average cost value = 7006.36		
	the worst case	the best case	the worst case	the best case	
x_1 (T_s)	1.000	1.000	1.000	1.000	discr.
x_2 (T_h)	0.625	0.625	0.625	0.625	discr.
x_3 (R)	51.8128	51.8128	51.8135	51.8135	cont.
x_4 (L)	84.5839	84.5839	84.5785	84.5785	cont.
f(x)	7106.53	7006.42	7006.36	7006.36	

26.1.3 Designing a coil compression spring

The third example involves the design of a coil compression spring (Fig. 26.4). The spring is to be a helical compression spring to which a strictly axial and constant load will be applied.

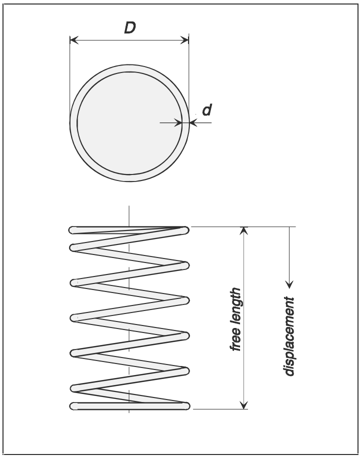

Fig. 26.4. Coil spring for Example 3.

The objective is to minimize the volume of spring steel wire needed to manufacture the spring. The design variables are the number of spring coils, N, the outside diameter of the spring, D, and the spring wire diameter, d. This example contains integer, discrete and continuous variables. The number of spring coils, N, is an integer variable and the outside diameter, D, is a continuous variable. Additionally, the spring wire diameter, d, is only available in the standard (discrete) sizes shown in Table 26.13.

Table 26.13. Allowable spring steel wire diameters for the coil spring design problem.

Allowable wire diameters [inch]					
0.009	0.0095	0.0104	0.0118	0.0128	0.0132
0.014	0.015	0.0162	0.0173	0.018	0.020
0.023	0.025	0.028	0.032	0.035	0.041
0.047	0.054	0.063	0.072	0.080	0.092
0.105	0.120	0.135	0.148	0.162	0.177
0.192	0.207	0.225	0.244	0.263	0.283
0.307	0.331	0.362	0.394	0.4375	0.500

The problem is formulated as follows:

$$
\begin{aligned}
&\text{Find} \\
&X = (x_1, x_2, x_3) = (N, D, d) \\
&\text{to minimize} \\
&f(X) = \frac{\pi^2 x_2 x_3^2 (x_1 + 2)}{4} \\
&\text{subject to} \\
&g_1(X) = \frac{8 C_f F_{\max} x_2}{\pi x_3^3} - S \le 0 \\
&g_2(X) = l_f - l_{\max} \le 0 \\
&g_3(X) = d_{\min} - x_3 \le 0 \\
&g_4(X) = x_2 - D_{\max} \le 0 \\
&g_5(X) = 3.0 - \frac{x_2}{x_3} \le 0 \\
&g_6(X) = \sigma_p - \sigma_{pm} \le 0 \\
&g_7(X) = \sigma_p + \frac{F_{\max} - F_p}{K} + 1.05(x_1 + 2)x_3 - l_f \le 0 \\
&g_8(X) = \sigma_w - \frac{F_{\max} - F_p}{K} \le 0 \\
&\text{where} \\
&C_f = \frac{4(x_2 / x_3) - 1}{4(x_2 / x_3) - 4} + \frac{0.615 x_3}{x_2} \le 0 \\
&K = \frac{G x_3^4}{8 x_1 x_2^3} \\
&\sigma_p = \frac{F_p}{K} \\
&l_f = \frac{F_{\max}}{K} + 1.05(x_1 + 2)x_3
\end{aligned}
\tag{26.7}
$$

The objective function, *f(X)*, computes the volume of spring steel wire as a function of the design variables. The design constraints are specified as follows:

a) The maximum working load is: F_{max} = 1000.0 lb.
b) The allowable maximum shear stress is: S = 189000.0 psi (g_1).
c) The maximum free length is: l_{max} = 14.0 inch (g_2).
d) The minimum wire diameter is: d_{min} = 0.2 inch (g_3).
e) The maximum outside diameter of the spring is: D_{max} = 3.0 inch (g_4).
f) The pre-load compression force is: F_p = 300.0 lb.
g) The allowable maximum deflection under pre-load is: σ_{pm} = 6.0 inch (g_6).
h) The deflection from pre-load position to maximum load position is: σ_w = 1.25 inch (g_8).
i) The combined deflections must be consistent with the length, i.e., the spring coils should not touch each other under the maximum load at which the maximum spring deflection occurs (g_7).
j) The shear modulus of the material is: $G = 11.5\times10^6$.
k) The spring is guided, so the buckling constraint is bypassed.
l) The outside diameter of the spring, D, should be at least three times greater than the wire diameter, d, to avoid lightly wound coils (g_5).

A more detailed explanation about the coil spring design procedure can be found in [1], [8] and in [13]. The SOMA control variable settings that solved the coil spring problem are described in Table 26.16 (Case A) and Table 26.18 (Case B). Although the problem statements do not define the boundaries for design variables, the constraints, g_3 and g_4 are pure boundary constraints and were treated as a lower boundary constraint for x_3 and as an upper boundary constraint for x_2, respectively. Furthermore, $g5$ can also be handled as a boundary constraint. In order to define the search space, the other boundary constraints were chosen based on the problem statements and the simple geometric space limitations elaborated in Table 26.14. The remaining constraints were handled as soft-constraint functions.

Table 26.14. Boundary constraints used for the coil spring example.

Boundary constraints for coil spring example		
Lower limitation	Constraint	Upper limitation
At least one spring coil is required to form a spring.	$1 \le x_1 \le \dfrac{l_{\max}}{d_{\min}}$	Upper and lower surfaces of unloaded spring coils touch each other.
Constraints g_3 and g_5 together	$3d_{\min} \le x_2 \le D_{\max}$	constraint g_4
Constraint g_3	$d_{\min} \le x_3 \le \dfrac{D_{\max}}{3}$	Constraints g_4 and g_5 together

The cost function for optimization was formulated as follows:

$$f_{cost}(X) = f(X) \cdot \prod_{i=1}^{2} c_i^3 \cdot \prod_{i=6}^{8} c_i^3 \tag{26.8}$$

where

$$c_i = \begin{cases} 1.0 + s_i \cdot g_i(X) & \text{if} \quad g_i(X) > 0 \\ 1 & \text{otherwise} \end{cases}$$

$$s_1 = s_2 = s_6 = 1.0 \,,\; s_7 = s_8 = 1.0 \cdot 10^{10}$$

Constraint functions: g_3, g_4 and g_5 did not have to be evaluated because they were handled as boundary constraints and SOMA was not allowed to generate a candidate vector that violated any of them.

Table 26.15. Optimal solutions for the coil spring problem. Case A: continuous solution.

Item	Optimum solution, Case A		Type
	Sandgren [1]	Lampinen & Zelinka [12]	
x_1 (N)	9.1918	7.117621	Continuous
x_2 (D)	1.2052	1.372407	Continuous
x_3 (d)	0.2814	0.2909652	Continuous
f(x)	2.6353	<u>2.61388</u>	

Table 26.16. Optimal solutions for the coil spring problem by SOMA. Case A: continuous solution.

Item	Optimum solution by SOMA – Case A				Type
	AllToOne PathLength=3, Step=0.14, PopSize =40, PRT=0.3, Migrations=30, MinDiv = negative Average cost value = 2.65614		AllToAll PathLength=3, Step=1.5, PopSize =40, PRT=0.1, Migrations=30, MinDiv = negative Average cost value = 2.61389		
	the worst case	the best case	the worst case	the best case	
x_1 (N)	12.6744	7.11045	7.08773	7.11854	cont.
x_2 (D)	1.02875	1.37328	1.37538	1.37232	cont.
x_3 (d)	0.270741	0.291027	0.291131	0.29096	cont.
f(X)	2.73034	2.6146	2.61393	2.61388	

Table 26.17. Optimal solutions for the coil spring problem. Case B: discrete solution.

Item	Optimum solution, Case B				Type
	Sandgren [1]	Chen and Tsao [6]	Wu and Chow [8]	Lampinen & Zelinka [12]	
X_1 (N)	10	9	9	9	integer
X_2 (D)	1.180701	1.2287	1.227411	1.2230410	continuous
x_3 (d)	0.283	0.283	0.283	0.283	discrete
f(X)	2.7995	2.6709	2.6681	<u>2.65856</u>	

Table 26.18. Optimal solutions for the coil spring problem by SOMA. Case B: discrete solution.

Item	Optimum solution by SOMA – Case B				Type
	AllToOne PathLength=3, Step=0.14, PopSize =40, PRT=0.5, Migrations=30, MinDiv = negative Average cost value = 2.71043		AllToAll PathLength=3, Step=1.5, PopSize =40, PRT=0.5, Migrations=30, MinDiv = negative Average cost value = 2.66346		
	the worst case	the best case	the worst case	the best case	
x_1 (N)	3	9	9	9	integer
x_2 (D)	2.17372	1.22304	1.22305	1.22304	cont.
x_3 (d)	0.331	0.283	0.283	0.283	discr.
f(X)	2.93811	2.65855	2.67599	2.65859	

Table 26.15 and Table 26.17 compare SOMA's solution with results obtained by other researchers. Table 26.16 and Table 26.18 shows results from the SOMA algorithm. Two different versions of the coil spring problem were solved. Case A (Table 26.15 and Table 26.16) reports the solutions generated when all variables are assumed to be continuous. Case B (Table 26.17 and Table 26.18) reports the mixed-variable solution according to the problem statements above. In both cases, SOMA generally obtained a better solution than the best solution found in the literature (except and comparable mostly with DE). In order to demonstrate the robustness of the SOMA algorithm, 100 independent optimization trials were performed for both Case A and Case B. All trials yielded the reported value of *f(X)* or better. All of the solutions were also within the feasible region of the design space.

26.2 Conclusion

This chapter describes the use of SOMA in mechanical engineering problems: it is a new algorithm for global optimization. The basic principles of this algorithm were introduced in Chapter 7, including versions of the algorithm, an overview of selected applications, as well as testing for robustness and the handling of various constraints. The methods described for handling constraints are relatively simple, easy to implement and easy to use. SOMA is capable of optimizing integer, discrete and continuous variables and capable of handling non-linear objective functions with multiple non-trivial constraints as well as differential evolution or the other evolutionary algorithms which use techniques described in Chapter 7.

A soft-constraint (penalty) approach is applied for the handling of constraint functions. Some optimization methods require a feasible initial solution as a starting point for a search. Preferably, this solution should be rather close to a global optimum to ensure convergence to it instead of to a local optimum. If non-trivial constraints are imposed, it may be difficult or impossible to provide a feasible initial solution.

The efficiency, effectiveness and robustness of many methods are often highly dependent on the quality of the starting point. The combination of the SOMA algorithm and the soft-constraint approach does not require any initial solution, but yet it can take advantage of a high quality initial solution if one is available.

For example, this initial solution can be used for initialization of the population in order to establish an initial population that is biased towards a feasible region of the search space. If there are no feasible solutions in the search space, as is the case for totally conflicting constraints, SOMA algorithms with the soft-constraint approach are still able to find the nearest feasible solution. This is often important in practical engineering optimization applications, because many non-trivial constraints are involved.

After using SOMA to optimize the well-known test functions (Chapter 7), it was then used on three optimization problems from the field of mechanical engineering. This set of problems is useful for testing because **11** other algorithms (Table 26.2) from the EA domain were already used there. Hence, it offers quite a rich source of information for comparing new algorithms. All these problems comprised variables from the continual, integer and discrete domain for the cost function arguments. All tests have shown that results from SOMA are fully comparable with other algorithms especially with DE, which was proven here to be the best of all methods used. For all problems, 100 simulations were carried out for two versions of SOMA (AllToOne and AllToAll). All these results are described discussed in this chapter.

The SOMA algorithm is undoubtedly one of the most promising and novel methods for non-linear optimization that can be applied generally, and they work with minimum assumptions with respect to the objective function.

The algorithm requires only the cost value returned from the objective function for guidance of its seeking for the optimum. No derivatives or other auxiliary information are needed. Including the algorithm's extensions discussed in this article, the SOMA algorithm can be applied to a wide range of optimization problems, which practitioners in the field of modern optimization would like to solve.

All results described here can be very easy checked and repeated by means of C source code and Mathematica software - all is accessible at [14]. All examples are predefined there and thus it is easy to use them.

Acknowledgements

This work was supported by the grant No. MSM 26500014 of the Ministry of Education of the Czech republic and by grants of Grant Agency of Czech Republic GACR 102/03/0070 and GACR 102/02/0204.

References

[1] Sandgren, E. (1990). *Nonlinear integer and discrete programming in mechanical design optimization.* Transactions of the ASME, Journal of Mechanical Design, 112(2):223–229, June 1990. ISSN 0738-0666

[2] Fu, J.-F., Fenton, R. G. and Cleghorn, W. L. (1991). *A mixed integer-discrete-continuous programming method and its application to engineering design optimization.* Engineering Optimization, 17(4):263–280, 1991. ISSN 0305-2154

[3] Loh, Han Tong and Papalambros, P. Y. (1991). *A sequential linearization approach for solving mixed-discrete nonlinear design optimization problems*. Transactions of the ASME, Journal of Mechanical Design, 113(3):325–334, September 1991.

[4] Loh, Han Tong and Papalambros, P. Y. (1991). *Computational implementation and tests of a sequential linearization algorithm for mixed-discrete nonlinear design optimization.* Transactions of the ASME, Journal of Mechanical Design, 113(3):335–345, September 1991.

[5] Zhang, Chun and Wang, Hsu-Pin (1993). *Mixed-discrete nonlinear optimization with simulated annealing.* Engineering Optimization, 21(4):277–291, 1993. ISSN 0305-215X

[6] Chen, J. L. and Tsao, Y. C. (1993). *Optimal design of machine elements using genetic algorithms.* Journal of the Chinese Society of Mechanical Engineers, 14(2):193–199, 1993.

[7] Li, H.-L. and Chou, C.-T. (1994). *A global approach for nonlinear mixed discrete programming in design optimization.* Engineering Optimization, 22():109–122, 1994.

[8] Wu, S.-J. and Chow, P.-T. (1995). *Genetic algorithms for nonlinear mixed discrete-integer optimization problems via meta-genetic parameter optimization.* Engineering Optimization, 24(2):137–159, 1995. ISSN 0305-215X

[9] Lin, Shui-Shun, Zhang, Chun and Wang, Hsu-Pin (1993). *On mixed-discrete nonlinear optimization problems: A comparative study.* Engineering Optimization, 23(4):287–300, 1995. ISSN 0305-215X

[10] Thierauf, G. and Cai, J. (1997*). Evolution strategies – parallelisation and application in engineering optimization.* In B.H.V. Topping (ed.) (1997). *Parallel and distributed processing for computational mechanics.* Saxe-Coburg Publications, Edinburgh (Scotland). ISBN 1-874672-03-2

[11] Cao, Y. J. and Wu, Q. H. (1997). *Mechanical design optimization by mixed-variable evolutionary programming*. Proceedings of the *1997 IEEE Confer-*

ence on Evolutionary Computation, IEEE Press, pp. 443–446.

[12] Lampinen Jouni, ZELINKA, Ivan. New Ideas in Optimization - Mechanical Engineering Design Optimization by Differential Evolution. Volume 1. London : McGraw-Hill, 1999. 20 p. ISBN 007-709506-5.

[13] Siddall, James N. (1982). *Optimal engineering design: principles and applications.* Mechanical engineering series / 14. Marcel Dekker Inc. ISBN 0-8247-1633-7

[14] Articles about SOMA algorithm (source codes, graphics animated gallery, bibliography, see http://www.ft.utb.cz/people/zelinka/soma