An Improved Cross-Entropy Method Applied to Inverse Problems

Siguang An [1,2], Shiyou Yang [1], S. L. Ho[3], and Peihong Ni[1]
[1]College of Electrical Engineering, Zhejiang University, Hangzhou, 310027, China
[2]Department of Electrical Engineering, China Jiliang University, Hangzhou, 310018, China
[3]The Hong Kong Polytechnic University, Hong Kong

Abstract — An improved cross-entropy method with continuous variables for global optimization of inverse problems is proposed. To alleviate the low convergence rates of conventional algorithms, improvements in algorithm design mechanism and iterative process are proposed. To monitor and guide the searching process, the design space is divided into several sub-domains and three indicators are introduced for each sub-domain to evaluate its performances. To balance exploitation and exploration searches, the whole iterative process is designed to include both the diversification and intensification phases. A novel mechanism is introduced to enhance the diversity to allow the algorithm escaping from the local optima in the diversification phase. The shifting away from the worst sub-domain in the intensification phase empowers the algorithm with enhanced convergence rates. The proposed method is applied to a mathematic function and a typical electromagnetic inverse problem: TEAM Workshop 22. Numerical results are reported to validate the effectiveness and efficiency of the proposed method.

I. An Improved Cross-Entropy Method

A. Sub-domains Divisions

The CE method was first proposed by Rubinstein and Kroses in 2004 [1]. The main idea is to mapping the optimization problem to a rare event estimation problem and tackling it using an adaptive algorithm with two iterative steps: (1) generate a random data sample according to a specified probability density function (pdf); (2) update the parameters of the pdf using elite solutions of the sample data to produce better samples [2]. The elite solutions occupy a very small proportion of the samples, typically in the order of 1%-1.5%. Most of the samples are abandoned without being used effectively for directing the subsequent searches. To increase the usage percentage of the samples and also to monitor the searched spaces, the feasible space is divided into *r* sub-domains. It is noted that the convergence speed of the algorithm is dependent on *r* in that if it is too large, more computational cost is needed to classify the samples; if it is too small, the algorithm will be easily trapped in a local optima. Empirically *r* is given as:

$$r = m^n \tag{1}$$

where, *n* is the number of design variables, *m* is the number of sub-domains.

For $n \le 3$, the value of *m* is in the range [5,10]. For $n \ge 4$, *m* is set to be 3~5 [3]. The three indicators defined to evaluate the performance of each sub-domain are: $reward_i(t)$: the total rewards for the i^{th} sub-domain up to the *t* sampling instant; $num_i(t)$: the number of times for the i^{th} sub-domain to be selected up to the *t* sampling instant; $precent_i(t)$: the reward percentage for the i^{th} sub-domain up to the *t* sampling instant, as expressed as:

$$precent_i(t) = reward_i(t) / num_i(t) \tag{2}$$

To estimate the performance of the sub-domains in the iterative process, for a minimization problem, these indictors are updated according to:

$$reward_i(t+1) = reward_i(t) + \frac{\max_{current} - f(x_i)}{\max_{current} - \min_{current}} \tag{3}$$

$$num_i(t+1) = num_i(t) + 1 \tag{4}$$

$$precent_i(t+1) = reward_i(t+1) / num_i(t+1) \tag{5}$$

B. Diversification Phase and a Novel Mechanism

The algorithm starts from a diversification phase to uniformly explore the whole feasible space without getting trapped onto a local optimum. Because of implementation simplicity, the normal distribution function $N(\mu, \sigma^2)$, with its mean μ and standard deviation σ, is selected as the *pdf*s. As a result, the updating formulae for the selected *pdf* in this phase are proposed as:

$$\mu(t+1) = x\_best(t) \tag{6}$$

$$\sigma(t+1) = std(x_{\rho N}(t)) \tag{7}$$

$$\sigma(t+1) = \beta\sigma(t+1) + (1-\beta)\sigma(t) \tag{8}$$

where, $x\_best(t)$ is the best point at the t^{th} sampling instant, $X_{\rho N}(t)$ are the elite solutions, β is the smoothing parameter ranging between 0.6 to 0.9, ρ is the percentage of elite solutions in the range of 1%~1.5%, *N* is the sample size. Moreover,

$$\mu_M = (1 - rand) \times \mu + rand \times new\_para \tag{9}$$

$$\sigma_M = \frac{1}{5}(x_u - x_l) \tag{10}$$

$$new\_para = rand \times sub\_space(\min(num_i)) \tag{11}$$

$$N_M = 2 \times N \tag{12}$$

where new_para is the random point in the sub-domain having the minimal evaluating indictor num_i.

When the number of successive iterations without improvements in the best objective function so far searched exceeds a predefined value in the diversification phase, it is futile to continue the exploiting search around the current solutions [4]. Accordingly, a novel modification mechanism is introduced to update μ and σ using (9)-(11), while the size of samples is increased in accordance to (12).

C. Intensification Phase and Shifting away from the Worst Sub-domains

An intensification phase is designed to efficiently find the global optimal solution. For this purpose, a dynamic

smoothing parameter is used to update the parameters of the *pdfs*. More specially,

$$\mu(t+1) = x\_best\_overall(t) \quad (13)$$

$$\sigma(t+1) = std(x_{\rho N}(t)) \quad (14)$$

$$\sigma(t+1) = \beta\sigma(t+1) + (1-\beta)\sigma(t) \quad (15)$$

$$\beta = \sigma_0 - \sigma_0(1-1/t)^q \quad (16)$$

$$\sigma_0 = \frac{1}{5}(x_u - x_l) \quad (17)$$

where; $x\_best\_overall(t)$ is the best sample point up to the t^{th} sampling instant; q is the attenuation parameter, generally within the range of 2~5; x_u, x_l are, respectively, the upper and lower limits of the variables.

To increase the convergence speed, most sampling points should be focused around the current best solution in the intensification phase. In this regard, in order to avoid generating samples that are far away from the current best solution, a mechanism to shift the solution away from the worst sub-domains is proposed. According to the definition of parameter $precent_i(t)$, this goal can be realized using:

$$x_i(j) = precent_i \times x_i(j) + (1-precent_i) \times x\_best\_overall(j) \quad (18)$$

D. Algorithm Description

To facilitate the implementation of the proposed CE method, its iterative procedures are summarized as:

Step 1: Define the number of variables and use the sub-domain of each variable to divide the feasible space into *r* sub-domains. Initialize $\mu_0 = (x_u + x_l)/2$, $\sigma_0 = (x_u - x_l)/3$, the sampling index *t*=0. the modification-index parameter *flag*=0; Also, define N_D the sampling number of the diversification phase; N_I the sampling number of the intensification phase; *N* the number of points in one sampling; ρ the percentage of the elite solutions; *w* the threshold value for successive iterations without improvement in the current best solutions; Generate *N* sampling points *X*.

Step 2: Calculate the function value of the sampling points and the parameters of the sub-domains which are, namely, $reward_i(t)$, $num_i(t)$, $precent_i(t)$; Record the current best point x_best and global best point $x\_best\_overall$. If the global best point has no improvement for predefined number of successive iterations, a threshold *w* is selected; set *flag*=1.

Step 3: In case that $t \le N_D$, and *flag*=0, use (6)(7) to update *pdf*; if *flag*=1, use (9)(10)(12) to update the *pdf* and the number of sampling points. In case that $t > N_D$, use (13)(14) to adjust the *pdf*. Use (18) to shift the solution away from the worst sub-domain.

Step 4: Generate new sampling points based on the new parameter of the pdf.

Step 5: if $t < (N_D + N_I)$, *t*=*t*+1, go to step 2; else go to step 6.

Step 6: Stop the algorithm.

II. Numerical Results

To validate and demonstrate the advantages of the proposed algorithm, a non-linear mathematic function and the TEAM workshop Problem 22 are solved [5],[6].

The global optimum of the non-linear function is located at point (3, 4) with the objective value of 1.2112. The parameters of the proposed algorithm for solving this function are set as: *N*=20, *r*=25, ρ =0.01, N_D=25, N_I=15, *w*=3,*q*=2, β =0.7. Table I shows a comparison of the averaged performance between the original and the proposed CE methods with 20 random running. Table II tabulates the final solution and the global optimum of one typical run among 20 random runs in solving the TEAM workshop Problem 22 for the proposed algorithm. The algorithm parameters in this case study are set as: *N*=40, *r*=125, ρ =0.01, N_D=25, N_I=20, *w*=3, *q*=2, β =0.8. Table III gives a comparison of the final solution and number of iterations between the original and the proposed CE methods. Obviously, the proposed algorithm can find the same qualified solutions with much less iterations in the case study being reported in this paper.

TABLE I
PERFORMANCE COMPARISON OF THE PROPOSED AND THE ORIGINAL CE METHODS IN SOLVING THE MATHEMATICAL FUNCTION

No. Iterations to find global optimum	No. of Averaged iterations	
	Proposed	Original CE
20/20	832	28965

TABLE II
THE FINAL SOLUTION OF A TYPICAL RUN OF THE PROPOSED ALGORITHM

Algorithm	Final Solution	f_{opt}	No. Iterations
Original CE	3.0, 3.9954	1.2112	30693
The proposed	3.0 , 3.9955	1.2112	838

TABLE III
PERFORMANCE COMPARISON OF THE PROPOSED AND THE ORIGINAL ALGORITHMS ON TEAM WORKSHOP PROBLEM 22

Algorithm	r_2 (m)	$h_2/2$ (m)	d_2 (m)	*f*_opt	No. Iterations
Original CE	3.0835	0.2414	0.3895	0.0858	59376
The proposed	3.0888	0.2404	0.3886	0.0860	1186

III. References

[1] R. Y. Rubinstein and D.P. Krosese, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*, NY: Springer-Verlag , 2004, pp.29-57.

[2] Dirk P. Kroese, Sergey Porotsky, Reuven Y. Rubinstein, "The cross-entropy method for continuous multi-external optimization," *Methodol Comput. Appl. Probab*., vol.8, no.3 pp.383-407, 2005

[3] Xianyi Zeng and Zeyi Liu, "A learning automata based algorithm for optimization of continuous complex functions," *Information Sciences*, vol.174, no.3, pp.165-175, 2005.

[4] S. L. Ho and Shiyou Yang, "The Cross-entropy method and its application to inverse problems," *IEEE Trans. on Magn*., vol.46, no.4, pp.3401-3404, 2010.

[5] P. Alotto, U. Baumgartner, F. Freschi, et.al., "SMES optimization benchmark extended: introducing Pareto optimal solution into TEAM 22," *IEEE Trans. Magn*., vol.44. no.6, pp.1066-1069, 2008.

[6] http://www-igte.tu-graz.ac.at/archive/team/team3dis.htm.