Society and Civilization: An Optimization Algorithm Based on the Simulation of Social Behavior

Tapabrata Ray and K. M. Liew

***Abstract*—The ability of an individual to mutually interact is a fundamental social behavior that is prevalent in all human and insect societies. Social interactions enable individuals to adapt and improve faster than biological evolution based on genetic inheritance alone. This is the driving concept behind the optimization algorithm introduced in this paper that makes use of the intra and intersociety interactions within a formal *society and the civilization model* to solve single objective constrained optimization problems. A society corresponds to a cluster of points in the parametric space while a civilization is a set of all such societies at any given point of time. Every society has its set of better performing individuals (henceforth, referred as leaders) that help others in the society to improve through an intrasociety information exchange. The intrasociety information exchange results in the migration of a point toward a better performing point in the cluster that is analogous to an intensified local search around a better performing point. Leaders of a society on the other hand improve only through an intersociety information exchange that results in the migration of a leader from a society to another that is headed by better performing leaders. This process of leader migration helps the better performing societies to expand and flourish that correspond to a search around globally promising regions in the parametric space. In order to study the performance of the proposed algorithm, four well-studied, single objective constrained engineering design optimization problems have been solved. The results indicate that the algorithm is capable of arriving at comparable solutions using significantly fewer function evaluations and stands out as a promising alternative to existing optimization methods for engineering design. Futhermore, the algorithm employs a novel nondominance scheme to handle constraints that eliminates the problem of scaling and aggregation that is common among penalty-function-based methods.**

***Index Terms*—Constrained optimization, Pareto, social behavior.**

I. Introduction

EVOLUTIONARY computation refers to the metaphorical use of concepts, principles, and mechanisms extracted from our understanding about the evolution of natural systems to help solve complex computational problems. Substantial part of the research on evolutionary computation focused on the processes of natural selection and genetics [1]. Computational methods based on socio-behavioral models are fairly recent developments that are driven by the fact that individuals adapt and evolve faster through cultural evolution than by genetic inheritance [2]. In order to harness the benefits of a faster change in an individuals' performance, Reynolds and Chung [3] introduced a cultural algorithm based on socio-behavioral dynamics. In the cultural algorithm framework, individuals are described by a set of traits, and these traits are passed from a generation to another through the socially motivated operators. The ant colony model [4] is another innovative approach that draws its inspiration from natural ant systems, where individuals interact with each another indirectly through their pheromone trails. Yet, another novel paradigm of behavioral modeling was proposed by Kennedy and Eberhart [5] based on the concepts of a swarm. The particle swarm optimization (PSO) algorithm mimics the behavior of flying birds and their means of information exchange to solve optimization problems. Kennedy [6] observed that the performance of the swarm improved if the individuals were allowed to stereotype (i.e., align themselves toward its group's central tendencies). The algorithm presented in this paper aims to exploit the benefits of a faster change in an individuals' performance through the social interactions within a formal society and civilization model. The concept of the algorithm and the complex intra and intersociety interactions are derived out of two fundamental observations from nature.

A civilization emerges and advances due to cooperative relationships among its societies.

Individuals in a society interact with each another with an aim to improve.

A society in the current context refers to a set of mutually interacting individuals, while a civilization is a collection of all such societies at any given point of time. Individuals are fundamental social entities that interact with each other in a quest to improve. Such an improvement in an individual's performance is due to a meaningful information acquisition from a better-performing individual (leaders) belonging to the same society. The above intrasociety interaction between an average individual and its leader results in an improvement of an individual's performance, which in turn leads to the better performing societies over time. However, such an intrasociety interaction cannot improve the performance of its leaders. The leaders, unlike an average individual in the society, communicate and collaborate externally with the leaders of other societies in the civilization in order to improve. Such an intersociety information exchange among leaders results in the migration of leaders to developed societies led by better-performing leaders. The above process of leader migration drags the poor-performing societies toward the better-performing ones, thereby turning the focus of search to globally promising regions in the parametric space.

A civilization is created through a random initialization of a number of individuals. At every time instant, the individuals are

Manuscript received June 4, 2001; revised November 16, 2001 and August 8, 2003.

T. Ray is with Temasek Laboratories, National University of Singapore, Singapore 119260 (e-mail: tsltray@nus.edu.sg).

K. M. Liew is with the Nanyang Centre for Supercomputing and Visualization, and the School of Mechanical and Production Engineering, Nanyang Technological University, Singapore 639798 (e-mail: mkmliew@ntu.edu.sg).

Digital Object Identifier 10.1109/TEVC.2003.814902

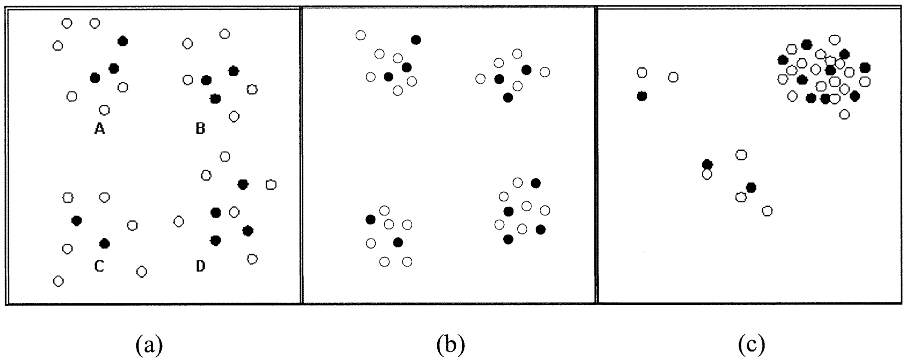

Fig. 1. (a) Clusters A, B, C, and D. (b) Individuals in every cluster migrating toward their leaders. (c) Leaders of the clusters migrated to cluster B.

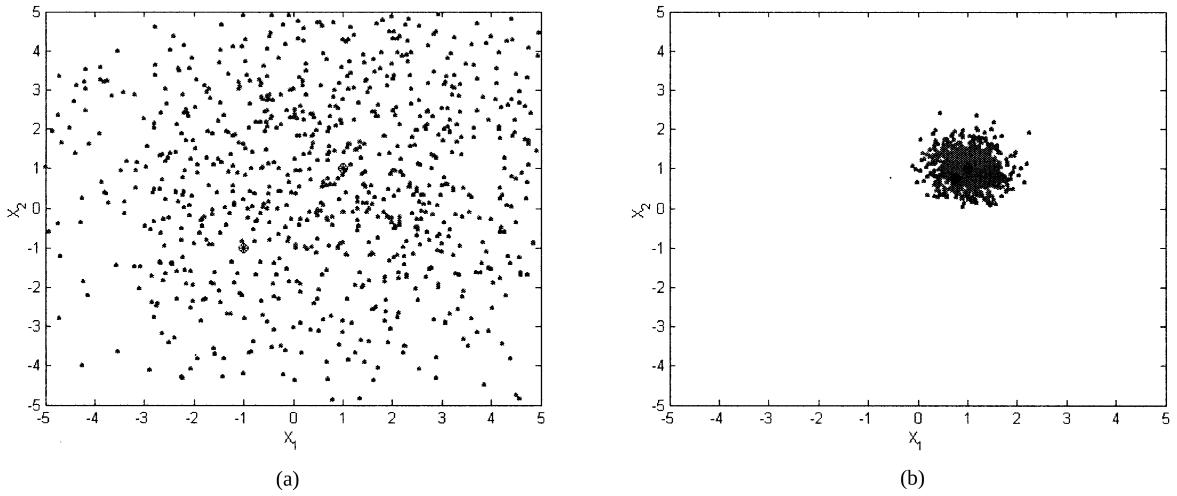

Fig. 2. Schematic diagram to illustrate information acquisition. (a) Leader at (1.0,1.0) and follower at (−1.0,−1.0). (b) Leader at (1.0,1.0) and follower at (0.75,0.75).

separated into a number of mutually exclusive clusters based on their Euclidean distance in the parametric space that correspond to the societies and a collection of all such clusters represent the civilization. For an unconstrained problem, the objective values of the individuals are used to generate the set of leaders, while a multilevel Pareto ranking scheme is implemented to generate the set of leaders for constrained problems. The individuals within a society extract information from its leader through an intrasociety interaction that translates to an intensified search around better-performing points. The leaders of different societies compete among themselves to attract leaders from other societies in order to expand its social boundary. 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 less- promising regions of the parametric space.

Fig. 1(a) schematically depicts the societies A, B, C, and D that form the civilization at an initial time instant. The individuals are clustered to form societies A, B, C, and D based on the parametric space. The leaders within the societies are marked using •, while an average individual is marked with O. Fig. 1(b) shows the migration of individuals of every cluster toward their leaders and Fig. 1(c) shows the state of the leaders in the civilization, where leaders of A, C, and D moved over toward the better-performing leaders of society B.

II. Related Works

Several mechanisms used in our proposed algorithm have similarities at least at a conceptual level with a number of existing optimization paradigms. The first is the use of multiple coevolving societies in our algorithm that exist in the form of multiple subpopulations in distributed genetic algorithm (GA) models. The second is an analogy between the mechanism of individual and leader migration from a society to another with the diffusion mechanisms used in distributed GA models. The third is the mechanism of information acquisition/sharing that is used in various machine-learning models (use of memory and development of operators for information acquisition). One might also argue that the method of leader selection is similar to the archival of Pareto solutions used in multiobjective optimization. In this section, we would like to briefly discuss the ideas that bear similarities with mechanisms used in the proposed algorithm before highlighting their differences.

Distributed GAs originated from Darwin's observation that in a natural system, spatial environment plays a decisive role in the evolution of individuals. Mountains and rivers are natural barriers between populations and occasional migration of individuals takes place between neighboring populations. Distributed evolutionary algorithms exploit this social model by maintaining distinct subpopulations. Better-performing individuals are copied to neighboring subpopulations at regular intervals. Voigt *et al.* [7] proposed a distributed evolutionary algorithm for a multiprocessor system, where the subpopulations evolved in separate processors and the migration of individuals from a subpopulation to another was guided by the topology. Startweather *et al.* [8] simulated a distributed evolutionary algorithm in a single processor environment by maintaining multiple subpopulations instead of a conventional single population and allowed occasional migrations of individuals between subpopulations. In both the above models, the subpopulations do not represent mutually exclusive regions in the parametric space. A spatially separated model was used by Ursem [9] in his multinational evolutionary algorithm. No explicit clustering method was used in the model although a fitness-topology function was used to decide upon the relationship between a point and a cluster. The algorithm was aimed to find all peaks of a multimodal function in an unconstrained optimization problem.

The concept of maintaining nonoverlapping societies in our proposed algorithm is meant to maintain diversity that has been known to improve the performance of GAs for multimodal problems. The concept of selecting a nearest leader for information acquisition can be linked to diffusion models in distributed GAs, where parents are chosen from neighboring populations.

From the point of sharing or acquiring information between individuals, links can be established with existing machine learning concepts where one aims to derive information from others and in the process avoid unpromising regions or exploit promising regions of the search space. Various methods proposed by Sebag and Schoenaeur [10]–[12] and Ravise and Sebag [13] deal with learning that enables individuals to avoid unpromising regions of the search space using its past memory and flee operators. In an attempt to make use of the memory of a population, Reynolds [14] introduced a cultural algorithm model that had two levels of evolution. At the macroevolutionary level, a belief space stored generalizations of individual's experiences that were updated from time to time while at the microevolutionary level, individual solutions were generated biased by existing beliefs. All the approaches discussed above can be viewed as methods to extract information from an existing population and subsequently use them to control the evolution of the individuals. There are also research efforts that concentrate on developments of operators such as simulated binary crossover [15], unimodal normal distribution crossover [16], parent centric crossover [17], or even operators used in differential evolution [18] that make use of interparent distances to allow meaningful information sharing among individuals. From the point of retaining useful information, similarities exist with archival methods used by a host of multiobjective algorithms [19].

It is well known that the presence of constraints significantly affects the performance of all optimization algorithms. There have been a number of approaches to handle constraints in the domain of evolutionary methods including rejection of infeasible individuals, penalty functions and their variants, repair methods, use of decoders, separate treatment of constraints and objectives, and hybrid methods incorporating knowledge of constraint satisfaction. Michalewicz [20] provides a comprehensive review on constraint-handling methods. All the cited methods have limited success as they are problem dependent and require a number of additional inputs. Penalty functions using static, dynamic, or adaptive concepts have been developed over the years but they suffer from common problems of aggregation and scaling. Repair methods are based on additional function evaluations, while the decoders and special operators or constraint-satisfaction methods are problem specific and cannot be used to model a generic constraint. Separate treatment of constraints and objectives is an interesting concept that eliminates the problem of scaling and aggregation that has been adopted in our proposed algorithm. Instead of a single measure of fitness, our proposed algorithm assigns two scores to every individual; one based on the objective while the other based on constraint satisfaction.

Our proposed algorithm is built upon the concept of learning from the good individuals unlike some of the learning models listed above that tend to learn from the bad, i.e., avoid unpromising regions of the search space. Another significant difference between the proposed algorithm and swarm or the cultural algorithms is the absence of individual or population memory. At every time instant, an average individual is only aware of other individuals that belong to the same society, while the leaders of a society are aware of all other leaders in the civilization. This multitier information hierarchy is close to what is observed in nature.

In our proposed algorithm, our leader identification mechanism is truly unique. The algorithm employs an adaptive criterion for leader selection. In a society where there are no feasible individuals, the leaders are the ones with constraint rank $= 1$, while in a society where all the individuals are feasible, the leaders are the ones with objective rank $<$ average objective rank. The use of the nondominated rank derived from the constraint matrix is particularly useful to drive the societies toward feasibility. If a society does not have any feasible solution, the individuals with rank constraint $= 1$ are the ones with minimal constraint violation. However, once there is a feasible individual in the society, it will assume a rank of one. This mechanism tends to move the societies toward feasibility first before improving the objective performance of the individuals. Since the global leaders (i.e., the best leaders of the civilization) do not move, it in a way ensures elitism. In a situation where the civilization does not have any feasible solution, there is likely to be a number of global leaders. Copying the global leaders is similar to Pareto archival processes used in multiobjective problems where nondominated solutions are stored in an archive.

The mathematical details of the algorithm are presented in Section III. In order to study the performance of the algorithm, four well-studied, single-objective, constrained engineering design optimization problems have been solved. The best result obtained by our algorithm over 50 independent trials is compared with those reported in literature. The major conclusions are summarized in Section V.

III. Society and Civilization Algorithm

The algorithm is described in the context of a constrained minimization problem of the following form:

Minimize

$$f(x) \tag{1}$$

subject to

$$g_i(\mathbf{x}) \geq 0, \qquad i = 1, 2, \ldots, q \tag{2}$$

$$h_j(\mathbf{x}) = 0, \qquad j = 1, 2, \ldots, r \tag{3}$$

where there are q inequality and r equality constraints, $\mathbf{x} = [\mathrm{x}_1\ \mathrm{x}_2\ \cdots\ \mathrm{x}_n]$ is the vector of n design variables. The equality constraints are transformed to a set of inequalities as $h_j(\mathbf{x}) \leq \delta$ and $h_j(\mathbf{x}) \geq -\delta$ (assuming δ is small positive quantity). Thus, r equality constraints will give rise to $2r$ inequalities, and the total number of inequalities (i.e., constraints) for the problem is denoted by s, where $s = q + 2r$. For each individual, c denotes the constraint satisfaction vector given by $\mathbf{c} = [c_1\ c_2\ \cdots\ c_s]$ where

$$c_i = \begin{cases} 0, & \text{if } i\text{th constraint satisfied} \\ & \quad i = 1, 2, \ldots, s \\ -g_i(\mathbf{x}), & \text{if } i\text{th constraint violated} \\ & \quad i = 1, 2, \ldots, q \\ h_i(\mathbf{x}) - \delta, & \text{if } i\text{th constraint violated} \\ & \quad i = q+1, q+2, \ldots, q+r \\ \delta - h_i(\mathbf{x}), & \text{if } i\text{th constraint violated} \\ & \quad i = q+r+1, q+r+2, \ldots, s. \end{cases} \tag{4}$$

For the above c_is, $c_i = 0$ indicates the ith constraint is satisfied, whereas $c_i > 0$ indicates the violation of the constraint. The CONSTRAINT matrix for a set of G solutions assume the form

$$\mathbf{CONSTRAINT} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1s} \\ c_{21} & c_{22} & \cdots & c_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{G1} & c_{G2} & \cdots & c_{Gs} \end{bmatrix} \tag{5}$$

where elements C_{ij}, $i = 1, \ldots, G$, $j = 1, \ldots, s$ denote the constraint satisfaction of constraint j for individual i. Among these G individuals, all nondominated individuals based on the constraint matrix are assigned a rank of one. The rank 1 individuals are removed from the set r and the new set of nondominated individuals is assigned a rank of two. The process is continued until every individual is assigned a rank. Rank = 1 in the constraint matrix indicates that the individual is nondominated. It can be observed from the constraint matrix that when all the individuals are infeasible, the nondominated ones will have a rank of one. Whenever there is one or more feasible individual, the feasible solutions will take over as rank 1 individuals.

The pseudocode of the main algorithm is presented as follows.

```
1. $t \leftarrow 0$
2. **Generate** $N$ individuals representing a
   civilization: $\mathrm{Civ}(t) = \{I_1, I_2, \ldots, I_N\}$ uni-
   formly in the parametric space.
3. **Evaluate** each individual: Compute the
   objective and constraints, i.e., $f(I_i)$
   and $C_j(I_i)$; for $i = 1, 2, \ldots, N$ individuals
   and $j = 1, \ldots, s$ constraints.
4. **Build Societies:** Create $K(t)$
   clusters representing societies:
   $\mathrm{Soc}_1(t), \ldots, \mathrm{Soc}_{K(t)}(t)$ such that
   $\bigcup_{i=1}^{K(t)} \mathrm{Soc}_i(t) = \mathrm{Civ}(t)$ and $\mathrm{Soc}_i(t) \cap \mathrm{Soc}_j(t) = \phi$
   for $i \neq j$
5. **Identify Leaders:** Leaders of every So-
   ciety: $\mathrm{Soc\_}L_i(t) \subset \mathrm{Soc}_i(t)$, $i = 1, \ldots, K(t)$
6. **Move:** For each $I_j \in \mathrm{Soc}_i(t)$ and $I_j \notin \mathrm{Soc\_}L_i(t)$
   move to a new location
   a) $\mathrm{Civ}(t+1) \leftarrow \phi$
   b) **For each** $I_j \in \mathrm{Soc}_i(t)$ and $I_j \notin \mathrm{Soc\_}L_i(t)$ **Do**
         **Move** $I_j \rightarrow I_j^1$: guided by the
         nearest leader of the society.
         $\mathrm{Civ}(t+1) \leftarrow \mathrm{Civ}(t+1) \cup \{I_j^1\}$
7. **Identify Leaders:** Leaders of the Civi-
   lization: $\mathrm{Civ\_}L(t) \subset \bigcup_{i=1}^{K(t)} \mathrm{Soc\_}L_i(t)$
8. **Move:** Move all leaders other than civi-
   lization leaders to a new location.
   a) **For each** $I_j \in \mathrm{Soc\_}L_i(t)$ and $I_j \notin \mathrm{Civ\_}L_i(t)$
      **Do**
         **Move** $I_j \rightarrow I_j^1$: guided by the
         nearest civilization leader.
         $\mathrm{Civ}(t+1) \leftarrow \mathrm{Civ}(t+1) \cup \{I_j^1\}$
9. $\mathrm{Civ}(t+1) \leftarrow \mathrm{Civ\_}L(t) \cup \mathrm{Civ}(t+1)$
10. $t \leftarrow t+1$
11. If $t < T_{\max}$ then Go to 3 Else Stop.
```

The pseudocode of the **Build Society** function is presented as follows.

```
1. Let $S = \{I_1, I_2 \ldots I_N\}$ denote the set of N
   Individuals.
2. $H \leftarrow \phi$; Assign the set of hubs as
   empty.
3. $H_1 \leftarrow I_j$; assign any random individual
   as the first hub.
4. $H_2 \leftarrow I_j$; assign the $j$ individual as
   the second hub if $D(H_1 - I_j)$ is maximum,
   where $D(H_1 - I_j)$ denotes the Euclidean
   distance between $H_1$ and $I_j$.
5. $H \leftarrow H_1 \cup H_2$; $k = 2$
6. For all $I_j \in S$ and $I_j \notin H$; assign them to
   $k$ *clusters* $\{C_1, C_2, \ldots, C_k\}$
   a) $I_j \in C_p$; if $D(H_p - I_j)$ is the minimum
   of $D(H_i - I_j)$ for $i = 1, 2, \ldots, k$ clusters.
7. If maximum of $D(H_i - I_j)$ for $i = 1, 2 \ldots, k$
   clusters > Average of all $D(H_i - H_j)$s
   a) $k = k+1$
   b) $H_k = I_j$
   c) Go to Step 6
```

The pseudocode of the **Leader Identification** function is presented as follows.

```
1. Let $S = \{I_1, I_2, \ldots, I_N\}$ denote the set of N
   individuals.
2. Compute the objective and constraints
   i.e., $f(I_i)$ and $C_j(I_i)$; for $i = 1, 2, \ldots, N$
   individuals and $j = 1, 2, \ldots, s$
   constraints.
```

TABLE I
RESULTS OF THE WELDED BEAM EXAMPLE

	Civ. Size	Gen./ Time Steps	Best	Average/ Median.	Worst	Standard Deviation	Function Evals.
Present	40	1,000	2.3854347	3.2551371 3.0025883	6.3996785	0.9590780	33,095
Deb[22]	80	500	2.38119	----------- 2.39289	2.64583	-----------	40,080
Deb[21]	80	4,000	2.38119	----------- 2.39203	2.64583	-----------	320,080

```
3. Compute the nondominated rank of every
   solution: $RC_i$ based on the Constraint
   matrix and $RO_i$ based on the Objective
   Vector.
4. Set of leaders $L \in S$ is formed via the
   following steps;
   a) $L \leftarrow \varphi$
   b) $L \leftarrow I_i$: if $RC_i = 1$ and $RO_i \leq (1/N)(\sum_{j=1}^{N} f(I_i)$
   c) If $L$ is empty then: $L \leftarrow I_i$ for all
   $RC_i = 1$.
```

The pseudocode of the **Move** function for Information Acquisition is presented as follows.

```
1. Scale every variable between 0–1 using
   the maximum and minimum value of vari-
   ables in the population.
2. $D = \sum_{j=1}^{M} \left(I_L^j - I_F^j\right)^2$; $j = 1, \ldots, M$ vari-
   ables; $I_L^j$ denotes the $j$th variable of
   the leader($P$) and $I_F^j$ denotes the $j$th
   variable of the follower($F$).
3. $C(i) = P(i) + N(\mu = 0, \sigma).D$; where $\sigma$ is the
   variance of the normal distribution,
   $i = 1, \ldots, M$ variables
4. Transform $C(i)$s back to original scale
   to get the new location of the indi-
   vidual F.
```

IV. NUMERICAL EXAMPLES

In order to study the performance of the algorithm, four well-studied engineering design examples have been solved and the best results obtained by our algorithm over 50 trials have been compared with those reported in literature. Each example has been solved using a civilization size of 10 N (where N is the number of variables for the problem) and it has been allowed to evolve over 1000 time steps. The computations were performed on a Pentium III, 1 GHz machine. We have also presented our results using 5 and 20 N civilization sizes and studied the performance with different σs (0.25, 0.5, 0.75,1.0).

A. Welded Beam Design

The first example deals with a welded beam design that is a well-studied single-objective optimization problem. It aims to minimize the cost of the beam subject to constraints on shear stress, bending stress, buckling load, and the end deflection. The four continuous design variables are thickness of the beam x_1, width of the beam x_2, length of the weld x_3, and the weld thickness x_4. The problem has been solved by a number of researchers; Deb [21], [22], Ragsdell and Phillips [23], and Siddall [24]. The formulation of the problem is presented in the Appendix.

We have used a civilization size of 40 to solve this problem. The results are reported based on 50 independent trials in which every time the civilization has been allowed to evolve over 1000 time steps. The best, worst, average, median, and the standard deviation is reported in Table I. The run that resulted in the best objective function value performed 33 095 function evaluations using 3.35 s of CPU time. A comparison of results is presented in Table II while a convergence plot is presented in Fig. 3(a). From Table II, it is clear that the present algorithm reported better results when compared with Siddal [24], Ragsdell and Phillips [23], and Deb [21]. However, the results reported by Deb [22] is the best-known result to this problem. From the convergence plot in Fig. 3(a), one can observe that the convergence is quite fast.

B. Spring Design

A tension/compression spring design is considered next that is to be designed for minimum weight subject to constraints on minimum deflection, shear stress, surge frequency, and limits on outside diameter. This design optimization problem involves three continuous variables and four nonlinear inequality constraints. The problem has been studied by Belegundu [25], Arora [26], Coello [27], and Ray and Saini [28]. The mathematical formulation of the problem is presented in the Appendix.

We have used a civilization size of 30 to solve this problem. The results are reported based on 50 independent trials, in which every time the civilization has been allowed to evolve over 1000 time steps. The best, worst, average, median, and the standard deviation is reported in Table III. The run that resulted in the best objective function value performed 25 167 function evaluations and required 0.84 s of CPU time. It is also interesting to note that our result using 25 167 function evaluations is better than the results of Coello [27], who reported using 900 000 function evaluations. A comparison of results is presented in Table IV while a convergence plot is presented in Fig. 3(b). From Table IV, the present algorithm reported the best-known result to this problem. The same nature of fast convergence can be observed from Fig. 3(b).

TABLE II
COMPARISON OF RESULTS FOR WELDED BEAM DESIGN

	Present	Deb [22]	Deb [21]	Reklaitis [35]	Ragsdell [23]	Siddal [24]
x_1	0.2444382760	---------	0.2489	0.2444	0.2455	0.2444
x_2	6.2379672340	---------	6.1730	6.2187	6.1960	6.2819
x_3	8.2885761430	---------	8.1789	8.2915	8.2730	8.2915
x_4	0.2445661820	---------	0.2533	0.2444	0.2455	0.2444
Best	2.3854347	**2.38119**	2.43	2.38116	2.386	2.3918
No. of Evals.	33,095	40,080	4,500	---------	---------	--------

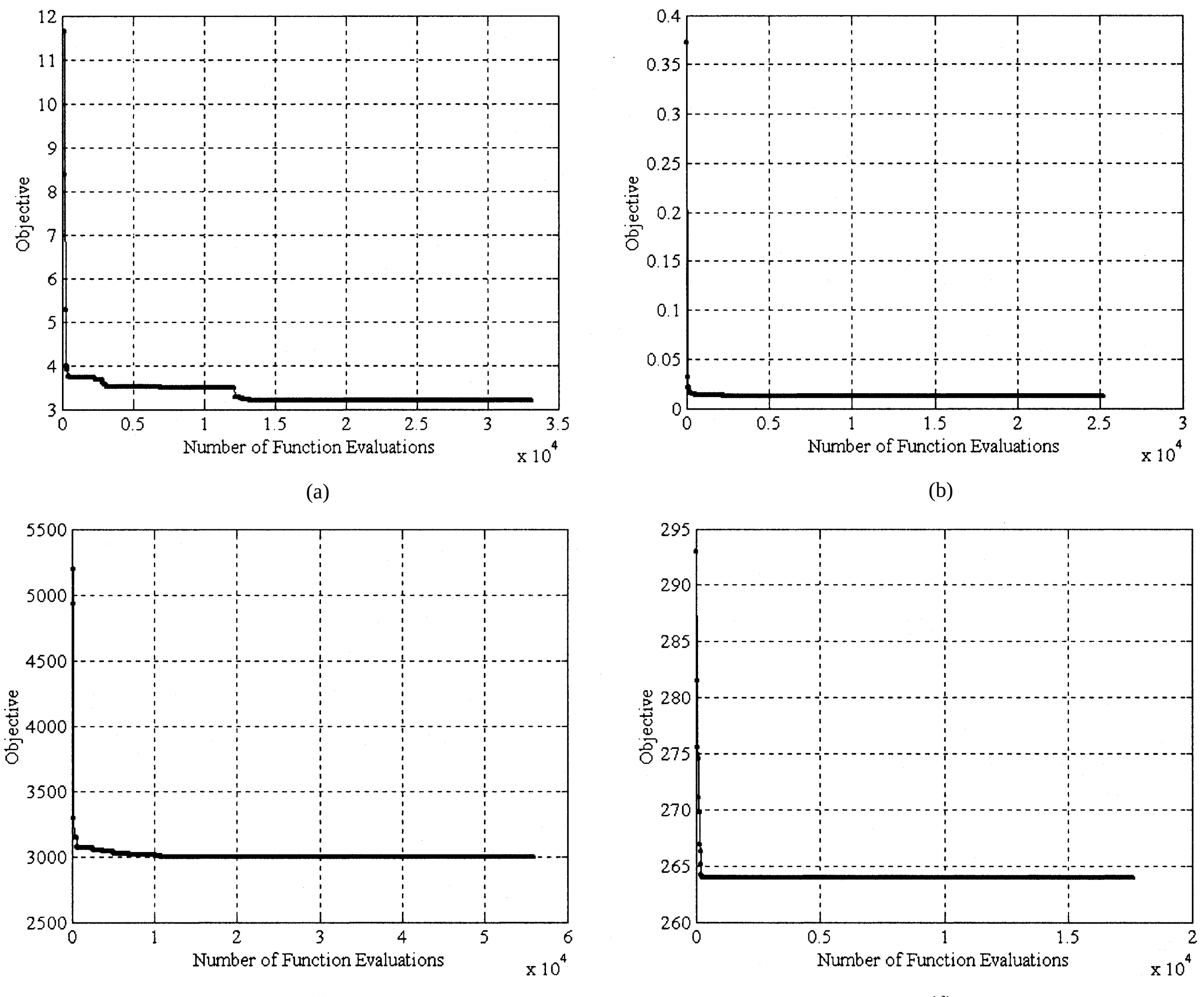

Fig. 3. Convergence plots for the numerical examples. (a) Welded beam design. (b) Spring design. (c) Speed reducer design. (d) Three-bar truss design.

C. Speed Reducer Design

A speed reducer problem is considered next. Previous methods had problems in locating a feasible solution [28]. 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 $x_1 \ldots x_7$ 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 first and second shafts. Many researchers Rao [29], Li and Papalambros [30], Kuang *et al.* [31], and Azarm and Li [32] have reported solutions to this problem. However, the solutions reported by all the above-mentioned researchers are infeasible, including the one that appears in the MDO Test Suite obtained using the constrained minimizer CONMIN.

We have used a civilization size of 70 to solve this problem. The results are reported based on 50 independent trials in which every time the civilization has been allowed to evolve over 1000 time steps. The best, worst, average, median, and the standard deviation is reported in Table V. The run that resulted in the best objective function value performed 54 456 function evaluations and required 9.62 s of CPU time. A comparison of results is presented in Table VI while a convergence plot is presented in Fig. 3(c). From Table VI, the present algorithm

TABLE III
RESULTS OF SPRING DESIGN

	Civ. Size	Gen./ Time Steps	Best	Average/ Median.	Worst	Standard Deviation	Function Evals.
Present	30	1,000	0.01266924934	0.012922669 0.012922669	0.016717272	5.92E-04	25,167
Coello [27]	-----	------	0.01270478	0.01276920 0.01275576	0.01282208	----------	900,000

TABLE IV
COMPARISON OF RESULTS FOR SPRING DESIGN

	Present	Arora [26]	Belegundu [25]	Coello [27]	Ray & Saini [28]
x_1	0.368158695	0.399180	0.315900	0.351661	0.321532
x_2	0.0521602170	0.053396	0.05	0.051480	0.050417
x_3	10.6484422590	9.185400	14.25	11.632201	13.979915
Best	**0.01266924934**	0.0127302737	0.0128334375	0.0127047834	0.013060
No. of Evals.	25,167	-----------	-----------	900,000	1,291

TABLE V
RESULTS FOR SPEED REDUCER DESIGN

	Civ. Size	Gen./ Time Steps	Best	Average/ Median.	Worst	Standard Deviation	Function Evals.
Present	70	1,000	2994.744241	3001.758264 3001.758264	3009.964736	4.00914232	54,456

TABLE VI
COMPARISON OF RESULTS FOR SPEED REDUCER DESIGN

	Present	Li & Pap. [30]	Kuang et al [31]	Azarm & Li [32]	MDO (NASA)	Rao [29]
x_1	3.50000681	3.5	3.6	3.5	3.5	3.5
x_2	0.70000001	0.7	0.70	0.7	0.7	0.70
x_3	17	17	17	17	17	17
x_4	7.32760205	7.3	7.3	7.3	7.3000002	7.3
x_5	7.71532175	7.71	7.8	7.71	7.3000002	7.3
x_6	3.35026702	3.35	3.4	3.35	3.3502145	3.35
x_7	5.28665450	5.29	5.0	5.29	5.2865176	5.29
Best	**2994.744241**	2996.309776	2876.117623	2996.309776	2985.151875	2987.298504
No. of Evals.	54,456	-----------	-----------	-----------	-----------	-----------

reported the best-known result to this problem. From the convergence plot in Fig. 3(d), one can observe a very fast convergence.

The results of Rao [29], Li and Papalambros [30], and Azarm and Li [32] violate constraints presented in [22] and [28]. Results of reported at the MDO Web Site violate [22], [23], and [28] while the results of Kuang *et al.* [31] violate [33]. A comprehensive discussion on the above design problem appears in Ray [33].

D. Three-Bar Truss Design

The last example deals with the design of a three-bar truss structure in which the volume is to be minimized subject to stress constraints [34]. We have used a civilization size of 20 to solve this problem. The results are reported based on 50 independent trials, in which every time the civilization has been allowed to evolve over 1000 time steps. The best, worst, average, median, and the standard deviation is reported in Table VII. The run that resulted in the best objective function value performed 17 610 function evaluations and required 0.46 s of CPU time. A comparison of results is presented in Table VI while a convergence plot is presented in Fig. 3(d). From Table VIII, the present algorithm reported the best-known result to this problem.

In order to visually illustrate the behavior of the algorithm, the distribution of solutions in the initial civilization is presented in Fig. 4(a). One can observe that although the individuals in the initial civilization are fairly distributed within the design space, individuals in the final civilization appear clustered around a region. Fig. 4(b) provides a closer look into the area of the final civilization that has feasible individuals.

E. Sensitivity Studies

The effect of the parameter σ on the behavior of our algorithm has been studied empirically. We have used four different values of σ [0.25, 0.5, 0.75 and 1.0] and observed the best, worst, median, and average performance based on 50 independent runs for

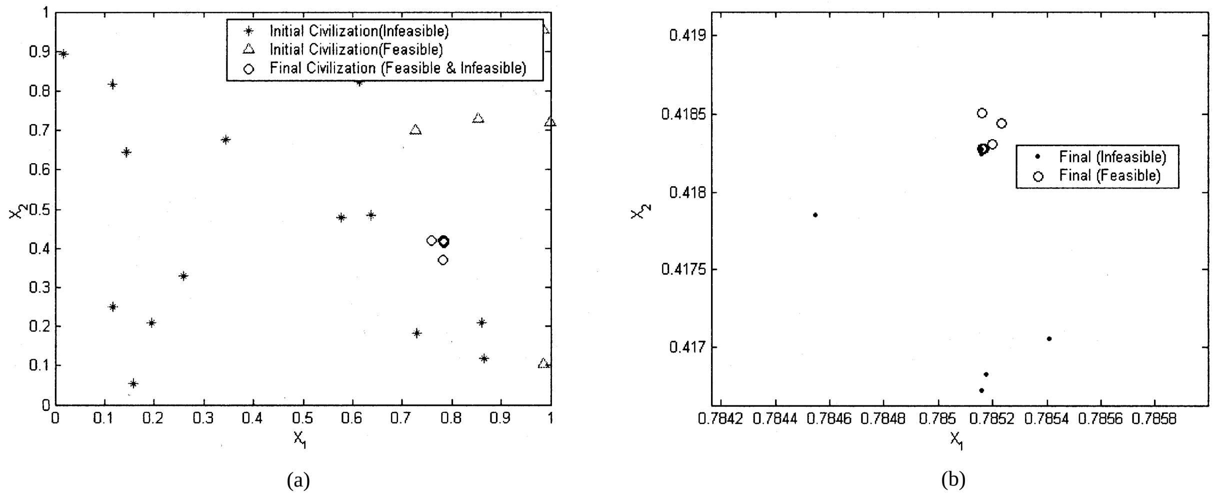

Fig. 4. (a) Initial civilization for the three bar truss design example. (b) Final civilization zoomed around feasible solutions.

TABLE VII
RESULTS FOR THREE-BAR TRUSS DESIGN

	Civ. Size	Gen./ Time Steps	Best	Average/ Median.	Worst	Standard Deviation	Function Evals.
Present	20	1,000	263.8958	263.9033 263.8989	263.96975	0.0125797	17,610

TABLE VIII
COMPARISON OF RESULTS FOR THREE-BAR TRUSS DESIGN

	Present	Hernendez [34]	Ray & Saini [28]
x_1	0.7886210370	0.788	0.795
x_2	0.4084013340	0.408	0.395
Best	**263.8958466**	263.9	264.3
No. of Evals.	17,610	-----------	2,712

TABLE IX
EFFECT OF FOR THE WELDED BEAM DESIGN

σ	Best	Worst	Average	Median
0.25000000	2.42169498	5.36210543	3.45962026	3.38883787
0.50000000	2.41539377	6.25703411	3.45764020	3.27937448
0.75000000	2.43105695	7.66182770	3.46714777	3.16772889
1.00000000	2.38543475	6.39967857	3.25513714	3.00258830

TABLE X
EFFECT OF FOR THE SPRING DESIGN

σ	Best	Worst	Average	Median
0.25000000	0.01266535	0.01405852	0.01297739	0.01284193
0.50000000	0.01266589	0.01533206	0.01308559	0.01290300
0.75000000	0.01266566	0.01530473	0.01303836	0.01280347
1.00000000	0.01266526	0.01671727	0.01292267	0.01292267

TABLE XI
EFFECT OF FOR THE SPEED REDUCER DESIGN

σ	Best	Worst	Average	Median
0.25000000	2996.57091885	3009.92687983	3003.44687120	3003.95235758
0.50000000	2995.65215018	3012.47947177	3003.55979257	3003.92618875
0.75000000	2995.06778674	3026.89326738	3003.16428810	3002.99708533
1.00000000	2994.74424174	3009.96473694	3001.75826452	3001.75826452

TABLE XII
EFFECT OF FOR THE THREE BAR TRUSS DESIGN

σ	Best	Worst	Average	Median
0.25000000	263.89587031	264.11080426	263.90755884	263.89889406
0.50000000	263.89584531	263.97073561	263.90405576	263.89842043
0.75000000	263.89584584	263.96700713	263.90635476	263.90065440
1.00000000	263.89584654	263.96975638	263.90335672	263.89893782

all the examples. The results are presented in Tables IX–XII. It is clear that the best result of the examples hardly vary with different choices of σ.

We have also studied the effect of civilization size for which we conducted our optimization runs using civilization sizes of 5, 10, and $20N$ and have used a $\sigma = 1$. The results presented in Tables XIII–XVI clearly illustrate the consistency in the behavior of the algorithm for different civilization sizes.

Finally, in order to illustrate the convergence of our algorithm, we have presented the Box plots for the individuals in the initial and the final civilization for the welded beam design example in Fig. 5(a) and (b). The convergence of our algorithm is clearly visible.

V. SUMMARY AND CONCLUSION

In this paper, we have introduced an optimization algorithm that is based on a society and civilization model. Unlike evolutionary algorithms, where only the better performing individuals are allowed mate to generate children, the proposed algorithm improves the performance of all individuals in every society, either through an intra or intersociety information exchange. The algorithm also maintains parametrically unique solutions across the civilization, which at the end corresponds to a set of elite near optimal individuals. The process of maintaining

TABLE XIII
EFFECT OF CIVILIZATION SIZE: WELDED BEAM DESIGN

Civilization Size	Best	Worst	Average	Median	Approx Number of Evaluations
20 N	2.38465793	5.01141863	2.96069597	2.65888509	64,862
10 N	2.38543475	6.39967857	3.25513714	3.00258830	33,095
5 N	2.44335748	7.73081247	3.89071416	3.54875116	14,600

TABLE XIV
EFFECT OF CIVILIZATION SIZE: SPRING DESIGN

Civilization Size	Best	Worst	Average	Median	Approx Number of Evaluations
20 N	0.01266528	0.01323172	0.01271600	0.01268204	50,119
10 N	0.01266526	0.01671727	0.01292267	0.01292267	25,167
5 N	0.01266529	0.01628065	0.01336319	0.01309829	10,933

TABLE XV
EFFECT OF CIVILIZATION SIZE: SPEED REDUCER DESIGN

Civilization Size	Best	Worst	Average	Median	Approx Number of Evaluations
20 N	2994.48480183	3007.16274952	2998.02684932	2998.07420919	110,235
10 N	2994.74424174	3009.96473694	3001.75826452	3001.75826452	54,456
5 N	2994.50081312	3011.02958318	3001.58481659	3001.14802279	26,571

TABLE XVI
EFFECT OF CIVILIZATION SIZE: THREE-BAR TRUSS DESIGN

Civilization Size	Best	Worst	Average	Median	Approx Number of Evaluations
20 N	263.89584342	263.92889313	263.89888794	263.89665894	36,113
10 N	263.89584654	263.96975638	263.90335672	263.89893782	17,610
5 N	263.89586101	264.41247940	263.94298710	263.90930191	8,045

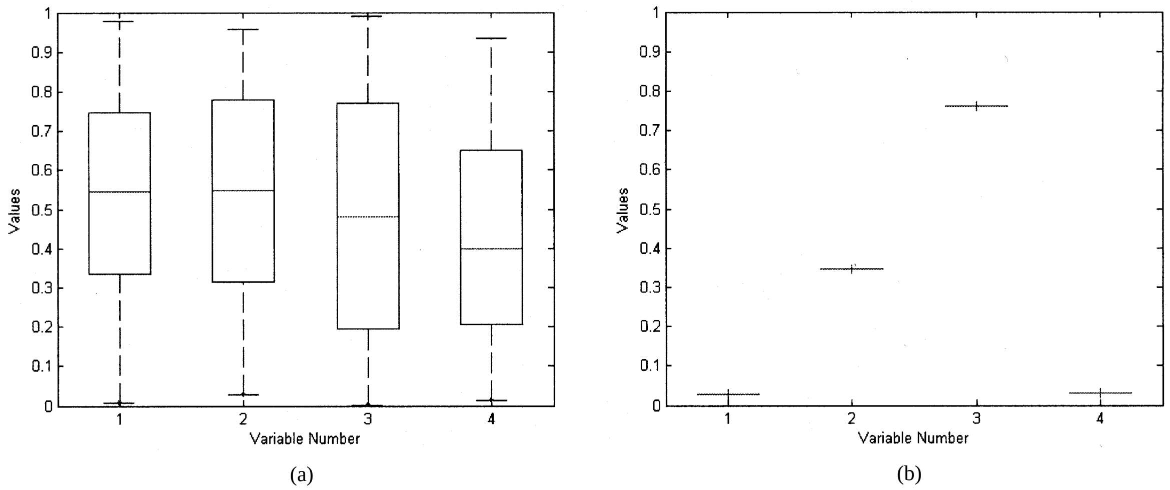

Fig. 5. (a) Box plot of the initial civilization: welded beam design example. (b) Box plot of the final civilization: welded beam design example. *Variables are scaled between 0 and 1.

unique individuals requires additional computations but it provides a room for diverse solutions to exist and evolve. The use of Pareto ranking to deal with constraints on one hand reduces the number of additional inputs required for constraint modeling while on the other increases the computational time involved in nondominated sorting. Although clustering, Pareto ranking, and maintaining unique individuals are computationally intensive operations, it is meaningful for problems where the computation of objectives or constraints are itself computationally expensive and best use of all such information should be made to keep the number of objective or constraint function evaluations to a minimum. In this algorithm, we have used a leader centric operator that results in search around the leader unlike some mean centric operators that explore regions around the mean. We have illustrated the behavior of the algorithm using the three-bar truss design example. One can observe that the initial civilization is fairly distributed over the entire design space while the final civilization has all its individuals concentrated around a particular region of the design space. The proposed algorithm is based on learning from the good instances. Currently, we are working on improvements that are based on learning from both the good and bad to enhance the algorithms performance.

The proposed algorithm performed consistently well on all the design examples. For welded beam design example, the best result reported by our algorithm is within 0.178% of the best-known result. For all other examples, our solutions are better than previously published results for the problems. The algorithm exhibited a very fast convergence in all the examples. This feature is particularly attractive, as it would mean achieving comparable solutions with fewer function evaluations. The results are promising as it indicates that the algorithm is capable of arriving at comparable solutions using significantly fewer function evaluations when compared with existing optimization algorithms.

APPENDIX
ENGINEERING DESIGN EXAMPLES

Welded Beam Design

Minimize

$$f(\mathbf{x}) = 1.104\,71 x_1^2 x_2 + 0.048\,11 x_3 x_4 (14.0 + x_2) \quad (6)$$

subject to

$$\tau(\mathbf{x}) - \tau_{\max} \leq 0 \quad (7)$$
$$\sigma(\mathbf{x}) - \sigma_{\max} \leq 0 \quad (8)$$
$$x_1 - x_4 \leq 0 \quad (9)$$
$$\delta(\mathbf{x}) - \delta_{\max} \leq 0 \quad (10)$$
$$P - P_C(x) \leq 0. \quad (11)$$

The other parameters are defined as follows:

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

where $P = 6000$ lb., $L = 14$, $\delta_{\max} = 0.25$ in, $E = 30 \times 10^6$ psi, $G = 12 \times 10^6$ psi, $\tau_{\max} = 13\,600$ psi, $\sigma_{\max} = 30\,000$ psi, $0.125 \leq x_1 \leq 10.0$, $0.1 \leq x_2 \leq 10.0$, $0.1 \leq x_3 \leq 10$, and $0.1 \leq x_4 \leq 10.0$.

Spring Design

Minimize

$$f(\mathbf{x}) = (x_3 + 2) x_1 x_2^2 \quad (12)$$

subject to

$$1 - \frac{x_1^3 x_3}{71\,785 x_2^4} \leq 0 \quad (13)$$
$$\frac{4x_1^2 - x_1 x_2}{12\,566(x_1 x_2^3 - x_2^4)} + \frac{1}{5108 x_2^2} - 1 \leq 0 \quad (14)$$
$$1 - \frac{140.45 x_2}{x_1^2 x_3} \leq 0 \quad (15)$$
$$\frac{x_1 + x_2}{1.5} - 1 \leq 0 \quad (16)$$

$0.25 \leq x_1 \leq 1.3$, $0.05 \leq x_2 \leq 2.0$, and $2 \leq x_3 \leq 15$.

Speed Reducer Design

Minimize

$$\begin{aligned} f(\mathbf{x}) = {} & 0.7854 x_1 x_2^2 (3.3333 x_3^2 + 14.9334 x_3 - 43.0934) \\ & - 1.508 x_1 (x_6^2 + x_7^2) + 7.4777 (x_6^3 + x_7^3) \\ & + 0.7854 (x_4 x_6^2 + x_5 x_7^2) \end{aligned} \quad (17)$$

subject to

$$\frac{27}{x_1 x_2^2 x_3} - 1 \leq 0 \quad (18)$$
$$\frac{397.5}{x_1 x_2^2 x_3^2} - 1 \leq 0 \quad (19)$$
$$\frac{1.93 x_4^3}{x_2 x_3 x_6^4} - 1 \leq 0 \quad (20)$$
$$\frac{1.93 x_5^3}{x_2 x_3 x_7^4} - 1 \leq 0 \quad (21)$$
$$\frac{\left[\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 16.9 \times 10^6\right]^{1/2}}{110.0 x_6^3} - 1 \leq 0 \quad (22)$$
$$\frac{\left[\left(\frac{745 x_5}{x_2 x_3}\right)^2 + 157.5 \times 10^6\right]^{1/2}}{85.0 x_7^3} - 1 \leq 0 \quad (23)$$
$$\frac{x_2 x_3}{40} - 1 \leq 0 \quad (24)$$
$$\frac{5 x_2}{x_1} - 1 \leq 0 \quad (25)$$
$$\frac{x_1}{12 x_2} - 1 \leq 0 \quad (26)$$
$$\frac{1.5 x_6 + 1.9}{x_4} - 1 \leq 0 \quad (27)$$
$$\frac{1.1 x_7 + 1.9}{x_5} - 1 \leq 0 \quad (28)$$

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

Three-Bar Truss Design

Minimize

$$f(\mathbf{x}) = (2\sqrt{2} x_1 + x_2) \times l \quad (29)$$

subject to

$$\frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0 \tag{30}$$

$$\frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0 \tag{31}$$

$$\frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \leq 0 \tag{32}$$

where $0 \leq x_1 \leq 1$ and $0 \leq x_2 \leq 1$; $l = 100$ cm, $P = 2$ KN/cm^2, and $\sigma = 2$ KN/cm^2

Acknowledgment

The authors would like to thank the anonymous referees, Editor-in-Chief, and Associate Editors for their constructive comments and suggestions that have helped a lot to come up with this improved form of the paper.

References

[1] T. Back, D. B. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*. Oxford, U.K.: Oxford Univ. Press, 1997.
[2] X. Jin and R. G. Reynolds, "Using knowledge-based evolutionary computation to solve nonlinear constrained optimization problems: A cultural algorithm approach," in *Proc. Congress Evolutionary Computation (CEC 1999)*, vol. 3, 1999, pp. 1672–1678.
[3] R. G. Reynolds and C. J. Chung, "A cultural algorithm framework to evolve multiagent cooperation with evolutionary programming," in *Proc. Evolutionary Programming VI*, 1997, pp. 323–333.
[4] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Systems, Man, Cybernetics—Part B*, vol. 26, pp. 29–41, Feb. 1996.
[5] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. 1995 IEEE Int. Conf. Neural Networks IV*, 1995, pp. 1942–1948.
[6] J. Kennedy, "Stereotyping: Improving particle swarm performance with cluster analysis," in *Proc. Congress Evolutionary Computation (CEC 2000)*, vol. 2, 2000, pp. 1507–1511.
[7] H. M. Voigt, J. Born, and I. Santibanez-Korek, "Modeling and simulation of distributed evolutionary search processes for function optimization," in *Proc. 1st Conf. Parallel Problem Solving from Nature*, LNCS 496, 1990, pp. 373–380.
[8] T. Starkweather, D. Whitley, and K. Mathias, "Optimization using distributed genetic algorithms," in *Proc. 1st Conf. Parallel Problem Solving from Nature*, LNCS 496, 1990, pp. 176–186.
[9] R. K. Ursem, "Multinational evolutionary algorithms," in *Proc. Congress Evolutionary Computation (CEC-99)*, 1999, pp. 1633–1640.
[10] M. Sebag and M. Schoenauer, "Controlling crossover through inductive learning," in *Proc. 3rd Conf. Parallel Problem Solving from Nature*, LNCS 866, 1994, pp. 209–218.
[11] M. Sebag and M. Schoenauer, "Toward civilized evolution: Developing inhibitions," in *Proc. Int. Conf. Genetic Algorithms*, 1997, pp. 291–298.
[12] M. Sebag and M. Schoenauer, "A society of hill-climbers," in *Proc. 4th IEEE Int. Conf. Evolutionary Computation, ICEC'97*, 1997, pp. 319–324.
[13] C. Ravise and M. Sebag, "An advanced civilization should not repeat its past errors," in *Proc. 13th Int. Conf. Machine Learning*, 1996, pp. 400–408.
[14] R. Reynolds, "An introduction to cultural algorithms," in *Proc. 3rd Annual Conf. Evolutionary Programming*, 1994, pp. 131–139.
[15] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, pp. 115–148, 1995.
[16] I. Ono and S. Kobayashi, "A real coded genetic algorithm for function optimization using unimodal normal distribution crossover," in *Proc. 7th Int. Conf. Genetic Algorithms*, 1997, pp. 246–253.
[17] K. Deb, D. Joshi, and A. Anand, "Real coded evolutionary algorithms with parent centric recombination," KanGAL, Rep. 2 001 003, 2001.
[18] R. Storn and K. Price, "Minimizing the real functions of the ICEC'96 contest by differential evolution," in *Proc. Int. Conf. Evolutionary Computation*, 1996, pp. 842–844.
[19] M. Laumanns, E. Zitzler, and L. Thiele, "On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization," in *Proc. Evolutionary Multiobjective Optimization(EMO 2001)*, 2001, pp. 181–196.
[20] Z. Michalewicz, "A survey of constraint handling techniques in evolutionary computation methods," in *Proc. 4th Annual Conf. Evolutionary Programming*, 1995, pp. 135–155.
[21] K. Deb, "Optimal design of a welded beam via genetic algorithms," *AIAA Journal*, vol. 29, no. 8, pp. 2013–2015, 1991.
[22] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, pp. 311–338, 2000.
[23] K. M. Ragsdell and D. T. Phillips, "Optimal design of a class of welded structures using geometric programming," *ASME J. Eng. Ind.*, ser. B, vol. 98, no. 2, pp. 1021–1025, 1976.
[24] J. N. Siddall, *Analytical Decision-Making in Engineering Design*. Englewood Cliffs, NJ: Prentice-Hall, 1972.
[25] A. D. Belegundu, "A study of mathematical programming methods for structural optimization," Dept. Civil Environ. Eng., Univ. Iowa, 1982.
[26] J. S. Arora, *Introduction to Optimum Design*. New York: McGraw-Hill, 1989.
[27] C. A. C. Coello, "Self-adaptive penalties for GA based optimization," in *Proc. Congress Evolutionary Computation (CEC 1999)*, vol. 1, 1999, pp. 573–580.
[28] T. Ray and P. Saini, "Engineering design optimization using a swarm with an intelligent information sharing among individuals," *Eng. Opt.*, vol. 33, no. 3, pp. 735–748, 2001.
[29] S. S. Rao, *Engineering Optimization*, 3rd ed. New York: Wiley, 1996.
[30] H. L. Li and P. Papalambros, "A production system for use of global optimization knowledge," *ASME J. Mech. Transm. Autom. Des.*, vol. 107, pp. 277–284, 1985.
[31] J. K. Kuang, S. S. Rao, and L. Chen, "Taguchi-aided search method for design optimization of engineering systems," *Eng. Opt.*, vol. 30, pp. 1–23, 1998.
[32] S. Azarm and W. C. Li, "Multi-Level design optimization using global monotonicity analysis," *ASME J. Mech. Transm. Autom. Des.*, vol. 111, pp. 259–263, 1989.
[33] T. Ray, "Golinski's speed reducer problem revisited," *AIAA J. Technical Note*, vol. 41, no. 3, pp. 556–558, 2003.
[34] S. Hernendez, "Multiobjective structural optimization," in *Geometry and Optimization Techniques for Structural Design*, S. Kodiyalam and M. Saxena, Eds. Amsterdam, The Netherlands: Elsevier, 1994, pp. 341–362.
[35] G. V. Reklaitis, S. Ravindran, and K. M. Ragsdell, *Engineering Optimization Methods and Applications*. New York: Wiley, 1983.

Tapabrata Ray received the B.Tech. (honors), M.Tech., and Ph.D. degrees from the Indian Institute of Technology, Kharagpur, India.

He is currently a Senior Research Scientist with Temasek Laboratories, National University of Singapore, Singapore.

K. M. Liew joined the Nanyang Technological University, Singapore, as Lecturer in 1991, and became a Tenured Full Professor in 2001 at the School of Mechanical and Production Engineering. He is the Founding Director of both the Centre for Advanced Numerical Engineering Simulations (CANES) and Nanyang Centre for Supercomputing and Visualisation (NCSV), Singapore.

Dr. Liew is a Fellow of American Society of Mechanical Engineers.