Eng. Opt., 2002, Vol. 34, pp. 341–354

A SOCIO-BEHAVIOURAL SIMULATION MODEL FOR ENGINEERING DESIGN OPTIMIZATION

SHAMIM AKHTAR*, KANG TAI† and TAPABRATA RAY‡

Centre for Advanced Numerical Engineering Simulations, School of Mechanical and Production Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798

(Received 29 June 2001; In final form 4 January 2002)

This paper proposes a method for solving single objective constrained optimization problems by way of a socio-behavioural simulation model. The essence of the methodology is derived from the concept that the behaviour of an individual changes and improves due to social interaction with the society leaders. Leaders are identified after all individuals of a society are Pareto ranked according to constraint satisfaction. At the higher end, leaders of all societies interact among themselves for the overall improvement of the societies. Such overall improvement of individual societies leads to a better civilization. Four well-studied single objective constrained optimization problems have been solved to show the efficacy of the proposed methodology.

Keywords: Socio-behavioural model; Constrained optimization; Constraint handling; Pareto ranking

1 INTRODUCTION

Recently much attention has been paid to the treatment of constrained optimization problems using evolutionary computation techniques in order to solve large-scale real world problems. Several methods have been proposed to take care of constraints in evolutionary computational models [1–4]. A substantial part of the research on evolutionary computations, however, focuses on the processes of natural selection and genetics [1, 5]. Computational methods based on socio-behavioural models are fairly new developments. Reynolds [6] introduced a cultural algorithm in which problem solving experience of individuals in a population space was collected and reasoned about within a space that he called the belief space. This cultural algorithm framework is based on a dual inheritance system with evolution taking place at the micro-evolutionary (population) level and at the macro-evolutionary (belief space) level. At the micro-evolutionary level, there is a population of individuals in which each individual is described by a set of behavioural traits. These traits are passed from generation to generation through socially motivated operators. At the macro-evolutionary

* Present address: Realistix Laboratories Pte Ltd, 73 Science Park Drive, CINTECH I #B1-18, Singapore Science Park, Singapore 118254. E-mail: Shamima@realistix.com

† Corresponding author. E-mail: mktai@ntu.edu.sg

‡ Present address: Temasek Laboratories, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260. E-mail: tsltray@nus.edu.sg

ISSN 0305-213X print; ISSN 1029-0273 online
DOI: 10.1080/0305215021000001677

level, individuals' experiences are collected, merged, generalized and specialized in the belief space [6]. The particle swarm optimization method is another highly innovative adaptive algorithm that is based on a social–psychological metaphor. In a particle swarm, a population of individuals adapts by moving stochastically towards previously successful regions in the search space and are influenced by the successes of their topological neighbours [7]. However, none of these algorithms demonstrate grouping/clustering of individuals in the parametric space which is essential to socio-behavioural dynamics. In the present work, an algorithm is suggested that is based on the socio-behavioural concept of society and civilization.

From time immemorial individuals of a society have interacted with one another. Such cooperative relations result in a rich amalgamation of information thereby improving the societies they live in. With these societies being subsets of the whole civilization, the civilization improves as well. In the present context, a society is considered as a set of mutually interacting individuals who are led by a group of leaders. These individuals are the fundamental social entities and they interact with the leaders in their society in a quest to improve. This improvement in an individual's performance is due to meaningful information acquisition from a better performing individual (leader) belonging to the same society. Such intra-society interactions between every individual and its leader results in an improvement of every individual's performance, which in turn leads to the emergence of more advanced (better performing) societies over time. However, intra-society interactions would not improve the performance of the leaders. These leaders, unlike other average individuals of the society, therefore communicate and collaborate externally with the leaders of other societies of the civilization. Such inter-society information exchange among leaders results in the migration of leaders toward developed societies led by better performing leaders.

2 SOCIO-BEHAVIOURAL MODEL

At any given time instant t, a civilization is defined as a set of m individuals:

$$\begin{aligned} &\mathrm{CIV} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\} \\ &\text{where } \mathbf{x}_i = [x_1 \quad x_2 \quad \cdots \quad x_n], \quad 1 \le i \le m \end{aligned} \tag{1}$$

m is the size of the whole civilization and n is the number of design variables defining an individual. These m individuals are parametrically clustered into p number of mutually exclusive clusters where each cluster corresponds to a society. The individuals that dominate other individuals within a society are then identified. They thus become the leaders of the society. Other individuals which are non-leaders share information with the leaders and move towards better positions. This movement is governed by the equation:

$$\mathbf{x}' = \xi(G_{\min}(\mathbf{x}), \mathbf{x}) \tag{2}$$

where $\mathbf{x}'$ is the new position of $\mathbf{x}$, $G_{\min}(\mathbf{x})$ is a function that returns the nearest leader to $\mathbf{x}$ in its own society and ξ is the information acquisition between a leader and a non-leader. This equation ensures an intensified local search within a society. Leaders of all societies are then grouped together to form a global society of leaders. The global leaders' society, once formed, behaves like any other society. Hence, here too, a few individuals that dominate other individuals become leaders of the society. Since this society is the leaders' society, the leaders of this society can be referred to as super leaders. All other individuals (within this society) which are not super leaders, acquire information from the super leaders

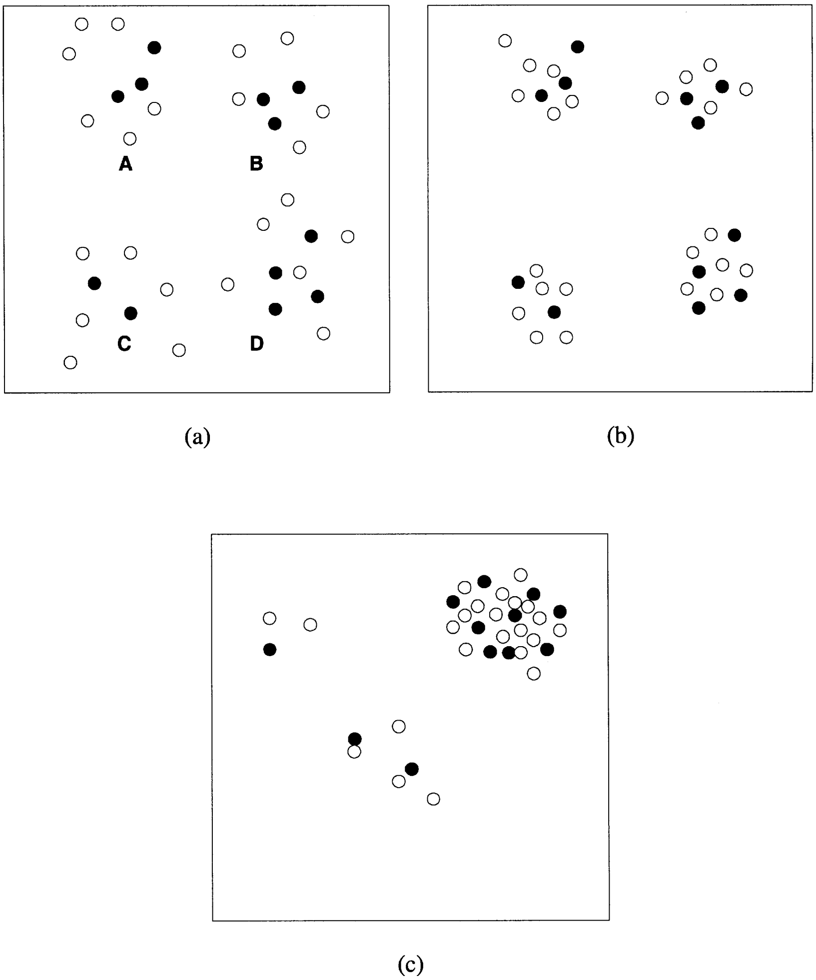

FIGURE 1 (a) Clusters A, B, C and D, where leaders are marked as ● while other individuals are marked as ○. (b) Individuals in every cluster relocate towards their leaders. (c) Cluster formation after a sufficient period of time.

and move using the same Eq. (1) described previously. With this equation applied to the leaders' society, a search for the global optimum is achieved.

A visual representation of the socio-behavioural model is shown in Figure 1. Figure 1(a) schematically depicts four societies A, B, C and D forming a civilization at an initial time instant. The individuals are clustered to form the societies based on their positions in the parametric space. Figure 1(b) shows the relocation of individuals towards their leaders and Figure 1(c) shows the formation of clusters after a sufficient period of time. It is seen that most of the individuals have gathered within the area occupied initially by cluster B (the optimum one in this case), while the other (initial) clusters become depleted.

3 ALGORITHM

The socio-behavioural algorithm developed here is based on the socio-behavioural model described in the previous section. An initial civilization is created through a random initialization of a number of individuals. At every time instant, the individuals are separated into a number of mutually exclusive clusters based on their positions in the parametric space. Such clusters represent the societies and the collection of all clusters represents the civilization. A multilevel Pareto ranking scheme based on constraint satisfaction is implemented to generate the set of leaders. The criterion for leader selection is thus emphasized to drive societies towards the feasible region. An individual in a society extracts information from its nearest leader through an intra-society interaction that translates into an intensified search around better performing points. The leaders of different societies compete among themselves to attract leaders from other societies. The migration of a leader from a society to another within the civilization is achieved through an information acquisition from a better performing leader. Over time, the above process reduces the number of individuals in non-feasible and non-optimum regions and enlarges feasible and optimum regions. This phenomenon promises search over global regions of the parametric space. The algorithm is as follows:

Step 1: Initialize m points that correspond to m individuals in the civilization at an initial time $t = 0$.
Step 2: Cluster the m points into p mutually exclusive clusters (each cluster represents a society).
Step 3: Identify a set of better performing individuals in each cluster as the leaders of that society.
Step 4: For every individual in a society that is not a leader, acquire information from its closest leader within the same society and move to a new position for the next time step.
Step 5: Collate the leaders of all the societies to form a global society of leaders for the entire civilization.
Step 6: Identify a set of better performing individuals for this global society of leaders to form super leaders for the civilization.
Step 7: For every leader in the global society that is not a super leader, acquire information from its closest super leader and move to a new position for the next time step.
Step 8: Super leaders do not change their position and as such go to the next time step.
Step 9: Increment time step by unity: $t = t + 1$.
Step 10: If $t = T$ (where T is the time frame for the advancement of the civilization), terminate the procedure. Else, return to Step 2.

3.1 Initialization

A civilization of m solutions is generated, where each solution vector $\mathbf{x}_i = [x_1 \ \ x_2 \ \ \cdots \ \ x_n]$ is generated for $1 \le i \le m$ as follows:

$$x_j = l_j + R(u_j - l_j), \quad 1 \le j \le n \tag{3}$$

where l_j and u_j are the lower and upper bounds, respectively, of the jth variable of each $\mathbf{x}_i$ and R is a random number between 0 and 1.

3.2 Clustering Algorithm

After initialization, the m individuals are clustered into p mutually exclusive clusters. While clustering, each individual is considered to be a point in the n-dimensional space, where n is the number of variables defining the individual. The distance between two points in such a space is essentially the Euclidean norm between the two points. The clustering is done using the following clustering algorithm:

Step 1: From the set of m points (individuals), randomly choose any one to be the first hub.
Step 2: Find the point that is farthest from this hub, and make it the second hub.
Step 3: For each remaining point, find the distance between the point and each hub and assign the point to the hub which is closer to it.
Step 4: Compute the average distance between hubs. This is done by adding up the distances between all possible pairings of hubs and dividing by the number of pairs. (For example, if there are four hubs, there will be 6 possible pairings.) In general the maximum possible number of pairs formed by k hubs is given by $k(k-1)/2$. The average distance is then divided by 2 and assigned to a variable D. Calculate the distance between each point and its hub. If none of these distances is greater than D, terminate the procedure. Else, if any of these distances is greater than D, continue to Step 5.
Step 5: Make the point that is farthest from any of the hubs to be a new hub.
Step 6: Calculate the distance between each point and the new hub. If this distance is less than the point's distance from the current hub, assign the point to the new hub.
Step 7: Return to Step 4.

3.3 Ranking Based on the Constraint Matrix

The process of Pareto ranking based on constraint satisfaction is described in the context of the following optimization problem statement:

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}) \geq 0, \quad \text{for } i = 1, 2, \ldots, q \qquad (4) \\ & h_j(\mathbf{x}) = 0, \quad \text{for } j = 1, 2, \ldots, r \qquad (5) \end{aligned}$$

The r equality constraints are transformed to $2r$ inequality constraints. This is done by converting each $h_j(\mathbf{x}) = 0$ to a pair of inequalities by introducing a small tolerence $\delta > 0$. These two inequalities are:

$$\begin{aligned} -h_j(\mathbf{x}) &\geq -\delta \\ h_j(\mathbf{x}) &\geq -\delta \end{aligned} \qquad (6)$$

Therefore, the total number of inequality constraints for the problem is given by $s = q + 2r$. The constraint satisfaction vector for each solution $\mathbf{x}$ is denoted by $\mathbf{c} = [c_1 \;\; c_2 \;\; \cdots \;\; c_s]$ where

$$c_i = \begin{cases} 0 & \text{if the } i\text{th constraint is satisfied,} \quad 1 \leq i \leq s \\ -g_i(\mathbf{x}) & \text{if the } i\text{th constraint is violated,} \quad 1 \leq i \leq q \\ -\delta - h_i(\mathbf{x}) & \text{if the } i\text{th constraint is violated,} \quad q+1 \leq i \leq q+r \\ -\delta + h_i(\mathbf{x}) & \text{if the } i\text{th constraint is violated,} \quad q+r+1 \leq i \leq s \end{cases} \qquad (7)$$

The **CONSTRAINT** matrix for a society (cluster) consisting of w individuals will therefore assume the form of

$$\mathbf{CONSTRAINT} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1s} \\ c_{21} & c_{22} & \cdots & c_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{w1} & c_{w2} & \cdots & c_{ws} \end{bmatrix} \quad (8)$$

In a society of w individuals, all non-dominated individuals based on the **CONSTRAINT** matrix are assigned a rank of 1. The rank 1 individuals are removed from the cluster and the new set of non-dominated individuals are assigned a rank of 2. The process is continued until every individual in the cluster is assigned a rank. Due to the way that the value of c is computed, it should be noted that for a society where there are some feasible individuals, all these feasible individuals are non-dominated and will be assigned a rank of 1 while all the infeasible individuals will be ranked 2,3,4,... and so on.

3.4 Leader Identification Algorithm

A leader in a society is determined using the leader identification algorithm. This algorithm is as follows: If the number of rank 1 individuals (ranked as described in the preceding section) does not exceed 50% of the society size, then choose all rank 1 individuals as leaders. If the number of rank 1 individuals exceed 50% of the society size, then calculate the average objective function value of the society. Choose as leaders those rank 1 individuals whose objective values are better than or equal to the average objective value of the society.

The resulting effect of this algorithm is such that, in the case where there are only a few feasible individuals in a society, only these feasible individuals will be selected as leaders. As the number of feasible individuals increases over time until there are many of them (more than 50% of society size), then only those who are better than average (in terms of objective function value) will become leaders. In the case where all individuals in a society are infeasible, then only the non-dominated ones (they are ranked 1 in terms of constraint satisfaction) are leaders. If there are too many of these non-dominated individuals (more than 50% of society size), then there is a need to be more selective and so only those with better than average objective values are chosen as leaders.

3.5 Information Acquisition Operator

The information acquisition is based on a simple operator. The operator can result in a variable value that does not already exist in either the individual or its leader, which is useful to avoid premature convergence. The probability of a variable value being generated that is in between that of the individual and its leader is 50%. There is a 25% probability of a variable value being generated between the lower bound of the variable and the individual or its leader (whichever is the lesser). There is also a 25% probability of a variable value being generated between the upper bound of the variable and the individual or its leader (whichever is the greater). This is explained using Figure 2. Consider the case where x_{Ai} is the ith variable of an individual A and x_{Li} the ith variable of a leader L, and $x_{Ai} < x_{Li}$. Then the probability that a value will be generated between x_{Ai} and x_{Li} is 50% while the probability is 25% each that the value generated is between l_i and x_{Ai} or between x_{Li} and u_i. Generating a value between l_i and x_{Ai} represents a move by the individual away from its leader. Within the socio-behavioural model, the 25% chance of this happening can be interpreted as allowing for some individuals who choose not to follow their leaders but to go against them

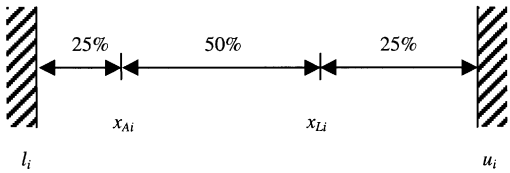

FIGURE 2 Schematic representation of the information acquisition operator.

instead. This generates some diversity in the society and thus helps to explore the parametric space for the global optimum.

Other operators that generate intermediate variable values like the blend crossover (BLX) as proposed by Eshelman and Shaffer [8] or the simulated binary crossover (SBX) as proposed by Deb and Agrawal [9] can also potentially be used in place of the above operator.

4 EXAMPLES

4.1 Two-Variable Constrained Optimization Problem

This is a two variable constrained optimization problem from Koziel and Michalewicz [3].

$$\begin{aligned}
\text{Minimize} \quad & f(\mathbf{x}) = (x_1 - 10)^3 + (x_2 - 20)^3 \\
\text{subject to} \quad & (x_1 - 5)^2 + (x_2 - 5)^2 - 100 \geq 0 \\
& -(x_1 - 6)^2 - (x_2 - 5)^2 + 82.81 \geq 0 \\
& 13 \leq x_1 \leq 100, \;\; 0 \leq x_2 \leq 100
\end{aligned}$$

It is subjected to two non-linear inequalities and the objective function is a cubic function. The ratio of feasible points to sampled number of points for a 1,000,000 point random sampling is reported to be 0.000066 [3]. The optimum solution is [14.095, 0.84296] with an objective function value of -6961.81381. Both the constraints are active at the optimum.

Koziel and Michalewicz [3] solved this problem using an evolutionary algorithm with homomorphous mapping between an n-dimensional cube and a feasible search space. Ray *et al.* [10] solved this problem using an evolutionary algorithm incorporating intelligent partner selection and cooperative mating. Using the present socio-behavioural algorithm, results were obtained for 10 runs with a civilization size of 100 and 200 time steps. The best, average and worst objective function values obtained are [-6938.9396, -6726.1586, -6405.1804]. For the best objective function value obtained, the number of function evaluations is 15,656, the values of the two variables are [14.105, 0.8633] and the constraint values are [0.0127, 0.0072]. Table I shows the comparison of results.

4.2 Welded Beam Design

The second example deals with a welded beam design that is a well-studied single objective optimization problem that aims to minimize the cost of the beam subject to constraints on shear stress, bending stress, buckling load and end deflection. The four continuous design variables are x_1, x_2, x_3 and x_4 which are, respectively, the h, l, t and b shown in Figure 3.

TABLE I Comparison of Results for the Two-Variable Constrained Optimization Problem.

	Number of function evaluations	Function value
Koziel and Michalewicz [3]	350,000	−6901.5 (best) −6192.2 (average) −4236.7 (worst)
	1,400,000	−6952.1 (best) −6342.6 (average) −5473.9 (worst)
Ray et al. [10]	39,164	−6819.0391 (best) −6773.0078 (average) −6525.8374 (worst)
Present	15,656	−6938.9396 (best) −6726.1586 (average) −6405.1804 (worst)

The length L is assumed to be specified at 14 inch. The detailed formulation of the problem is as follows:

$$\begin{aligned}
\text{Minimize}\quad & f(\mathbf{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)\\
\text{subject to}\quad & \tau(\mathbf{x}) - \tau_{\max} \le 0\\
& \sigma(\mathbf{x}) - \sigma_{\max} \le 0\\
& x_1 - x_4 \le 0\\
& 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0\\
& 0.125 - x_1 \le 0\\
& \delta(\mathbf{x}) - \delta_{\max} \le 0\\
& P - P_C(\mathbf{x}) \le 0
\end{aligned}$$

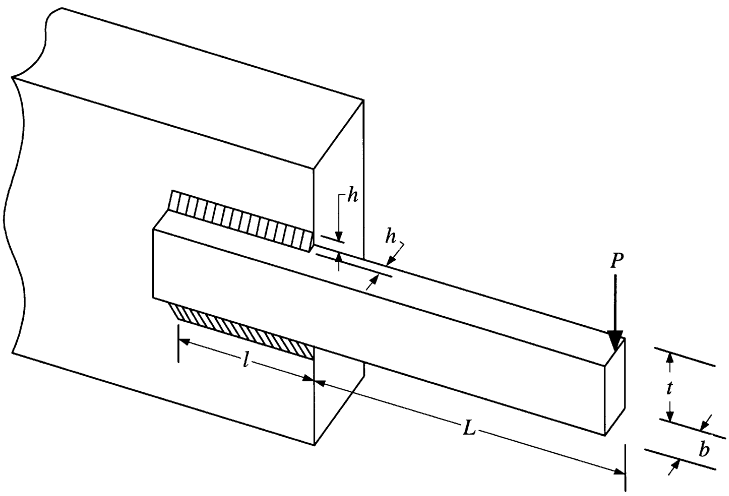

FIGURE 3 The welded beam structure.

The other parameters are defined as follows:

$$\tau(\mathbf{x}) = \sqrt{(\tau')^2 + \frac{2\tau'\tau'' x_2}{2R} + (\tau'')^2}$$
$$\tau' = \frac{P}{\sqrt{2}x_1x_2}$$
$$\tau'' = \frac{MR}{J}$$
$$M = P\left(L + \frac{x_2}{2}\right)$$
$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$
$$J = 2\left\{\frac{x_1x_2}{\sqrt{2}}\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$
$$\sigma(\mathbf{x}) = \frac{6PL}{x_4x_3^2}$$
$$\delta(\mathbf{x}) = \frac{4PL^3}{Ex_4x_3^3}$$
$$P_C(\mathbf{x}) = \frac{4.013\sqrt{EGx_3^2x_4^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

where $P = 6000$ lb., $L = 14$ in, $\delta_{\max} = 0.25$ in, $E = 30 \times 10^6$ psi, $G = 12 \times 10^6$ psi, $\tau_{\max} =$ 13,600 psi, $\sigma_{\max} = 30{,}000$ psi, $0.1 \leq x_1 \leq 2.0$, $0.1 \leq x_2 \leq 10.0$, $0.1 \leq x_3 \leq 10.0$ and $0.1 \leq x_4 \leq 2.0$.

Deb [11] solved this problem using a simple genetic algorithm with a traditional penalty function, Ragsdell and Phillips [12] used geometric programming while Siddall [13] solved this problem with a suite of optimization techniques: ADRANS – Gall's adaptive random search with penalty function; APPROX – Griffith and Stewart's successive linear approximation; DAVID – Davidon-Fletcher-Powell with penalty function; MEMGRD – Miele's memory gradient with penalty function; SIMPLEX – Simplex method with penalty function and RANDOM – Richardson's random method.

With a civilization size of 100 over 200 time steps, the best, average and worst objective function values obtained after 10 runs are [2.4426, 2.5215, 2.6315]. For the best objective value obtained, the number of function evaluations is 19,259, the values of the four variables are [0.2407, 6.4851, 8.2399, 0.2497] and the constraint values are [−129.8545, −270.4023, −0.009008, −2.9663, −0.1157, −0.2343, −372.4990]. For the run that produced the best objective value, the iteration history of the best objective value attained is plotted in Figure 4. Table II provides a comparison of results with the various other methods (it may be worth noting that this comparison has been done with results of those researchers who solved this problem with the exact same formulation as presented here, while there exist other results in the literature for this welded beam design based on a slightly different problem formulation.) The results obtained from the present method are comparable to those of the genetic algorithm by Deb [11] although Deb used only 4500 function evaluations.

4.3 Speed Reducer Design

The third example is a more complicated example of a speed reducer design. This has been reported to pose difficulties for various optimization algorithms in finding the feasible space.

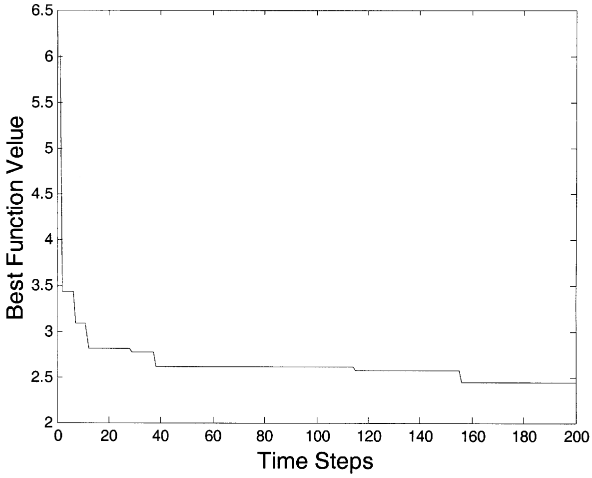

FIGURE 4 Iteration history of the best objective function value attained for the welded beam structure.

The weight of the speed reducer is to be minimized subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stresses in the shafts. The variables $x_1, x_2, \ldots, x_7$ are the face width, module of teeth, number of teeth in the pinion, length of the first shaft between bearings, length of the second shaft between bearings and the diameter of the first and second shafts. This is an example of a mixed integer programming problem. The third variable x_3 (number of teeth) is of integer value while all other variables are continuous.

$$\begin{aligned}
\text{Minimize} \quad & f(\mathbf{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) \\
& - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) \\
& + 0.7854(x_4x_6^2 + x_5x_7^2) \\
\text{subject to} \quad & \frac{27}{x_1x_2^2x_3} - 1 \leq 0 \\
& \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0 \\
& \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \leq 0 \\
& \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0
\end{aligned}$$

TABLE II Comparison of Results for the Welded Beam Design.

	Siddall [13]	*Ragsdell and Phillips [12]*	*Deb [11]*	*Present*
x_1	0.2444	0.2455	0.2489	0.2407
x_2	6.2819	6.1960	6.1730	6.4851
x_3	8.2915	8.2730	8.1789	8.2399
x_4	0.2444	0.2455	0.2533	0.2497
Objective	2.3815	2.3859	2.4431	2.4426

$$\frac{\left[(745x_4/(x_2x_3))^2 + 16.9 \times 10^6\right]^{1/2}}{110.0x_6^3} - 1 \leq 0$$

$$\frac{\left[(745x_5/(x_2x_3))^2 + 157.5 \times 10^6\right]^{1/2}}{85.0x_7^3} - 1 \leq 0$$

$$\frac{x_2x_3}{40} - 1 \leq 0$$

$$\frac{5x_2}{x_1} - 1 \leq 0$$

$$\frac{x_1}{12x_2} - 1 \leq 0$$

$$\frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0$$

$$\frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0$$

where $2.6 \leq x_1 \leq 3.6$, $0.7 \leq x_2 \leq 0.8$, $17 \leq x_3 \leq 28$, $7.3 \leq x_4 \leq 8.3$, $7.8 \leq x_5 \leq 8.3$, $2.9 \leq x_6 \leq 3.9$ and $5.0 \leq x_7 \leq 5.5$.

A detailed description of this single objective problem with 11 behavioural constraints is outlined in Rao [14]. Li and Papalambros [15] used the global optimization knowledge to solve this example. Kuang *et al.* [16] solved the same using the Taguchi method. With a civilization size of 100 for 200 time steps, the best, average and worst objective values obtained after 10 runs are [3008.08, 3012.12, 3028.28]. The number of function evaluations for the best objective function obtained is 19,154, with the corresponding variable values [3.506122, 0.700006, 17, 7.549126, 7.859330, 3.365576, 5.289773] and the constraint values of [−0.075548, −0.199413, −0.456175, −0.899442, −0.013213, −0.001740, −0.702497, −0.001738, −0.582608, −0.079580, −0.017887]. Table III gives the comparison of results with the other methods. Although it appears that the present algorithm produced the poorest result, it may be worth noting that it did produce a strictly feasible solution whereas there are some constraint violations in the other results. The fifth and eleventh constraints are slightly violated in the results of [14], while the sixth and eleventh constraints in [15] are also violated due to the allowable tolerances used. There is a significant violation of the sixth constraint in the results of [16].

4.4 Pressure Vessel Design

The last example is the design of a cylindrical vessel that has both ends capped by hemispherical heads as shown in Figure 5. The objective is to minimize the total cost comprising of

TABLE III Comparison of Results for Speed Reducer Design.

	Rao [14]	*Li and Papalambros [15]*	*Kuang et al. [16]*	*Present*
x_1	3.5	3.5	3.6	3.506122
x_2	0.7	0.7	0.7	0.700006
x_3	17	17	17	17
x_4	7.3	7.299999	7.3	7.549126
x_5	7.3	7.715317	7.8	7.859330
x_6	3.35	3.350541	3.4	3.365576
x_7	5.29	5.286654	5.0	5.289773
Objective	2985.22	2994.4	2876.22	3008.08

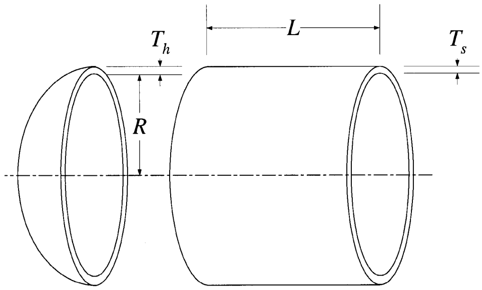

FIGURE 5 Centre and end section of the pressure vessel.

material, forming and welding costs. The four design variables are x_1 (thickness T_s of the shell), x_2 (thickness T_h of the head), x_3 (inner radius R) and x_4 (length L of the cylindrical section of the vessel, not including the head). x_1 and x_2 are to be in integral multiples of 0.0625 inch which are the available thicknesses of rolled steel plates. The radius x_3 and the length x_4 are continuous variables.

$$\begin{aligned}
\text{Minimize} \quad & f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\
\text{subject to} \quad & -x_1 + 0.0193x_3 \le 0 \\
& -x_2 + 0.00954x_3 \le 0 \\
& -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1{,}296{,}000 \le 0 \\
& x_4 - 240 \le 0
\end{aligned}$$

and $x_1 = 0.0625n_1$, $x_2 = 0.0625n_2$ where $1 \le n_1 \le 99$, $1 \le n_2 \le 9$, $10.0 \le x_3 \le 200.0$ and $10.0 \le x_4 \le 200.0$, and where n_1 and n_2 are integers.

This problem has been solved by Kannan and Kramer [17] using an augmented Lagrangian multiplier approach. Cao and Wu [18] applied an evolutionary programming model. Deb [19] solved the above problem using a genetic adaptive search (GeneAs) and Coello [20] used a genetic algorithm with self adaptive penalties for handling constraints.

With a civilization size of 100 over 200 time steps, the best, average and worst objective values obtained are [6171.00, 6335.05, 6453.65]. The solution with the best objective value was obtained in 12,630 function evaluations. The variable values for this solution are [0.8125, 0.4375, 41.9768, 182.2845] with constraint values of [−0.0023, −0.0370, −23420.5966, −57.7155]. Table IV shows the comparison of results for this example. The result obtained by the present algorithm is the best.

TABLE IV Comparison of Results for Pressure Vessel Design.

	Kannan and Kramer [17]	*Cao and Wu [18]*	*Deb [19]*	*Coello [20]*	*Present*
x_1	1.125	1.000	0.9375	0.8125	0.8125
x_2	0.625	0.625	0.5000	0.4375	0.4375
x_3	58.291	51.1958	48.3290	40.3239	41.9768
x_4	43.690	60.7821	112.6790	200.0000	182.2845
Objective	7198.042	7108.616	6410.381	6228.744	6171.000

5 CONCLUDING REMARKS

In this paper an optimization algorithm has been introduced that is based on a socio-behavioural concept of society and civilization. The results obtained for all the examples illustrate the capabilities of the proposed optimization algorithm. The concept of societies being formed among individuals is based on parametric spacing rather than on individual experiences over time (belief space), which is usually seen among other cultural algorithms. It is also quite different from evolutionary algorithms, where only the better performing individuals are given more chances to mate and generate children. The proposed algorithm improves the performance of all individuals in every society either through an intra- or inter-society information exchange. The use of Pareto ranking to deal with constraints reduces the number of additional inputs/parameters required for constraint handling, as well as providing an intelligent constraint information sharing which leads to reasonably fast convergence. Although the number of variables and types of problems solved in this work are limited and hence cannot substantiate the range or scope of application of the algorithm, the authors believe the algorithm's working principles and features are appealing and its effectiveness/efficiency can be further improved with future development work and experimentation.

Acknowledgement

This work was partially supported by the National Science and Technology Board of Singapore through the RICURF research grant EMT/98/013, which the authors gratefully acknowledge.

References

[1] Michalewicz, Z. (1995). A survey of constraint handling techniques in evolutionary computation methods. In: *Proceedings of the 4th Annual Conference on Evolutionary Programming*, MIT Press, Cambridge, MA, pp. 135–155.

[2] Michalewicz, Z. and Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, **4**(1), 1–32.

[3] Koziel, S. and Michalewicz, Z. (1999). Evolutionary algorithms, homomorphous mappings and constrained parameter optimization. *Evolutionary Computation*, **7**(1), 19–44.

[4] Ray, T., Tai, K. and Seow, K. C. (2001). Multiobjective design optimization by an evolutionary algorithm. *Engineering Optimization*, **33**(4), 399–424.

[5] Fogel, D. B. (1995). *Evolutionary Computation, Towards a New Philosophy of Machine Intelligence*. IEEE Press, New York.

[6] Reynolds, R. G. (1994). An introduction to cultural algorithms. In: *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, World Scientific, River Edge, New Jersey, pp. 131–139.

[7] Kennedy, J. and Spears, W. M. (1998). Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator. *Proceedings of the IEEE International Conference on Evolutionary Computation*, Anchorage, Alaska, USA, May 4–9, 1998.

[8] Eshelman, L. J. and Shaffer, J. D. (1993). Real coded genetic algorithms and interval schemata. *Foundations of Genetic Algorithms II*. Morgan Kaufmann, pp. 187–202.

[9] Deb, K. and Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex Systems*, **9**, 115–148.

[10] Ray, T., Tai, K. and Seow, K. C. (2000). An evolutionary algorithm for constrained optimization. In: Whitley, D., Goldberg, D. E., Cantú-Paz, E., Spector, L., Parmee, I. C. and Beyer, H. G. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*. Morgan Kaufmann, San Francisco, pp. 771–777

[11] Deb, K. (1991). Optimal design of a welded beam via genetic algorithms. *AIAA Journal*, **29**(11), 2013–2015.

[12] Ragsdell, K. M. and Phillips, D. T. (1976). Optimal design of a class of welded structures using geometric programming. *ASME Journal of Engineering for Industry Series B*, **98**(3), 1021–1025.

[13] Siddall, J. N. (1972). *Analytical Decision-making in Engineering Design*. Prentice-Hall, Englewood Cliffs.

[14] Rao, S. S. (1996). *Engineering Optimization*, 3rd ed. John Wiley, New York.

[15] Li, H. L. and Papalambros, P. (1985). A production system for use of global optimization knowledge. *ASME Journal of Mechanisms, Transmission, and Automation in Design*, **107**, 277–284.

[16] Kuang, J. K., Rao, S. S. and Chen, Li. (1998). Taguchi-aided search method for design optimization of engineering systems. *Engineering Optimization*, **30**, 1–23.
[17] Kannan, B. K. and Kramer, S. N. (1994). An augmented Langrangian multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *ASME Journal of Mechanical Design*, **116**, 318–320.
[18] Cao, Y. J. and Wu, Q. H. (1997). Mechanical design optimization by mixed variable evolutionary programming. In: *Proceedings of the 1997 International Conference on Evolutionary Computation*, Indianapolis, pp. 443–446.
[19] Deb, K. (1997). GeneAS: A robust optimal design technique for mechanical component design. *Evolutionary Algorithms in Engineering Applications*. Springer-Verlag, pp. 497–514.
[20] Coello, C. A. C. (1999). Self-adaptive penalties for GA based optimization. *Proceedings of the Congress on Evolutionary Computation*, **1**, 573–580.