Cultural Algorithm for Engineering Design Problems

Xuesong Yan[1,2], Wei Li[1], Wei Chen[1], Wenjing Luo[1], Can Zhang[1] and Hanmin Liu[1,3]

**[1] School of Computer Science, China University of Geosciences
Wuhan, Hubei 430074, China**

**[2] Department of Computer Science, University of Central Arkansas
Conway, AR 72035, USA**

**[3] Wuhan Institute of Ship Building Technology
Wuhan, Hubei 430050, China**

Abstract

Many engineering optimization problems can be state as function optimization with constrained, intelligence optimization algorithm can solve these problems well. Cultural Algorithms are a class of computational models derived from observing the cultural evolution process in nature, cultural algorithms in the optimization of the complex constrained functions of its superior performance. Experiment results reveal that the proposed algorithm can find better solutions when compared to other heuristic methods and is a powerful optimization algorithm for engineering design problems.

Keywords: *Engineering Design Problems, Cultural Algorithms, Constrained Optimization, Population.*

1. Introduction

Evolutionary computation has found a wide range of applications in various fields of science and engineering. Among others, evolutionary algorithms (EA) have been proved to be powerful global optimizers. Generally, evolutionary algorithms outperform conventional optimization algorithms for problems which are discontinuous, non-differential, multi-modal, noisy and not well-defined problems, such as art design, music composition and experimental designs [1]. Besides, evolutionary algorithms are also well suitable for multi-criteria problems.

Many engineering optimization design problems can be formulated as constrained optimization problems. The presence of constraints may significantly affect the optimization performances of any optimization algorithms for unconstrained problems. With the increase of the research and applications based on evolutionary computation techniques [2], constraint handling used in evolutionary computation techniques has been a hot topic in both academic and engineering fields [3,4]. A general constrained optimization problem may be written as follows:

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

Subject to:

$$\begin{aligned} g_i(x) &= c_i, i = 1, 2, ..., n, \\ h_j(x) &\le d_j, j = 1, 2, ..., m. \end{aligned} \tag{2}$$

Where x is a vector residing in an n-dimensional space, $f(x)$ is a scalar valued objective function, $g_i(x) = c_i, i = 1, 2, ..., n,$ and $h_j(x) \le d_j, j = 1, 2, ..., m,$ are constraint functions that need to be satisfied.

Cultural Algorithms (CA) proposed by Reynolds in 1994 [5]. Cultural algorithm is in-depth analysis of the superiority of the original evolution theory on the basis of drawing on the social (cultural) evolution theory in the social sciences and has achieved broad consensus on the research results, and proposed a new algorithm. Cultural algorithm is used to solve complex calculations of the new global optimization search algorithms, cultural algorithms in the optimization of the complex constrained functions of its superior performance.

2. Cultural Algorithm

Evolutionary computation (EC) [6,7] methods have been successful in solving many diverse problem in search and optimization due to the unbiased nature of their operations which can still perform well in situation with little or no domain knowledge. However, there can be considerable improvement in their performance when problem specific knowledge is used to bias the problem solving process in order to identify patterns in their performance environment. These patterns are used to promote more instances of desirable candidates or to reduce the number of less desirable candidates in the population. In either case, this can afford the system an opportunity to reach the desired solution more quickly.

Adaptive evolutionary computation takes place when an EC system is able to incorporate such information into its representation and operators in order to facilitate the pruning and promoting activities mentioned above. Some research works have shown that self-adaptation can take place on several levels within a system such as the population level, the individual level, and the component level. At the population level, aspects of the system parameters that control all elements of the population can be modified. At the individual level, aspects of the system that control the action of specific individual can be modified. If the individual is specified as s collection of components then component level adaptation is possible. This involves the adaptation of parameters that control the operation of one or more components that make up an individual.

In human societies, culture can be a vehicle for the storage of information in a form that is independent of the individual or individuals that generated and are potentially accessible to all members of the society. As such culture is useful in guiding the problem solving activities and social interaction of individuals in the population. This allows self-adaptive information as well as other knowledge to be stored and manipulated separately from the individuals in the social population. This provides a systematic way of utilizing self-adaptive knowledge to direct the evolution of a social population. Thus, cultural systems are viewed as a dual inheritance system where, at each time step, knowledge at both the population level and the level of acquired beliefs is transmitted to the next generation. This acquired knowledge is viewed to act as beacons by which to guide individuals towards perceived good solutions to problems and away from less desirable ones at a given time step. Cultural Algorithms in order to model the evolution of cultural systems based upon principles of human social evolution taken from the social science literature.

Cultural Algorithms are a class of computational models derived from observing the cultural evolution process in nature [5, 8, 9]. In this algorithm, individuals are first evaluated using a performance function. The performance information represents the problem-solving experience of an individual. An acceptance function determines which individuals in the current population are able to impact, or to be voted to contribute, to the current beliefs. The experience of these selected individual is used to adjust the current group beliefs. These group beliefs are then used to guide and influence the evolution of the population at the next step, where parameters for self-adaptation can be determined from the belief space. Information that is stored in the belief space can pertain to any of the lower levels, e.g. population, individual, or component. As a result, the belief space can be used to control self-adaptation at any or all of these levels. The cultural algorithm is a dual inheritance system with evolution taking place at the population level and at the belief level. The two components interact through a communications protocol. The protocol determines the set of acceptable individuals that are able to update the belief space. Likewise the protocol determines how the updated beliefs are able to impact and influence the adaptation of the population component.

The Cultural Algorithm is a dual inheritance system that characterizes evolution in human culture at both the macro-evolutionary level, which takes place within the belief space, and at the micro-evolutionary level, which occurs at the population space. CA consists of a social population and a belief space. Experience of individuals selected from the population space by the acceptance function is used to generate problem solving knowledge that resides in the belief space. The belief space stores and manipulates the knowledge acquired from the experience of individuals in the population space. This knowledge can control the evolution of the population component by means of the influence function. As a result, CA can provide an explicit mechanism for global knowledge and a useful framework within which to model self-adaptation in an EC system. The population level component of the cultural algorithm will be Evolutionary Programming (EP). The global knowledge that has been learned by the population will be expressed in terms of both normative and situational knowledge as discussed earlier.

A flow-chart of the Cultural Algorithms is shown in Fig.1.

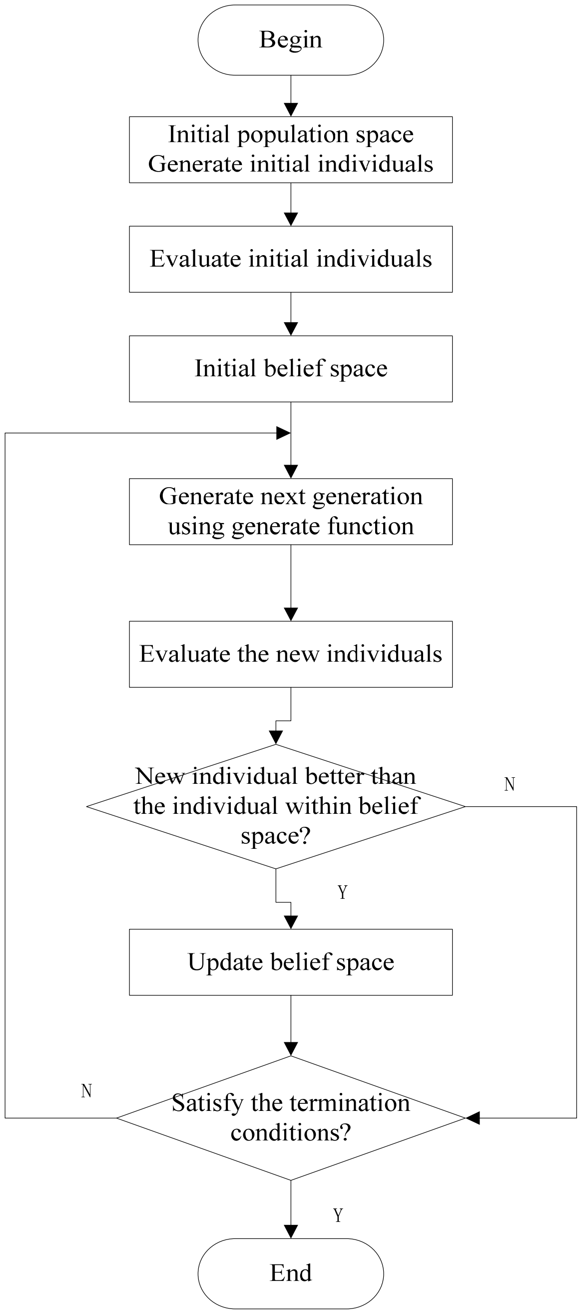

Fig. 1 Flow-hart of cultural algorithms

In this algorithm, first the belief space and the population space are initialized. Then, the algorithm will repeat processing for each generation until a termination condition is achieved. Individuals are evaluated using the performance function. The two levels of Cultural Algorithm communicate through the acceptance function and the influence function. The acceptance function determines which individuals from the current population are selected to impact the belief space. The selected individuals' experiences are generalized and applied to adjust the current beliefs in the belief space via the update function. The new beliefs can then be used to guide and influence the evolutionary process for the next generation.
Cultural algorithms as described above consist of three components. First, there is a population component that contains the social population to be evolved and the mechanisms for its evaluation, reproduction, and modification. Second there is a belief space that represents the bias that has been acquired by the population during its problem-solving process. The third component is the communications protocol that is used to determine the interaction between the population and their beliefs.

3. Engineering Design Algorithm based on CA

3.1 Individual Initialization

The traditional method of genetic algorithm is randomly initialized population, that is, generate a series of random numbers in the solution space of the question. Design the new algorithm, we using the orthogonal initialization in the initialization phase. For the general condition, before seeking out the optimal solution the location of the global optimal solution is impossible to know, for some high-dimensional and multi-mode functions to optimize, the function itself has a lot of poles, and the global optimum location of the function is unknown. If the initial population of chromosomes can be evenly distributed in the feasible solution space, the algorithm can evenly search in the solution space for the global optimum. Orthogonal initialization is to use the orthogonal table has the dispersion and uniformity comparable; the individual will be initialized uniformly dispersed into the search space, so the orthogonal design method can be used to generate uniformly distributed initial population.

3.2 Belief Space Structure

In this paper, we define the belief space as $< N[n], C[m] >$, in here N denotes the normative knowledge, consist of the change interval information of variables; and C is the belief-cells information consist of the constrained knowledge, m is the number of cells.

The normative knowledge N, a set of interval information for each of the n parameters is defined formally as 4-tuple: $N = \langle I_j, L_j, U_j, adjust_N \rangle, j = 1, 2, \ldots, n$, where I_j denotes the closed interval of variable j, that is a continuous set of real numbers x represented as a ordered number pair: $I_j = [l_j, u_j] = \{x \mid l_j \le x \le u_j, x \in R\}$. l_j (lower bound) and u_j (upper bound) are initialized by the give domain values. L_j represents the performance score of the lower bound l_j for parameter j. U_j represents the performance score of the upper bound u_j for parameter j.

The constrained information $C[i] =< Class_i, Cnt1_i, Cnt2_i, W_i, Pos_i, Csize_i >$, in here $Class_i$ denotes the status of ith unit in belief space，as feasible or infeasible. $Cnt1_i$ and $Cnt2_i$ denotes the number of individual locates in feasible region or infeasible region, the initial value is 0. W_i denotes the weight of ith unit, in this paper the higher the fitness value of the unit, the weight value is smaller. Pos_i is vector denotesThe left-most position of the corner. $Csize_i$ denotes the size of the ith unit.

Update the normative knowledge N in belief space uses the Eq. (3):

$$l_i^{t+1} = \begin{cases} x_{j,i} & x_{j,i} \le l_i^t \text{ or } f(x_j) < L_i^t \\ l_i^t & \text{others} \end{cases}$$

$$L_i^{t+1} = \begin{cases} f(x_j) & x_{j,i} \le l_i^t \text{ or } f(x_j) < L_i^t \\ L_i^t & \text{others} \end{cases}$$

$$u_i^{t+1} = \begin{cases} x_{j,i} & x_{j,i} \ge u_i^t \text{ or } f(x_j) < U_i^t \\ u_i^t & \text{others} \end{cases}$$

$$U_i^{t+1} = \begin{cases} f(x_j) & x_{j,i} \ge u_i^t \text{ or } f(x_j) < U_i^t \\ U_i^t & \text{others} \end{cases}$$

(3)

Update the C in belief space uses the Eq. (4):

$$Class_i = \begin{cases} unknown \quad if Cnt1_i = 0 and Cnt2_i = 0 \\ feasible \quad if Cnt1_i > 0 and Cnt2_i = 0 \\ unfeasible \quad if Cnt1_i = 0 and Cnt2_i > 0 \\ semi\_feasible \quad if\, Cnt1_i > 0 and Cnt2_i > 0 \end{cases} \tag{4}$$

3.3 Influence Function

In this paper, the knowledge represented in the belief space can be explicitly used to influence the creation of the offspring via an influence function. For normative knowledge, the influence function shown as the Eq. (5):

$$x_{N+i,j} = \begin{cases} x_{N,j} + \left|(u_j - l_j) \times N(0,1)\right| & if\ x_{N,j} < l_j \\ x_{N,j} - \left|(u_j - l_j) \times N(0,1)\right| & if\ x_{N,j} > u_j \end{cases} \tag{5}$$

For constrained knowledge, the influence function shown as the Eq. (6):

$$x_{N,j} = \begin{cases} moveTo(choose(Cell[m])) & if\ x_{N,j} \in \{unfeasibleCells\} \\ x_{N,j} + (u_j - l_j) * N(0,1) / m_j & otherwise \end{cases} \tag{6}$$

In here, m_j is the number of cells for variable j, $moveTo()$ is move function, $choose(Cell[m])$ is selection function.

4. Simulation Experiment

In this section, we will carry out numerical simulation based on some well-known constrained engineering design problems to investigate the performances of the proposed algorithm. The selected problems have been well studied before as benchmarks by various approaches, which is useful to show the validity and effectiveness of the proposed algorithm. For each testing problem, the parameters of our algorithm are set as follows: the size of population is 100, the number of iteration is 1000 and the run time is 100.

4.1 Tension/Compression String Problem

This problem is described by Arora [10], Coello and Montes [11] and Belegundu [12]. It consists of minimizing the weight ($f(x)$) of a tension/compression string subject to constraints on shear stress, surge frequency and minimum deflection as shown in Fig. 2. The design variables are the mean coil diameter $D(=x_1)$; the wire diameter $d(=x_2)$ and the number of active coils $N(=x_3)$. The problem can be stated as:

Minimize:

$$f(x) = (x_3 + 2) x_2 x_1^2 \tag{7}$$

Subject to:

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0,$$

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \le 0,$$

$$g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0,$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \le 0.$$

(8)

This problem has been solved by Belegundu [12] using eight different mathematical optimization techniques, Arora [10] also solved this problem using a numerical optimization technique called constraint correction at constant cost, Additionally, Coello solved this problem using GA-based method [13] and a feasibility-based tournament selection scheme [11]. Table 1 presents the best solution of this problem obtained using the CA and compares the CA results with solutions reported by other researchers. It is obvious from the Table 1 that the result obtained using CA is better than those reported previously in the literature.

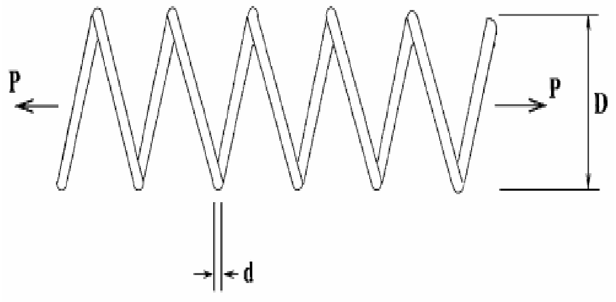

Fig. 2 Tension/compression string problem

4.2 Pressure Vessel Problem

A cylindrical vessel is capped at both ends by hemispherical heads as shown in Fig. 3. The objective is to minimize the total cost, including the cost of material, forming and welding. There are four design variables: T_s (thickness of the shell, x_1), T_h (thickness of the head, x_2), R (inner radius, x_3) and L (length of cylindrical section of the vessel, not including the head, x_4). T_s and T_h are integer multiples of 0.0625 inch, witch are the available thickness of rolled steel plates, and R and L are continuous.

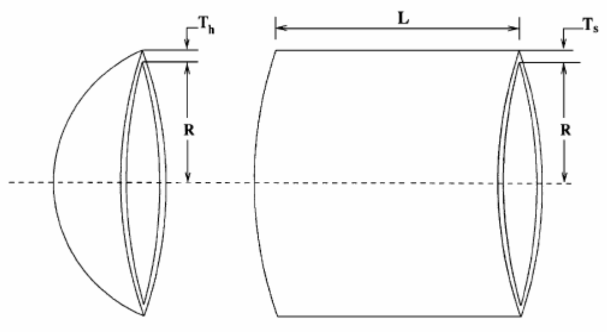

Fig. 3 Pressure vessel problem

Using the same notation given by Coello [14], the problem can be stated as follows:

Minimize:

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (9)$$

Subject to:

$$\begin{aligned} &g_1(x) = -x_1 + 0.0193x_3 \le 0, \\ &g_2(x) = -x_2 + 0.00954x_3 \le 0, \\ &g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1{,}296{,}000 \le 0, \\ &g_4(x) = x_4 - 240 \le 0. \end{aligned} \quad (10)$$

This problem has been solved before by Sandgren using a branch and bound technique [15], by Kannan and Kramer using an augmented Lagrangian Multiplier approach [16], by Deb and Gene using Genetic Adaptive Search [17], by Coello using GA-based co-evolution model [13] and a feasibility-based tournament selection scheme [11]. The comparisons of results are shown in Table 2. The results obtained using the CA, were better optimized than any other earlier solutions reported in the literature.

4.3 Welded Beam Problem

The welded beam structure, shown in Fig. 4, is a practical design problem that has been often used as a benchmark for testing different optimization methods. The objective is to find the minimum fabricating cost of the welded beam subject to constraints on shear stress (τ), bending stress (σ), buckling load (P_c), end deflection (δ), and side constraint. There are four design variables: $h(=x_1)$; $l(=x_2)$; $t(=x_3)$ and $b(=x_4)$.

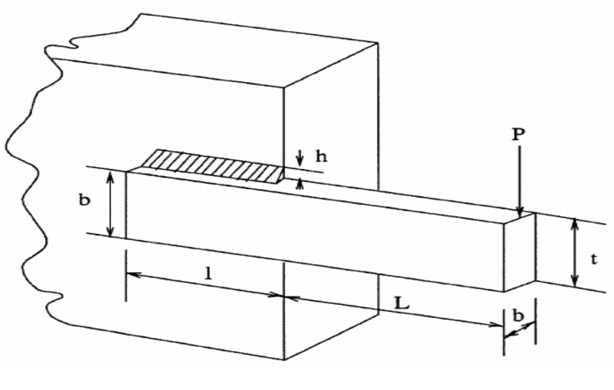

Fig. 4 Welded beam problem

The mathematical formulation of the objective function $f(x)$, which is the total fabricating cost mainly comprised of the set-up, welding labor, and material costs, is as follows:

Minimize:

$$f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \quad (12)$$

Subject to:

$$g_1(x) = \tau(x) - 13000 \le 0,$$
$$g_2(x) = \sigma(x) - 30000 \le 0,$$
$$g_3(x) = x_1 - x_4 \le 0,$$
$$g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0, \quad (13)$$
$$g_5(x) = 0.125 - x_1 \le 0,$$
$$g_6(x) = \delta(x) - 0.25 \le 0,$$
$$g_7(x) = 6000 - P_c(x) \le 0,$$

Where:

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},$$
$$\tau' = \frac{6000}{\sqrt{2}x_1x_2},$$
$$\tau'' = \frac{MR}{J},$$
$$M = 6000(14 + \frac{x_2}{2}),$$
$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2}, \quad (14)$$
$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + (\frac{x_1 + x_3}{2})^2\right]\right\},$$
$$\sigma(x) = \frac{504000}{x_4x_3^2},$$
$$\delta(x) = \frac{2.1952}{x_3^4x_4},$$
$$P_c(x) = 64746.022(1 - 0.0282346x_3)x_3x_4^3.$$

The approaches applied to this problem include geometric programming [18], genetic algorithm with binary representation and traditional penalty function [19], a GA-based co-evolution model [13] and a feasibility-based tournament selection scheme inspired by the multi-objective optimization techniques [11]. The comparisons of results are shown in Table 3. The results obtained using the CA, were better optimized than any other earlier solutions reported in the literature.

4.4 Satellite Costs Problem

The objective is to find the minimum cost of the satellite design, subject to constraints on Covering bandwidth S_{w0}, Flexible transfer time Δt_0, Resolution d_{s0}, Eclipse factor $k_{e,\max}$, The total mass of the satellite M_{total}, Battery cycle life $N_{BA,\max}$ and Satellite volume $V_{sat,0}$. There are six design variables: Orbital altitude h, Angle of inclination of the orbit i, local time of the Descending node DNT, Transfer orbit apogee radius r_a, CCD camera focal length f_c, The side length of the satellite structure b and height l. This problem can be stated as follows [20]:

$$\min C_{total} \quad (15)$$

Subject to:

$$g_1: S_w = \frac{2R_e}{\sin i}\{\sin^{-1}\left[\frac{14336}{\sqrt{49N_p{}^2 + f_c{}^2}}\cdot\frac{h + R_e}{R_e}\right]\} -$$
$$\sin^{-1}(\frac{7N_P}{\sqrt{49N_p{}^2 + f_c{}^2}}) \ge 50$$
$$g_2: h / f_c \le 15/7$$
$$g_3: \Delta t = \sqrt{\frac{(r_a + r_1)^3}{8\mu}}\left\{\begin{array}{l}\arccos[\frac{(r_a - r_1) + (r_a + r_1)\cos\theta}{(r_a + r_1) + (r_a - r_1)\cos\theta}] - \\ \frac{4r_ar_1}{(r_a + r_1) + (r_a - r_1)\cos\theta}\end{array}\right\} \quad (16)$$
$$g_4: \frac{1}{2} - \frac{1}{\pi}\sin^{-1}(\frac{\sqrt{2R_eh + h^2}}{R_e + h}) < 0.35$$
$$g_{5:}: M_{total} \le 500$$
$$g_6: 2\pi\sqrt{\frac{(r_a + r_1)^3}{\mu}} \ge 5256s$$
$$g_7: l^2b \ge 0.5\times 10^9$$

In here, $R_e = 6378.1km$ and $\mu = 3.986006\times 10^5 km/s$.

We use our algorithm for this problem and Table 4 is the result of our algorithm for the design variable and constrained variables.

Table 4: Experiment result

Design Variables	***Lower Bound***	***Upper Bound***	***Our Algorithm***
h (km)	500	1200	574.091
DNT (hr)	8	12	8.4
r_a (km)	550	5000	743.181
f_c (mm)	200	1000	267.909
l (mm)	500	1500	1283.70
b (mm)	500	1500	823.61
S_{w0} (km)	50		51.45
Δt_0 (s)		1200	1123.886
d_{s0}		30	30
$k_{e,\max}$		0.35	0.264
M_{total} (kg)		500	343.94
$N_{BA,\max}$ (s)		6000	5475
$V_{sat,0}$ (m3)	0.5		1.2372
C_{total}			382150.62

5. Conclusions

This paper introduces a new method-cultural algorithm for solving the engineering design problem. For the empirical studies this algorithm has proved to be efficient, and the experiments results shown the new method are effective for engineering optimization design.

Acknowledgments

This paper is supported by the Provincial Natural Science Foundation of Hubei (No. 2011CDB334 and 2011CDB346), the Fundamental Research Founds for National University, China University of Geosciences (Wuhan) and National Civil Aerospace Pre-research Project of China.

References

[1] H.-P. Schwefel, "Evolution and Optimum Seeking", Wiley, 1995.
[2] Qie He and Ling Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems", Engineering Applications of Artificial Intelligence, 20, 2007, pp.89-99.
[3] Coello, C.A.C., Montes, E.M., "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection", Advanced Engineering Informatics, 16, 2002, pp.193–203.
[4] Michalewicz, Z. (Eds.), "Evolutionary Algorithms in Engineering Applications", Springer, Berlin, 1995, pp. 497–514.
[5] R. Reynoids, "An introduction to cultural algorithms", in Proceedings of the 3rd Annual Conference on Evolutionary Programming, Sebald, AX; Fogel, L.J. (Editors), River Edge, NJ, World Scientific Publishing, 1994, pp.13 1-139.
[6] Goldbeg, D. E, "Genetic Algorithms in Search, Optimization and Machine Learning", Reading, Mass.: Addison-Wesley, 1989.
[7] Michalewicz, Z, "Genetic Algorithms + Data Structures = Evolution Programs", 3rd Ed. Berlin: Springer – Verlag, 1996.
[8] CHUNG C, "Knowledge-based approaches to self-adaptation in cultural algorithms", Ph.D. Thesis, Wayne State University, Detroit, Michigan, USA, 1997.
[9] ZHANG Yin, "Cultural algorithm and its application in the portfolio", Master Thesis, Harbin University of Science and Technology, Harbin, China, 2008. (in Chinese)
[10] Arora, J.S., "Introduction to Optimum Design", New York, McGraw-Hill, 1989.
[11] Coello, C.A.C., Montes, E.M., "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, Advanced Engineering Informatics", 2002, 16, pp.193–203.
[12] Belegundu, A.D., "A study of mathematical programming methods for structural optimization", Department of Civil and Environmental Engineering, University of Iowa, Iowa City, Iowa, 1982.
[13] Coello, C.A.C., "Use of a self-adaptive penalty approach for engineering optimization problems", Computers in Industry, 2000, 41, pp.113–127.
[14] Coello, C.A.C., "Theoretical and numerical constraint handling techniques used with evolutionary algorithms: a survey of the state of the art", Computer Methods in Applied Mechanics and Engineering, 2002, 191 (11/12), pp.1245–1287.
[15] Sandgren, E., "Nonlinear integer and discrete programming in mechanical design", In: Proceedings of the ASME Design Technology Conference, Kissimine, FL, 1988, pp. 95–105.
[16] Kannan, B.K., Kramer, S.N., "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design", Transactions of the ASME, Journal of Mechanical Design, 116, 1994, pp.318–320.
[17] Deb, K., "GeneAS: a robust optimal design technique for mechanical component design", Evolutionary Algorithms in Engineering Applications, 1997, pp.497–514.
[18] Ragsdell, K.M., Phillips, D.T., "Optimal design of a class of welded structures using geometric programming", ASME Journal of Engineering for Industries, 98 (3), 1976, pp.1021–1025.
[19] Deb, K., Optimal design of a welded beam via genetic algorithms", AIAA Journal, 29 (11), 1991, pp.2013–2015.
[20] Xiji Wang, Dayao Li, "Satellite design methodology", Beijing: Atomic Energy Press, 2005.
[21] Xuesong Yan, Qinghua Wu, et.al, "An Efficient Function Optimization Algorithm based on Culture Evolution", International Journal of Computer Science Issues, Vol. 9, No. 2, 2012, pp. 11-18.
[22] X.S. Yan, Q.H. Wu, "Function Optimization Based on Cultural Algorithms", Journal of Computer and Information Technology, Vol.2, 2012, pp. 152-158.
[23] X.S. Yan et, al, "Orthogonal evolutionary algorithm and its application in relay volume optimization design", Computer Engineering and Applications, 47(18), 2011, pp.215-217. (in Chinese).
[24] X.S. YAN, Q.H. WU, C.Y. HU, Q.Z. LIANG, "Electronic Circuits Optimization Design Based On Cultural Algorithms", International Journal of Information Processing and Management, 2(1), 2011, pp. 49-56.
[25] X.S. Yan et.al, "Designing Electronic Circuits Using Cultural Algorithms", Proceedings of Third International Workshop on Advanced Computational Intelligence, 2010, pp. 299-303.
[26] X.S.Yan et.al, "Application of Cultural Algorithm in Stock Data Modeling", Computer Engineering & Science, Vol. 32, No.6, 2010, pp. 68-71.(in Chinese).

Xuesong Yan associate professor received him B.E. degree in Computer Science and Technology in 2000 and M.E. degree in Computer Application from China University of Geosciences in 2003, received he Ph.D. degree in Computer Software and Theory from Wuhan University in 2006. He is currently with School of Computer Science, China University of Geosciences, Wuhan, China and now as a visiting scholar with Department of Computer Science, University of Central Arkansas, Conway, USA. He research interests include evolutionary computation, data mining and computer application.

Wei Li received her B.E. degree in Computer Science and Technology in 2012. She is currently is the M.E. degree candidate with School of Computer Science, China University of Geosciences, Wuhan, China. Her research interests include evolutionary computation.

Can Zhang received him B.E. degree in Computer Science and Technology in 2011. He is currently is the M.E. degree candidate with School of Computer Science, China University of Geosciences, Wuhan, China. Her research interests include evolutionary computation.

Wei Chen received him B.E. degree in Computer Science and Technology in 2012. He is currently is the M.E. degree candidate with School of Computer Science, China University of Geosciences, Wuhan, China. Her research interests include evolutionary computation.

Wenjing Luo received her B.E. degree in Computer Science and Technology in 2012. She is currently is the M.E. degree candidate with School of Computer Science, China University of Geosciences, Wuhan, China. Her research interests include evolutionary computation.

Hanmin Liu associate professor. He is currently as a Ph.D candidate of School of Computer Science, China University of Geosciences, Wuhan, China. He research interests include evolutionary computation and applications.

Table 1: Comparison of the best solution for tension/compression string problem

Design variables	***Belegundu (1982)***	***Arora (1989)***	***Coello (2000)***	***Coello (2002)***	***CA***
$x_1(d)$	0.050000	0.053396	0.051480	0.051989	0.051728
$x_2(D)$	0.315900	0.399180	0.351661	0.363965	0.357644
$x_3(N)$	14.250000	9.185400	11.632201	10.890522	11.244543
$g_1(x)$	-0.000014	0.000019	-0.002080	-0.000013	-0.000845
$g_2(x)$	-0.003782	-0.000018	-0.000110	-0.000021	-1.2600e-05
$g_3(x)$	-3.938302	-4.123832	-4.026318	-4.061338	-4.051300
$g_4(x)$	-0.756067	-0.698283	-4.026318	-0.722698	-0.727090
$f(x)$	0.0128334	0.0127303	0.0127048	0.0126810	0.0126747

Table 2: Comparison of the best solution for pressure vessel problem

Design variables	***Sandgren (1988)***	***Kannan (1994)***	***Deb (1997)***	***Coello (2000)***	***Coello (2002)***	***CA***
$x_1(T_s)$	1.125000	1.125000	0.937500	0.812500	0.812500	0.812500
$x_2(T_h)$	0.625000	0.625000	0.500000	0.437500	0.437500	0.437500
$x_3(R)$	47.700000	58.29100	48.329000	40.323900	42.097398	38.860100
$x_4(L)$	117.701000	43.690000	112.679000	200.000000	176.654050	221.365000
$g_1(x)$	-0.204390	0.000016	-0.004750	-0.034324	-0.000020	-0.000000
$g_2(x)$	-0.169942	-0.068904	-0.038941	-0.052847	-0.035891	-0.004300
$g_3(x)$	54.226012	-21.220104	-3652.876838	-27.105845	-27.886075	-0.000000
$g_4(x)$	-122.299000	-196.310000	-127.321000	-40.000000	-63.345953	-18.63500
$f(x)$	8129.1036	7198.0428	6410.3811	6288.7445	6059.9463	5850.3800

Table 3: Comparison of the best solution for welded beam problem

Design variables	***Ragsdell (1976)***	***Deb (1991)***	***Coello (2000)***	***Coello (2002)***	***CA***
$x_1(h)$	0.245500	0.248900	0.208800	0.205986	0.202369
$x_2(l)$	6.196000	6.173000	3.420500	3.471328	3.544214
$x_3(t)$	8.273000	8.178900	8.997500	9.020224	9.048210
$x_4(b)$	0.245500	0.253300	0.210000	0.206480	0.205723
$g_1(x)$	-5743.826517	-5758.603777	-0.337812	-0.074092	-12.839796
$g_2(x)$	-4.715097	-255.576901	-353.902604	-0.266227	-1.247467
$g_3(x)$	0.000000	-0.004400	-0.001200	-0.000495	-0.001498
$g_4(x)$	-3.020289	-2.982866	-3.411865	-3.430043	-3.429347
$g_5(x)$	-0.120500	-0.123900	-0.083800	-0.080986	-0.079381
$g_6(x)$	-0.234208	-0.234160	-0.235649	-0.235514	-0.235536
$g_7(x)$	-3604.275002	-4465.270928	-363.232384	-58.666440	-11.681355
$f(x)$	2.385937	2.433116	-1.748309	1.728226	1.728024