Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 534162, 8 pages
http://dx.doi.org/10.1155/2014/534162

Research Article

A Guaranteed Global Convergence Social Cognitive Optimizer

Jia-ze Sun,[1,2] Shu-yan Wang,[1] and Hao Chen[1]

[1] *School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an 710121, China*
[2] *School of Information Science and Technology, Northwest University, Xi'an 710127, China*

Correspondence should be addressed to Jia-ze Sun; sunjiaze@126.com

Received 11 July 2014; Accepted 10 September 2014; Published 29 September 2014

Academic Editor: Shifei Ding

From the analysis of the traditional social cognitive optimization (SCO) in theory, we see that traditional SCO is not guaranteed to converge to the global optimization solution with probability one. So an improved social cognitive optimizer is proposed, which is guaranteed to converge to the global optimization solution. The global convergence of the improved SCO algorithm is guaranteed by the strategy of periodic restart in use under the conditions of participating in comparison, which helps to avoid the premature convergence. Then we give the convergence proof for the improved SCO based on Solis and Wets' research results. Finally, simulation results on a set of benchmark problems show that the proposed algorithm has higher optimization efficiency, better global performance, and better stable optimization outcomes than the traditional SCO for nonlinear programming problems (NLPs).

1. Introduction

Swarm intelligence (SI) is the collective behavior of decentralized, self-organized systems, natural or artificial in artificial intelligence field. Swarms often are large groups of small insects in which each member performs a simple role, but the action produces complex behavior as a whole. The emergence of such complex behavior extends beyond swarms. Similar complex social structures also occur in higher-order animals and insects that do not swarm: colonies of ants, flocks of birds, packs of wolves, or colonies of bees, and so on.

Human has higher adaptability and social intelligence than insect swarm, and human intelligence derives from the interactions of individuals, including interacting with the environment, in a social world from the study of social cognitive theory. Xie et al. [1] present social cognitive optimization (SCO) algorithm which is a novel heuristic swarm intelligence optimization algorithms. SCO algorithm has a probabilistic iterative procedure of lots of learning agents. A learning agent obtains vicarious capability by tournament selection and shares information by the knowledge library with symbolizing capability. Because SCO algorithm fully makes use of the interactions and share of the entire social swarm, it greatly improves the convergence speed and accuracy of the swarm intelligence algorithm and makes it better than many other well-used intelligent optimization methods, such as PSO and ACO, in many applications. Such applications include nonlinear programming problems (NLPs), nonlinear complementarity's problem (NCP) [2], fractional programs [3], nonlinear system of equation [4], engineering design problems [5], and Web service composition selection [6].

Many researchers improved the traditional SCO. Wang et al. [7] improved SCO algorithm through joining self-organizing migrating algorithm (SOMA) migration in the process of SCO and adding two parameters in SCO algorithm to solve the SAT problem, which showed the improved SCO may obtain the quick convergence rate in the optimization early and only small effect on the last result. Ma et al. [8] brought in the chaos and Kent mapping function to modify and optimize the conditions of neighborhood search and got more reasonable knowledge points which were distributed more uniformly for solving the nonlinear constraint problems. Sun et al. [9] presented a hybrid social cognitive optimization algorithm based on elitist strategy and chaotic optimization is proposed to solve constrained nonlinear programming problems, which partitions learning agents into three groups in proportion: elite learning agents, chaotic learning agents, and common learning agents. Zhizhong et al. [10] improve the social cognitive optimization, put the improved SCO algorithm into the framework of culture algorithm, constructed a novel algorithm, culture social cognitive optimization (C-SCO), and used C-SCO to solve the QoS-aware cloud service composition problem. Sun et al. [11] present a social cognitive optimization algorithm (SCO) to generate optimal evidence weight values for the Dempster-Shafer (D-S) evidence model based on historical training data.

These improved algorithms, called hybrid optimization algorithms, are mainly based on empirical analysis of the experiment while the global convergence analysis of hybrid algorithm has not been studied in theory. Because the SCO is originated from simulation social cognitive progress and involves sophisticated stochastic behavior, it is hard to perform theoretical analysis, resulting in the lack of solid theoretical foundation. Particularly, the performance of SCO as optimization techniques requires theoretical support. The lack of theoretical foundation injures the further development of SCO and blocks the application of SCO in problems where serious algorithms are required.

In this paper, from the analysis of the traditional social cognitive optimization (SCO) in theory, we see that traditional SCO is not guaranteed to converge to the global optimization solution with probability one. So a novel social cognitive optimizer is called stochastic SCO that is guaranteed to converge to the global optimization solution. The global convergence of the improved SCO algorithm is guaranteed by the strategy of periodic restart in use under the conditions of participating in comparison, which helps to avoid the premature convergence. Then we give the convergence proof for the stochastic SCO based on Solis and Wets' research results [12]. Finally, simulation results on a set of benchmark problems show that the proposed algorithm has higher optimization efficiency, better global performance, and better stable optimization outcomes than the traditional SCO for NLPs.

The remainder of this paper is organized as follows. In Section 2, we survey the traditional SCO and analyse global convergence of traditional SCO algorithm. In Section 3, our improvement to traditional SCO is described concretely and the proof of the global convergence of the proposed algorithms is presented. In Section 4 the typical experiments are employed to evaluate the performance of the improved SCO and the conclusions are showed in Section 5, and finally the last section presents the acknowledgment and the appendix.

2. Convergence Analysis of Traditional SCO Algorithm

2.1. Social Cognitive Optimization Algorithm (SCO). Social cognitive theory (SCT) agrees that people learn by observing others, with the environment, behavior, and cognition all as the chief reciprocal factors in influencing development. Human learning possesses the abilities to symbolize, learn from others, plan alternative strategies, regulate one's own behavior, and engage in self-reflection. So human has higher adaptability and social intelligence than insect swarm. By introducing human social intelligence based on SCT to artificial system, Xie et al. [1] proposed social cognitive optimization (SCO) algorithm in 2002. In SCO optimization procedure, a knowledge library with symbolizing capability consists of a number of knowledge points which are denoted by the location x in search space S and its fitness values; learning agents, on behalf of human individuals, in possession of a knowledge point in the knowledge library, act observational learning via the neighborhood local searching by observing the selected model from tournament selection. The neighborhood local searching for x_2 referring to x_1 is finding a new point x', which is for d dimension

$$x'_d = U\left(x_{1,d}, x_{m,d}\right), \tag{1}$$

where $U(a,b)$ is a uniform distribution which is usually generated by linear congruential method within $[a,b]$, $x_{m,d} = 2 * x_{2,d} - x_{1,d}$. SCO algorithm basic steps are clearly described in [1].

2.2. Basic Conception and Theory for Global Convergence Theorem. The general global optimization problem (P) used here is defined as

$$\min_{x \in S} f(x), \tag{2}$$

where x is a vector of n decision variables, S is an n-dimensional feasible region and is assumed to be nonempty, a subset of R^n, and $f(x)$ is a real-valued function defined over S from R^n to R. The goal is to find a value for x contained in S that minimizes f. Notice that the feasible region may include both continuous and discrete variables. Denote the global optimal solution to (P) by (x^{**}, y^{**}), where

$$\begin{gathered} x^{**} = \arg\min_{x \in S} f(x), \\ y^{**} = f\left(x^{**}\right) = \min_{x \in S} f(x). \end{gathered} \tag{3}$$

Solis and Wets [12] provide a convergence proof, in probability, to the global minimum for general step size algorithms with conditions on the method of generating the step length and direction.

Conceptual algorithm [12] is as follows.

Step 0. Find x^0 in S and set $k = 0$.

Step 1. Generate ξ^k from the sample space (R^n, B, μ_k).

Step 2. Set $x^{k+1} = D(x^k, \xi^k)$, choose μ_{k+1}, set $k = k + 1$, and return to Step 1.

(R^n, B, μ_k) is probability space on iteration k. The μ_k are probability measures corresponding to distribution functions defined on R^n as conditional probability measures. The B is Borel subsets of R^n. B is the σ-algebra of subset of R^n. The map D with domain $S \times R^n$ and S satisfies the following condition.

Hypothesis 1 (H1). Consider $f(D(x,\xi)) \leq f(x)$; if $\xi \in S$, then $f(D(x,\xi)) \leq f(\xi)$.

Clearly, the monotone sequence $\{f(x^k)\}_{k=1}^{\infty}$ converges to the infimum of f on S. In order to avoid excluding some pathological situations, we replace the search for the infimum by that for α, the essential infimum of f on S, defined as follows:

$$\alpha = \inf\{x \mid v(z \in S \mid f(z) < x) > 0\}, \quad (4)$$

where v is a nonnegative measure defined on the (Borel) subsets B of R^n with $v(S) > 0$. Typically $v(A)$ is simply the n-dimensional volume of the set A; more generally v is the Lebesgue measure.

Hypothesis 2 (H2). For any subset A of S with $v(A) > 0$ we have that

$$\prod_{k=0}^{\infty}(1-\mu_k(A)) = 0. \quad (5)$$

It means that given any subset A of S with positive Lebesgue measure the probability of repeatedly missing the set A, when generating the random samples ξ^k, must be zero.

This requires that the sampling strategy determined by the choice of the ξ^k cannot rely exclusively on distribution functions concentrated on proper subsets of S of lower dimension (such as discrete distributions), or that the sampling strategy consistently ignore a part of S with positive "volume" (with respect to v).

Theorem 1 (convergence theorem (global search)). *Suppose that f is a measurable function. S is a measurable subset of R^n and (Hypotheses 1 and 2) is satisfied. Let $\{x^k\}_{k=1}^{+\infty}$ be a sequence generated by the algorithm. Then*

$$\lim_{k\to\infty} P\left[x^k \in R_\varepsilon\right] = 1, \quad (6)$$

where $P[x^k \in R_\varepsilon]$ is the probability that, at step k, the point x^k generated by the algorithm is in R_ε. R_ε is the optimality region.

2.3. Convergence Analysis of Traditional SCO Algorithm. Although traditional SCO may outperform other evolutionary algorithms in the early iterations, its performance may not be competitive as the number of generations is increased. Traditional SCO algorithm is not guaranteed to converge to global optimal solution. If the optimization algorithm satisfies the Hypotheses 1 and 2, general convergence proofs are given.

The traditional SCO algorithm saves the best solution in the knowledge, so it obviously satisfies Hypothesis 1. But, global optimization point GB_P of the agents in SCO algorithm will not be set at a random solution in the search space in the end of every generation. It is obvious that the algorithm does not satisfy Hypothesis 2 and is not an optimization algorithm with global search convergence properties according to Theorem 1. So the traditional SCO algorithm is not guaranteed to converge to global optimal solution with probability one.

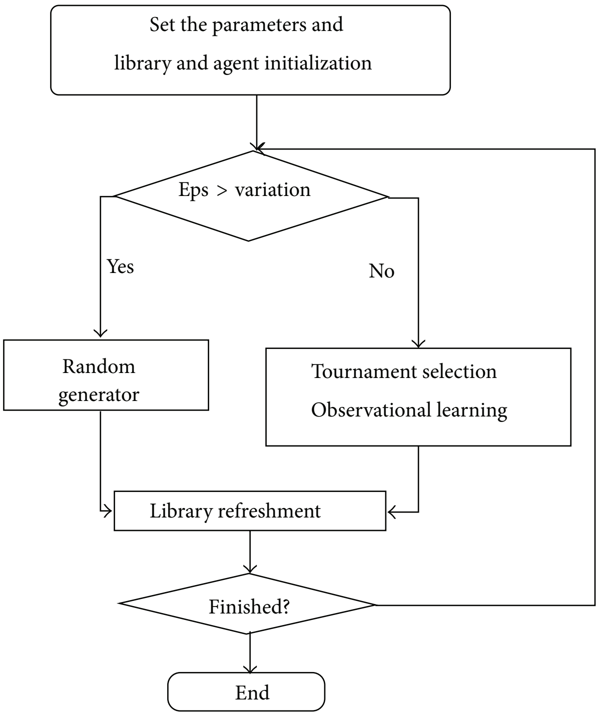


Figure 1: Flowchart of SSCO.

3. Method

3.1. Overview. Swarm intelligence optimization algorithms are also called metaheuristic algorithms because they provide a high-level framework which can be adapted to solve optimization problems. So, when swarm intelligence is used to solving a specific problem, it must be modified to fit the problem.

The SCO's iterative procedure is rooted in human intelligence with the social cognitive theory. When the learning agent falls into the local optimal, learning agents learn without observation, but with full randomicity by stochastic search, which helps to increase the global ergodicity of the knowledge library and to avoid premature convergence.

In this paper, a novel stochastic social cognitive optimization (SSCO) algorithm based on periodic partial reinitialization is proposed to solve NLPs to improve the global convergence speed of social cognitive optimization (SCO) algorithm. Simulation results show that the performance of SSCO is evidently better than traditional SCO for NLPs.

3.2. Stochastic Social Cognitive Optimization (SSCO) Algorithm. In our study, we incorporate the periodic partial reinitialization of the population into the SCO to enhance the overall performance of the algorithm. Figure 1 shows the flowchart of SSCO. The SSCO is described as follows.

Step 1 (initialization). (a) Set the parameters of SSCO: N_p, N_a, T, τ_W, τ_B, T_R, ε_R, where N_p denotes the number of knowledge points in knowledge library; N_a denotes the number of learning agents. T denotes the times of maximum learning cycle; τ_B denotes the tournament width; and T_R

Table 1: Summary of eight test cases.

Func.	n	Type of func.	p	LI	NE	NI	a
G_1	13	Quadratic	0.0111%	9	0	0	6
G_2	20	Nonlinear	99.8474%	0	0	2	1
G_3	5	Quadratic	52.1230%	0	0	6	2
G_4	2	Cubic	0.0066%	0	0	2	2
G_5	10	Quadratic	0.0003%	3	0	5	6
G_6	2	Nonlinear	0.8560%	0	0	2	0
G_7	7	Polynomial	0.5121%	0	0	4	2
G_8	8	Linear	0.0010%	3	0	3	6

denotes the maximum iterate number of reinitialization; ε_R denotes the maximum variation of fitness in T_R generation.

(b) Randomly create N_p knowledge points in knowledge library (KL), and then evaluate their fitness values basis on objective function, and save the global optimization point GB_P: the best point with the best fitness.

(c) Assign a knowledge point in KL to a learning agent randomly, but not repeatedly.

Step 2. For each learning agent, SCA learning cycle is as follows.

(a) If the variation of fitness of the global optimization point SCA in the previous T_R generation is less than ε_R, a new point TS_O is randomly generated; if the new point TS_O is better than GB_P, TS_O is assigned to GB_P.

(b) Otherwise, we have the following.

  (1) Tournament selection: select a best knowledge point TS_P from arbitrary τ_B knowledge points in KL not repeatedly with SCA itself.

  (2) Observational learning: compare the fitness of TS_P with that of SCA. The neighborhood local searching for the better referring to the worse is finding a new point TS_O according to (1); if the new point TS_O is better than GB_P, TS_O is assigned to GB_P.

(c) Library refreshment: remove the worst knowledge point TS_w in KL, and add the new point into KL.

Step 3. Repeat Step 2 until a stop condition (e.g., maximum number of iterations or a satisfactory fitness value). The total evaluation times are $T_e = N_p + N_a * T$.

3.3. Convergence Analysis of SSCO Algorithm. Traditional SCO algorithm is not guaranteed to converge to global optimal solution with probability one. According to Theorem 1, the proof presented here casts the SSCO into the framework of a global stochastic search algorithm, thus allowing the use of Theorem 1 to prove convergence. It remains to show that the SSCO satisfies both (H1) and (H2).

Let $\{\text{GB}_{P,t}\}$ be a sequence generated by the SSCO algorithm, where $\text{GB}_{P,t}$ is the current best position of the swarm at time t.

Define function D:

$$D\left(\text{GB}_{P,t}, x_{i,t}\right) = \begin{cases} \text{GB}_{P,t}, & f\left(g\left(x_{i,t}\right)\right) \geq f\left(\text{GB}_{P,t}\right) \\ g\left(x_{i,t}\right), & f\left(g\left(x_{i,t}\right)\right) \geq f\left(\text{GB}_{P,t}\right), \end{cases}$$

$$f\left(D\left(x,\xi\right)\right) \leq f\left(x\right), \quad \text{if } \xi \in S, \text{ then } f\left(D\left(x,\xi\right)\right) \leq f\left(\xi\right). \tag{7}$$

The definition of D above clearly complies with hypothesis H1, since the sequence $x_{i,t}$ is monotonic by definition because of always saving the best solution in the knowledge.

If the SSCO algorithm satisfies hypothesis H2, the union of the sample spaces of the agents must cover S, so that $S \subseteq \bigcup_{i=1}^{s} M_{i,k}$ at time step t, where $M_{i,k}$ denotes the support of the sample space of agent i. Because every learning agent has a chance to get a random solution in the search space in every generation when the iteration is static in a certain precision, $M_{i,k} = S$, $\bigcup_{i=1}^{s} M_{i,k} = S$. Define the Borel subset A of S, and $A = M_{i,k}$; then $v(A) > 0$, $\mu_k(A) = \sum_{i=1}^{s} \mu_{k,t}(A) = 1$. Thus by hypothesis H2 satisfied by Theorem 1, SSCO can be convergent to global best solution with probability one.

4. Numerical Experiment

4.1. Experimental Settings. Nonlinear programming problems (NLPs) always are nonconvex, highly nonlinear, nondifferentiable, and discontinuous, which is a constrained global optimization problem; the traditional deterministic algorithms for solving the NLPs are very difficult. Furthermore, the constrained global optimization is NP-hard [13], which does not admit efficient deterministic solutions in practice.

In our experiments, the test cases of eight benchmark NLPs from literature [1] and literature [13] in Table 1, except four problems because of lack of result in the referenced literatures, will be applied to show the way in which the proposed algorithm works. The eight benchmark problems almost include all the kinds of the constraints (linear inequalities LI, nonlinear inequalities NI, and nonlinear equalities NE). With respect to constraint handling of the NLPs in the proposed algorithms, there are two effective methods [14], basic constraint handing (BCH) rule for common inequalities constraint and adaptive constraints relaxing (ACR) rule for equalities constraint. The rules are described in detail in [15].

Table 1 lists the parameters of each test case: number of variables, type of the function, relative size of the feasible region in the search space given by the ratio p, the number of

Table 2: Summary of results of the SSCO on eight test cases.

NLP.	Type	Opt.	GA			PSO	SCO	SSCO		
			Worst	Best	Avg.	Avg.	Avg.	Worst	Best	Avg.
G_1	min	−15	−14.6154	−14.7864	−14.7082	−14.7951	−14.8891	−14.8767	−15.0000	−14.9953
G_2	max	0.803553	0.79119	0.79953	0.79671	0.79592	0.75475	0.79124	0.80109	0.79638
G_3	min	−30665.5	−30645.9	−30645.5	−30645.3	−30655.429	−30665.539	−30665.539	−30665.539	−30665.539
G_4	min	−6961.814	−5473.9	−6952.1	−6342.6	−6781.913	−6961.812	−6961.805	−6961.814	−6961.813
G_5	min	24.306	25.069	24.620	24.826	24.713	24.742	24.681	24.306	24.359
G_6	max	0.095825	0.0291438	0.0958250	0.0891568	0.091573	0.095158	0.095231	0.095825	0.095811
G_7	min	680.63	683.18	680.91	681.16	690.642	680.699	680.811	680.631	680.671
G_8	min	7049.33	9659.3	7147.9	8163.6	7531.9	7407.7	7296.35	7049.41	7124.72

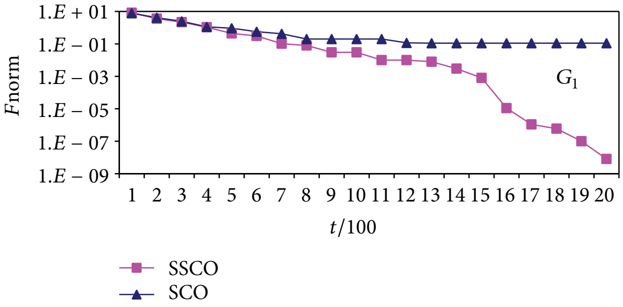


Figure 2: Fnorm versus $t/100$ for G_1 by SSCO and SCO.

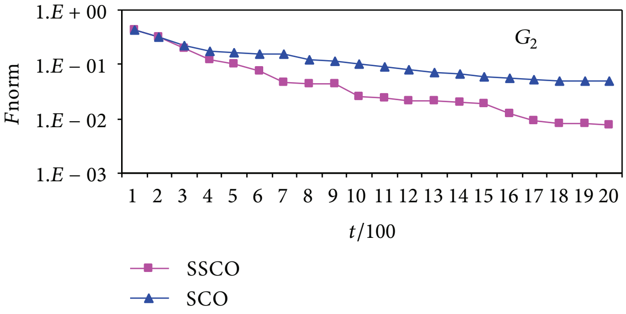


Figure 3: Fnorm versus $t/100$ for G_2 by SSCO and SCO.

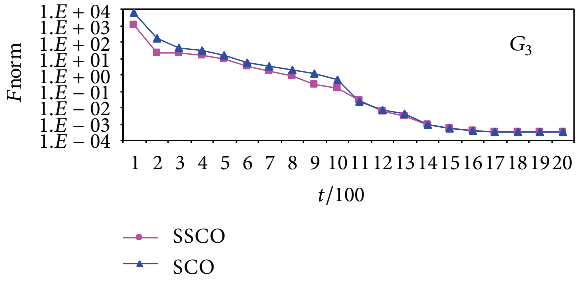


Figure 4: Fnorm versus $t/100$ for G_3 by SSCO and SCO.

constraints of each category (LI, NE, and NI), and the number a of active constraints at the optimum.

4.2. Result and Discussion. To evaluate the performance of the SSCO, we compare SSCO with standard gray-coded GA and traditional social cognitive optimization. The experiments setting for SSCO is the same as that of SCO from [1]. Let $T_R = 5$, $N_p = 98$, $N_a = 14$, $T = 2000$, (for G_6, $T = 200$). Each problem is executed 50 times. We calculate the best solution, worst solution, and mean solution by means of having statistical computation for each running of the SSCO and other algorithms. The experimental results obtained by other algorithms are provided in [1].

Table 2 shows the comparison of the test results between the SSCO and the known three algorithms. Opt. is the optimum value of each NLP. The results indicate that SSCO is superior to the known two algorithms from the viewpoints of the best solution, worst solution, and mean solution. Meanwhile, we can see that best solutions obtained by SSCO are better than other three algorithms since those solutions are much close to the true optimal solutions.

Figures 2, 3, 4, 5, 6, 7, 8, and 9 show the relative fitness value $F\text{norm} = |F\text{best} - F\text{opt}|$, which are performed by SCO and SSCO, versus $t/100$ for different benchmark NLPs, respectively, where t is current generation number. The SSCO, which has periodic partial reinitialization, shows higher convergence velocity and higher sustainable evolutionary capability at the process of evolution than traditional SCO. Furthermore, the iterative process of the SSCO has no more computing than traditional SCO.

For SSCO and SCO, the only difference is that the SSCO agents include the strategy of periodic restart, which is guaranteed to converge to the global optimization solution. It makes no effect on the total computing, however, because the SSCO considers static individuals are substituted with new randomly generated individuals to avoid the premature convergence in the learning iteration; the SSCO shows higher performance than SCO, PSO, and GA in all the eight test cases.

From the above analysis, the SSCO has higher efficiency in solving NLPs for reaching the near-optimal solutions. Consequently, the experimental results indicate that the SSCO has better robustness, effectiveness, and stableness than SCO and GA reported in the literature.

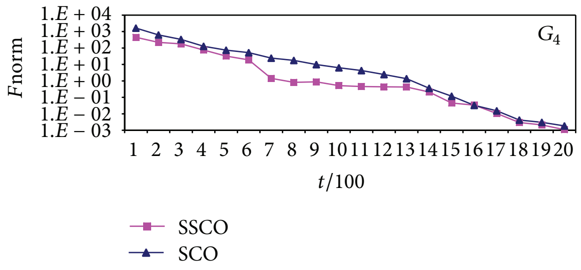


Figure 5: *F*norm versus $t/100$ for G_4 by SSCO and SCO.

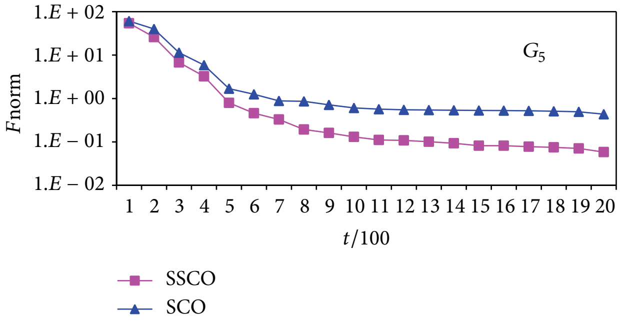


Figure 6: *F*norm versus $t/100$ for G_5 by SSCO and SCO.

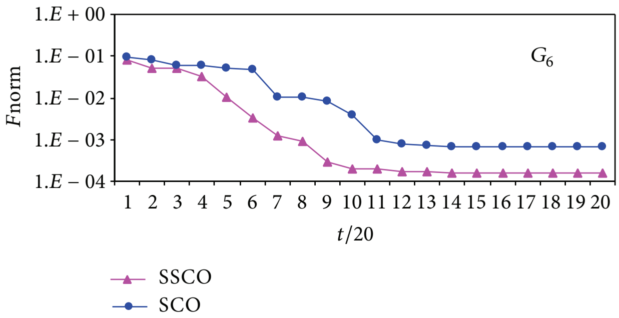


Figure 7: *F*norm versus $t/100$ for G_6 by SSCO and SCO.

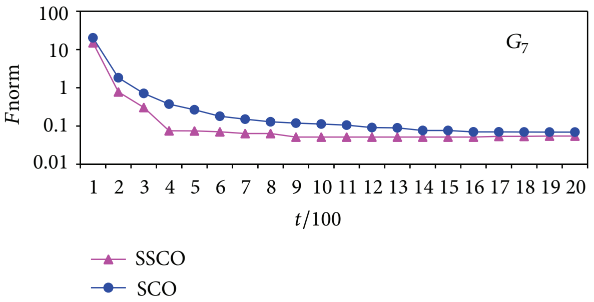


Figure 8: *F*norm versus $t/100$ for G_7 by SSCO and SCO.

5. Conclusions

A stochastic social cognitive optimization (SSCO) algorithm with the strategy of periodic restart is proposed in this paper for solving NLPs. The periodic restart of SCO for static individuals can avoid the premature convergence, improve the global searching performance, and ensure the algorithm to obtain stable optimal solution. The convergence proof for the stochastic SCO is given based on Solis and Wets' research results. The final experiment results indicate that the new algorithm has good capability to find optimal solution.

Appendix

The appendix provides the description of eight test functions.

G_1:

$$\begin{aligned}
\text{Minimize}\quad & G_1(\vec{x}) = 5x_1 + 5x_2 + 5x_3 + 5x_4 - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i \\
\text{Subject to}\quad & 2x_1 + 2x_2 + x_{10} + x_{11} \le 10, \\
& 2x_1 + 2x_3 + x_{10} + x_{12} \le 10, \\
& 2x_2 + 2x_3 + x_{11} + x_{12} \le 10, \\
& -8x_1 + x_{10} \le 0, \quad -8x_2 + x_{11} \le 0, \\
& -8x_3 + x_{12} \le 0, \quad -2x_4 - x_5 + x_{10} \le 0, \\
& -2x_6 - x_7 + x_{11} \le 0, \quad -2x_8 - x_9 + x_{12} \le 0,
\end{aligned} \tag{A.1}$$

where

$$\begin{gathered}
0 \le x_i \le 1, \quad i = 1, \ldots, 9, \\
0 \le x_i \le 100, \quad i = 10, 11, 12, \\
0 \le x_{13} \le 1
\end{gathered} \tag{A.2}$$

$$G_1(\vec{x}^*) = -15.$$

G_2:

$$\begin{aligned}
\text{Maximize}\quad & G_2(\vec{x}) = \left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right| \\
\text{Subject to}\quad & \prod_{i=1}^{n} x_i \ge 0.75, \quad \sum_{i=1}^{n} x_i \ge 7.5n,
\end{aligned} \tag{A.3}$$

where

$$\begin{gathered}
0 \le x_i \le 10 \quad \text{for } 1 \le i \le n. \\
G_2(\vec{x}^*) = 0.803553.
\end{gathered} \tag{A.4}$$

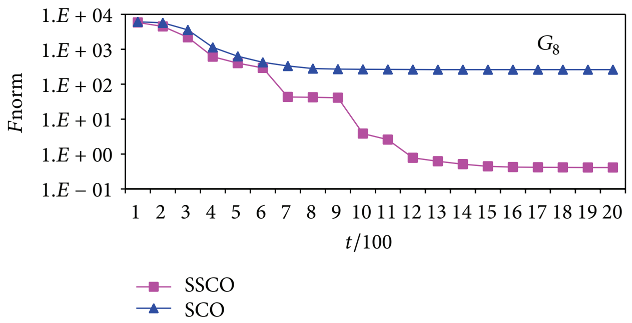


Figure 9: *F*norm versus $t/100$ for G_8 by SSCO and SCO.

G_3:

$$\begin{aligned}\text{Minimize}\quad & G_3(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 \\ & \qquad + 37.293239x_1 - 40792.141 \\ \text{Subject to}\quad & 0 \le 85.334407 + 0.0056858x_2x_5 \\ & \qquad + 0.0006262x_1x_4 - 0.0022053x_3x_5 \le 92 \\ & 90 \le 80.51249 + 0.0071317x_2x_5 \\ & \qquad + 0.0029955x_1x_2 + 0.0021813x_3^2 \le 110 \\ & 20 \le 9.300961 + 0.0047026x_2x_5 \\ & \qquad + 0.0012547x_1x_3 + 0.0019085x_3x_4 \le 25,\end{aligned} \tag{A.5}$$

where

$$\begin{gathered} 78 \le x_1 \le 102, \quad 33 \le x_2 \le 45, \\ 27 \le x_i \le 45 \quad \text{for } i = 3,4,5 \\ G_3(\vec{x}^*) = -30665.5.\end{gathered} \tag{A.6}$$

G_4:

$$\begin{aligned}\text{Minimize}\quad & G_4(\vec{x}) = (x_1 - 10)^3 + (x_2 - 20)^3 \\ \text{Subject to}\quad & (x_1 - 5)^2 + (x_2 - 5)^2 - 100 \ge 0 \\ & -(x_1 - 6)^2 - (x_2 - 5)^2 + 82.81 \ge 0,\end{aligned} \tag{A.7}$$

where

$$\begin{gathered} 13 \le x_1 \le 100, \quad 0 \le x_2 \le 100, \\ G_4(\vec{x}^*) = -6961.81381.\end{gathered} \tag{A.8}$$

G_5:

$$\begin{aligned}\text{Minimize}\quad & G_5(\vec{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 \\ & \qquad + (x_3 - 10)^2 + 4(x_4 - 5)^2 \\ & \qquad + (x_5 - 3)^2 + 2(x_6 - 1)^2 \\ & \qquad + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 \\ & \qquad + (x_{10} - 7)^2 + 45 \\ \text{Subject to}\quad & 105 - 4x_1 - 5x_2 + 3x_7 - 9x_8 \ge 0, \\ & -3(x_1 - 2)^2 - 4(x_2 - 3)^2 - 2x_3^2 \\ & + 7x_4 + 120 \ge 0, \\ & -10x_1 + 8x_2 + 17x_7 - 2x_8 \ge 0, \\ & -x_1^2 - 2(x_2 - 2)^2 + 2x_1x_2 - 14x_5 + 6x_6 \ge 0, \\ & 8x_1 - 2x_2 - 5x_9 + 2x_{10} + 12 \ge 0, \\ & -5x_1^2 - 8x_2 - (x_3 - 6)^2 + 2x_4 + 40 \ge 0, \\ & 3x_1 - 6x_2 - 12(x_9 - 8)^2 + 7x_{10} \ge 0, \\ & -0.5(x_1 - 8)^2 - 2(x_2 - 4)^2 - 3x_5^2 + x_6 + 30 \ge 0,\end{aligned} \tag{A.9}$$

where

$$\begin{gathered} -10.0 \le x_i \le 10.0, \quad i = 1,\ldots,10, \\ G_5(\vec{x}^*) = 24.3062091.\end{gathered} \tag{A.10}$$

G_6:

$$\begin{aligned}\text{Maximize}\quad & G_6(\vec{x}) = \frac{\sin^3(2\pi x_1)\cdot\sin(2\pi x_2)}{x_1^3\cdot(x_1 + x_2)} \\ \text{Subject to}\quad & x_1^2 - x_2 + 1 \le 0 \\ & 1 - x_1 + (x_2 - 4)^2 \le 0,\end{aligned} \tag{A.11}$$

where

$$\begin{gathered} 0 \le x_1 \le 10, \quad 0 \le x_2 \le 10, \\ G_6(\vec{x}^*) = 0.095825.\end{gathered} \tag{A.12}$$

G_7:

$$\begin{aligned}\text{Minimize}\quad G_7(\vec{x}) = {} & (x_1-10)^2 + 5(x_2-12)^2 + x_3^4 \\ & + 3(x_4-11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 \\ & - 4x_6x_7 - 10x_6 - 8x_7 \\ \text{Subject to}\ & 127 - 2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 \geq 0, \\ & 282 - 7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 \geq 0 \\ & 196 - 23x_1 - x_2^2 - 6x_6^2 + 8x_7 \geq 0, \\ & -4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 - 5x_6 + 11x_7 \geq 0,\end{aligned} \tag{A.13}$$

where

$$\begin{gathered} -10.0 \leq x_i \leq 10.0, \quad i = 1, \ldots, 7, \\ G_7(\vec{x}^*) = 680.6300573. \end{gathered} \tag{A.14}$$

G_8:

$$\begin{aligned}\text{Minimize}\quad & G_8(\vec{x}) = x_1 + x_2 + x_3 \\ \text{Subject to}\quad & 1 - 0.0025(x_4 + x_6) \geq 0, \\ & 1 - 0.0025(x_5 + x_7 - x_4) \geq 0 \\ & 1 - 0.01(x_8 - x_5) \geq 0, \\ & x_1x_6 - 833.33252x_4 - 100x_1 + 83333.333 \geq 0 \\ & x_2x_7 - 1250x_5 - x_2x_4 + 1250x_4 \geq 0, \\ & x_3x_8 - 1250000 - x_3x_5 + 2500x_5 \geq 0,\end{aligned} \tag{A.15}$$

where

$$\begin{gathered} 100 \leq x_1 \leq 10000, \quad 1000 \leq x_i \leq 10000, \\ i = 2, 3, \quad 100 \leq x_i \leq 10000, \quad i = 4, \ldots, 8, \\ G_8(\vec{x}^*) = 7049.330923. \end{gathered} \tag{A.16}$$

Conflict of Interests

The authors (Jia-ze Sun, Shu-yan Wang, and Hao Chen) declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (611721701, 61203311, 61105064, and 61050003), by the Scientific Research Program of Shaanxi Provincial Education Department (12JK0732 and 13JK1183), by the Natural Science Foundation of XUPT (ZL2013-26), and by special funding for Key Discipline Construction of General Institutions of Higher Learning from Shaanxi Province and special funding for course development for Xi'an University of Posts and Telecommunications.

References

[1] X.-F. Xie, W.-J. Zhang, and Z.-L. Yang, "Social cognitive optimization for nonlinear programming problems," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 779–783, Beijing, China, November 2002.

[2] J. Z. Sun, S. Y. Wang, J. K. Zhang, and X.-P. Cao, "SCO algorithm based on entropy function for NCP," *Computer Engineering and Applications*, vol. 46, no. 21, pp. 40–42, 2010, (Chinese).

[3] K. J. Zhang, Z. J. Sun, and X. L. Kou, "Social cognitive optimization for fractional programs," *Computer Engineering and Design*, vol. 29, no. 21, pp. 5543–5545, 2008 (Chinese).

[4] J. Sun Z and J. Zhang, "Solving nonlinear systems of equations based on social cognitive optimization," *Computer Engineering and Applications*, vol. 44, no. 28, pp. 42–44, 2008 (Chinese).

[5] X. F. Xie and W. J. Zhang, "Solving engineering design problems by social cognitive optimization," in *Genetic and Evolutionary Computation—GECCO 2004*, vol. 3102 of *Lecture Notes in Computer Science*, pp. 261–262, Springer, Berlin, Germany, 2004.

[6] Y.-P. Chen, J.-K. Zhang, J.-Z. Sun, Q.-H. Zheng, and Z.-Z. Li, "A service selection model using mixed intelligent optimization," *Chinese Journal of Computers*, vol. 33, no. 11, pp. 2116–2125, 2010 (Chinese).

[7] L.-J. Wang, C.-J. Wei, and C.-X. Li, "Social cognitive optimization modified with self-organizing migrating algorithm," *Computer Engineering*, vol. 22, pp. 201–203, 2008.

[8] L. Ma, R.-X. Wang, and Y. P. Chen, "Solving the nonlinear problem based on the modified social cognitive optimization algorithm," *Computer Engineering*, vol. 10, pp. 170–172, 2011.

[9] J. Z. Sun, G. H. Geng, S. Y. Wang, and M. Q. Zhou, "Hybrid social cognitive optimization algorithm for constrained nonlinear programming," *Journal of China Universities of Posts and Telecommunications*, vol. 19, no. 3, pp. 91–99, 2012.

[10] L. Zhi-zhong, W. Zhi-Jian, and X. Xiao, "Research on cloud service composition based on culture social cognitive optimization algorithm," *Computer Science*, vol. 40, no. 5, pp. 103–106, 2013.

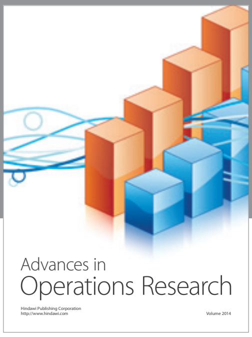
[11] J.-Z. Sun, G.-H. Geng, and X.-Y. Pan, "D-S algorithm based on SCO for matching fragments," *Applied Mechanics and Materials*, vol. 411–414, pp. 1876–1879, 2013.

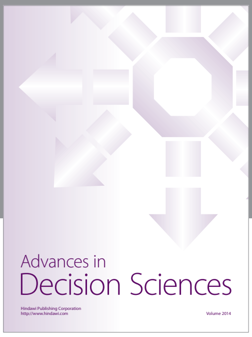
[12] F. J. Solis and R. J.-B. Wets, "Minimization by random search techniques," *Mathematics of Operations Research*, vol. 6, no. 1, pp. 9–30, 1981.

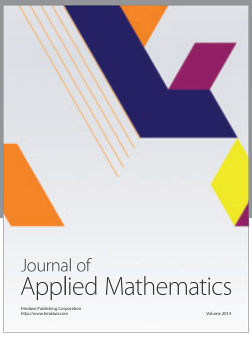
[13] S. Koziel and Z. Michalewicz, "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization," *Evolutionary Computation*, vol. 7, no. 1, pp. 19–44, 1999.

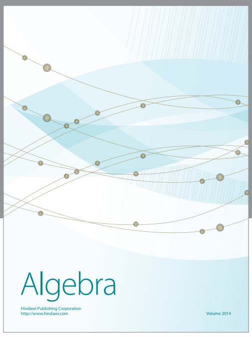
[14] X.-F. Xie, W.-J. Zhang, and D.-C. Bi, "Handling equality constraints by adaptive relaxing rule for swarm algorithms," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, pp. 2012–2016, Portland, Ore, USA, June 2004.

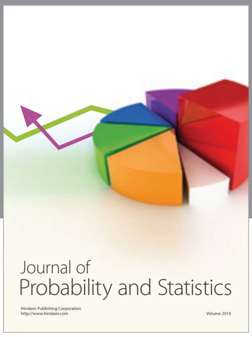
[15] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 311–338, 2000.


Advances in
Operations Research

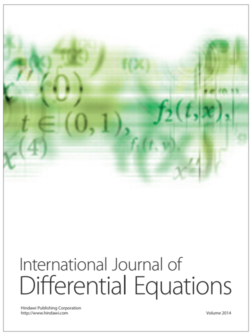

Advances in
Decision Sciences


Journal of
Applied Mathematics

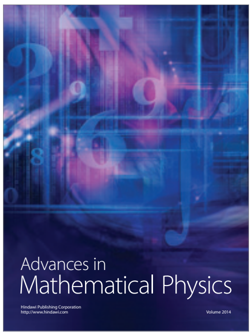

Algebra

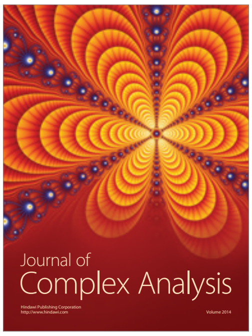

Journal of
Probability and Statistics

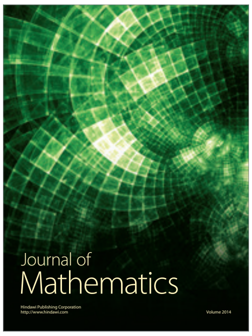
The Scientific
World Journal

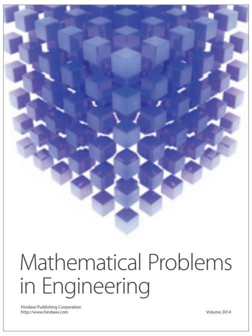

International Journal of
Differential Equations

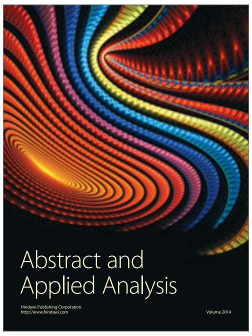
International Journal of
Combinatorics

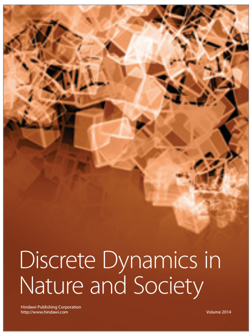

Advances in
Mathematical Physics

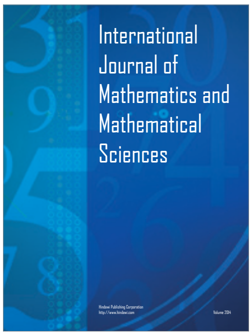

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

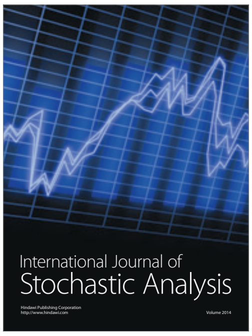

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization