Multi-Objective Optimization with Cross Entropy Method: Stochastic Learning with Clustered Pareto Fronts

Ahmet Ünveren, *Member* and Adnan Acan, *Senior Member, IEEE*

***Abstract*—This paper presents a novel multiobjective optimization strategy based on the cross entropy method (MOCE). The cross-entropy method (CE) is a stochastic learning algorithm inspired from rare event simulations and proved to be successful in the solution of difficult single objective real-valued optimization problems. The presented work extends the use of cross-entropy method to real-valued multiobjective optimization. For this purpose, parameters of CE search are adapted using the information collected from clustered nondominated solutions on the Pareto front.**

Comparison with well known multiobjective optimization algorithms on the solution of provably difficult benchmark problem instances demonstrated that CEMO performs at least as good as its competitors.

I. Introduction

MULTIOBJECTIVE optimization (MOO) framework provides more realistic formulation of many real-life problems since a set solutions, rather than a single solution, exhibiting different forms of concession among multiple and often conflicting objectives is provided as result of the optimization process [1]-[11]. Such a set of solutions is commonly known as a Pareto-optimal set in which Pareto-optimality is defined in terms of a dominance relation between two solutions as follows: given two solutions u and v, $u \neq v$, u is said to dominate v if u is not worse than v in all objectives and u is strictly better than v for at least one objective. For example, for a maximization problem,

$$\begin{aligned} &max\ f(x)=(f_1(x), f_2(x),\ldots,f_K(x)) \\ &x=(x_1,x_2,\ldots,x_n) \in R^n \end{aligned} \quad (1)$$

solution vector u is better than solution vector v with respect to objective i, if $f_i(u) \geq f_i(v)$, and u is said to dominate v, denoted as $u \succ v$, if and only if

$$\begin{aligned} &f_i(u) \geq f_i(v) \ \ for\ i=1,2,\ldots,j-1,j+1,\ldots K \\ &f_j(u) > f_j(v) \ \ for\ at\ least\ one\ 1<= j <= K. \end{aligned} \quad (2)$$

A common difficulty with multi-objective optimization problems is the presence of a number of conflicting objectives and, in general, none of the feasible solutions allow simultaneous optimality for all objectives. Hence, any favorable Pareto-optimum provides a solution exhibiting a subjective compromise between the problem objectives. In order to find such a solution, classical methods transform a multiobjective optimization problem into a single-objective one through different scalarization and objective combination methods that include serious drawbacks in terms of appropriate representation of the real-world problem and quality of resulting solutions [13].

In order to have better mathematical models for real-world problems and increase the efficiency of search within arbitrarily complex solution spaces through providing a set of solutions rather than a single solution, some advanced MOO techniques have been proposed in the last few years [1]-[17]. These techniques are generally based on some metaheuristics such as Simulated Annealing, Evolutionary Algorithms, Tabu Search, Particle Swarm Optimization, Artificial Immune Systems, Cultural Algorithms and Ant Colony Optimization.

In this study, the use of the CE method in multiobjective optimization (MOO) is presented. The CE method is a stochastic learning algorithm inspired from the concept of rare event simulations which involve the estimation of parameters for a number of probability distributions associated with some rare events. Application of the CE method is carried out in two phases. First, a sample of random data is generated using a family of probability distribution functions. Secondly, parameters of the probability distribution functions are updated on the basis of a performance metric on the generated sample, so as to produce a better sample at the next iteration.

In applications of the CE method for single objective optimization problems, a subset of best performing samples is extracted from a large set of samples and parameters of the underlying probability distributions are adapted based on the mean and standard deviation of this subset of elite samples. Since, it is not objectively easy to define elite individuals in MOO, we introduced the notion of clustered nondominated solutions on the Pareto front to adapt the probability distribution parameters in applying CE method for MOO.

In principle, the set of all nondominated solutions is divided into clusters, using the fuzzy c-means (FCM) algorithm, and the statistical features of each cluster are used to adapt the parameters of an associated CE method. That is, associated with each cluster, there is a dedicated CE method which adapts itself using statistical features of its cluster and explores the solution space as directed by these features. Hence, each CE method is expected to find nondominated solution within a particular region of the Pareto boundary because clusters are determined based on the Euclidean distance measure by the FCM algorithm. The details of

A. Ünveren is with the Eastern Mediterranean University, Computer Engineering Department, TRNC, Via Mersin 10, TURKEY (corresponding author phone: +90 392 6301193; fax: +90 392 3650711; e-mail: ahmet.unveren@emu.edu.tr).

A. Acan is with the Eastern Mediterranean University, Computer Engineering Department, TRNC, Via Mersin 10, TURKEY (e-mail: adnan.acan@emu.edu.tr).

implementation for the proposed MOCE strategy are given in Section 3.

This paper is organized as follows. Basic principles of the CE method as applied for single objective numerical optimization problems are introduced in Section 2. The proposed multiobjective CE strategy is described with its implementation details in Section 3. Section 4 covers the experimental setup, results, and related discussions. Finally, conclusions and future research directions are given in Section 5.

II. THE CROSS ENTROPY METHOD FOR FUNCTION OPTIMIZATION

The CE method, pioneered by Rubinstein in 1997 as a stochastic learning algorithm for estimating probabilities of rare events, has been broadened as a generic and efficient tool for solving difficult numerical and NP-hard combinatorial optimization problems [18].

Regardless of the application at hand, the implementation mechanisms of the CE method remain the same. In principle, the CE method is iteratively running in two phases; random sampling of the object parameters space and adapting algorithm's parameter based on good hits during the first phase. The main idea is to take more and more samples around the good hits where the goodness is measured using a predefined objective function [16]-[19].

Mathematical preliminaries of the CE method, as it is used for single-objective function optimization, can be stated as follows: Consider a solution space X and a performance metric $S(x)$ defined over X such that $S(x) \in R$, for all $x \in X$. We are interested in finding those elements of X for which $S()$ reaches its global minimum/maximum value. If we define the extreme value of interest for $S(.)$ as γ^*, then there is a set of samples $X^* \subset X$ such that

$$S(x^*) = \gamma^*, \text{ for all } x^* \in X^*. \quad (3)$$

Using the inspirations taken from rare-event simulations, the CE method considers the above described search (and optimization) problem as an estimation problem in which parameters of a family of probability density functions are to be estimated based on features extracted from random samples of X.

For the purpose of iteratively estimating samples of X^*, a collection of indicator functions $I_{\{S(x) \geq \gamma\}}$ that indicates if $S(x)$ is above the level γ for sample x, are defined. If $\{f(.,v), v \in R^m\}$ is a family of probability density functions on X, then for a vector u, $|m|$, of probability density function parameters, the optimization problem will be transformed into the problem of estimating the probability $P_u(S(X) \geq \gamma)$. Using the indicator functions as described above this probability is equal to,

$$l(\gamma) = P_u(S(X) \geq \gamma) = \sum_x I_{\{S(x) \geq \gamma\}} f(x,u) = E_u I_{\{S(x) \geq \gamma\}} \quad (4)$$

Where P_u is the probability distribution associated with the probability density function $f(.,u)$, and E_u denotes the corresponding expectation operator. When $\gamma = \gamma^*$, $l(\gamma)$ can be estimated using the likelihood ratio estimator with reference to $v^* \in R^m$ as [15],

$$v^* = \arg\max_v \frac{1}{N} \sum_{i=1}^{N} I_{\{S(X_i) \geq \gamma\}} \ln f(X_i, v) \quad (5)$$

where X_i's are generated using pdf $f(.,v)$. Hence, if γ becomes closer to γ^*, $f(.,v^*)$ takes most of its samples close to x^*, and can be used generate an approximate solution to the underlying search/optimization problem with respect to the performance metric $S(x)$. It is important to note that, instead of finding the optimal solutions x^* to a particular problem directly, the CE method aims to find the most favorable sampling density $f(.,v^*)$ such optimal (or near optimal) solutions can be sampled from it.

The single objective numerical function optimization with CE method entiles the following general iterative procedure, where Gaussian type probability density functions are assumed to be used:

Algorithm I. CE Algorithm for Function Optimization

1. Initialize algorithm parameters, like means (mu) and standard deviations (sigma) of pdf's, sample size N, and tolerance variable *eps* that is used for termination check.
2. Generate N sample vectors X_i, $i=1,...,N$ as:
   $x_{i,j} = mu + sigma * randn(), i = 1,..., N; j = 1,..., m.$
   where *randn()* produces Gaussian distributed random numbers and m defines the number of object parameter variables.
3. Compute sample performances $S(X_i)$, $i=1,..., N$.
4. Determine a subset of best performing samples and update pdf parameters using features, such as mean and standard devation.
5. Check if stopping criterion is met, for example: if maximum iteration count is exceeded or pdf parameters are below the tolerance value eps. If yes, then output the best-so-far sample and terminate, otherwise go to step 2.

III. THE PROPOSED MULTIOBJECTIVE CROSS ENTROPY OPTIMIZATION APPROACH

The most important step in applying the CE method for function optimization is to update the algorithm's current parameters in a way that the next data samples will improve the performance metric. In multiobjective function optimization, the set of nondominated solutions on the Pareto front is clearly the set or best performing samples because goodness of solutions is defined in terms of dominance relations. Since, in MOO, one looks for the set of globally nondominated solutions, it is natural to consider the set of nondominated solutions at each iteration to extract features required to update *pdf*'s parameters. However, since nondominated solutions are distributed along the Pareto front, guiding the CE method with features extracted from

the overall set misleads the algorithm and causes exploration of nondominated solutions in the middle of the Pareto front much more frequently than compared to tails towards the axis. This is an expected phenomena because taking means of samples over iterations causes *pdf* to take more and more samples towards the middle of Pareto front. In order to overcome this problem, the presented CE strategy partitions the set of nondominated solutions into clusters using the fuzzy c-means (FCM) clustering algorithm [20]. FCM clusters data samples based on the Euclidean distance among samples, samples within the vicinity of cluster centers are grouped together. This way, each cluster combines locally closer samples and carries characteristic information on a particular region of the Pareto front. This way, depending on the number of clusters used, the CE method will be guided to explore the Pareto boundary more uniformly, as demonstrated by the experimental evaluations below. Fig. 1 exhibits a Pareto boundary generated by the MO CE method using 3 clusters of nondominated solutions, the arrows point possible improvement directions for a maximization problem.

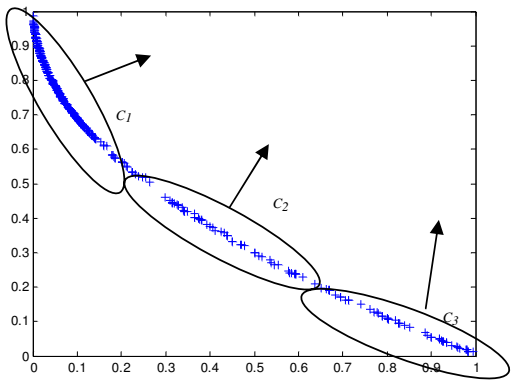


Fig. 1. Nondominated set with 3 clusters.

Based on the above explanations, the presented multiobjective CE (MOCE) strategy can be algorithmically described as follows;

Algorithm I. CE Algorithm for Multiobjective Function Optimization

1. Initialize algorithm parameters, like means (*mu*) and standard deviations (*sigma*) of pdf's, sample size N, and tolerance variable *eps* for each cluster.
2. Generate C×*N* sample vectors X_i, i=1,...,N as:
   $x_{i,j}^{c} = mu + sigma * randn(), i = 1,...,N; j = 1,...,m.$
   $c = 1,...,C.$
   where *randn()* produces Gaussian distributed random numbers and *m* defines the number of object parameter variables. C stands for the number of clusters.
3. Compute sample performances $S(X_i)$, i=1,..., N, within each cluster.
4. For each cluster, determine a subset of best performing samples and update pdf parameters using features, such as mean and standard devation.
5. Check if stopping criterion is met, for example: if maximum iteration count is exceeded or *pdf* parameters are below the tolerance value *eps*. If yes, then output the best-so-far sample and terminate, otherwise go to step 2.

At the end of each iteration, nondominated solutions explored within each cluster are combined together and reclustered before the next iteration starts. This way, distribution of nondominated solutions along the Pareto front become more uniform because solutions found starting from a particular cluster may cross border and belong to neighbor cluster.

IV. EXPERIMENTAL EVALUATIONS

In all experiments the presented method is tested on seven well-known multiobjective benchmark functions listed in Table I. These problems are selected from publications [1]-[12], [21]. Comparisons with the widely known existing methods and related conclusions are discussed below. The labels used for different multiobjective methods within the following illustrations are:

- MOPSO [8]: Multi-objective Particle Swarm Optimization.
- NPGA [7]: The Niched Pareto Genetic Algorithm.
- HLGA [11]: Hajela and Lin's weighted-sum based approach.
- NSGA [4]: The Nondominated Sorting Genetic Algorithm.
- SOEA [21]: A single-objective evolutionary algorithm using weighted-sum aggregation.
- SPEA [24]: The Strength Pareto Evolutionary Algorithm.
- NSGA II [23]: The Nondominated Sorting Genetic Algorithm.
- SPEA II [22]: The Strength Pareto Evolutionary Algorithm.

A key point in the implementation of the CE method is that the population size, hence the number samples generated at each iteration, should be set to a large value in order to effectively explore and extract sufficient number of good performing samples. Otherwise, the CE method would not be guided efficiently enough and focused around optimal or near-optimal solutions. For this purpose, in our tests, *N=population size* is set to 5000. Number of clusters is experimentally set to the number of objectives plus 1. Each experiment is performed over 1000 iterations. Other methods used for comparisons have the following parameter settings [20]:

Number of Generations :250
Population size : 100
Crossover rate : 0.8
Mutation rate : 0.01
Niching parameter : 0.48862
Domination pressure : 10

For each of the benchmark problems under consideration, 10 consecutive executions were made using MOCE and each of other MOO approaches. The results listed below correspond to the best-found solutions related with each approach throughout the 10 experimental trials.

Even though large population size of MOCE seems to be a disadvantage due to computational overhead for fitness evaluations, indeed this is not the case because of the comparable simplicity of generating individuals. In MOCE, individuals are generated through random sampling from an adaptively modified normal distribution, whereas other MOO methods require dedicated recombination and ordering operators for offspring generation.

Performance indices used to compare MOCE and other widely known MOO approaches are Convergence metric (*Cm*), Generational Distance (*GD*) and Diversity metric (*Dm*) [27-28]. The definitions of these three performance measures are as follows:

Cm and *GD* measure the distance between the obtained nondominated front (O_{NDF}) and the Pareto-optimal solutions (P_{Pos}) by using the following equations;

$$Cm = \frac{\sum_{i=1}^{|O_{NDF}|} d_i}{|O_{NDF}|}, \tag{6}$$

$$GD = \frac{\sqrt{\sum_{i=1}^{|O_{NDF}|} d_i^2}}{|O_{NDF}|} \tag{7}$$

where d_i is the Euclidean distance between the solution $i \in O_{NDF}$ and the nearest member of P_{Pos}.

Dm measures the extent of spread achieved among the nondominated solutions with the following expression

$$Dm = \frac{d_f + d_l + \sum_{i=1}^{|O_{NDF}|-1} |d_i + \overline{d}|}{d_f + d_l + (|O_{NDF}| - 1)\overline{d}} \tag{8}$$

where d_i is the Euclidean distance between consecutive solutions in the O_{NDF}, and $\overline{d}$ is the average of these distances. The parameters d_f and d_l represent the Euclidean distances between the extreme solutions of the P_{Pos} and the boundary solutions of the O_{NDF}.

TABLE I
MULTIOBJECTIVE BENCHMARK PROBLEMS USED IN EXPERIMENTAL EVALUATIONS

Problem	Objective Functions
FON (n=3) $x_i \in$ [-4,4] *i=1,...,n*	$f_1(x) = 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{3}}\right)^2\right)$ $f_2(x) = 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{3}}\right)^2\right)$
ZDT1 (n=30) $x_i \in$ [0,1] *i=1,...,n*	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{f_1 / g(x)}]$ $g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i\right) \Big/ (n-1)$
ZDT2 (n=30) $x_i \in$ [0,1] *i=1,...,n*	$f_1(x_1) = x_1$ $f_2(x) = g(x)[1 - (x_1 / g(x))^2]$ $g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i\right) \Big/ (n-1)$ $h(f_1, g) = 1 - (f_1 / g)^2$
ZDT3 (n=30) $x_i \in$ [0,1] *i=1,...,n*	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \sqrt{f_1 / g(x)} - (f_1 / g(x)) \sin(10\pi f_1)\right]$ $g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i\right) \Big/ (n-1)$
ZDT4 (n=30) $x_1 \in$ [0,1] $x_i \in$ [-5,5] *i=2,...,n*	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{f_1 / g(x)}]$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n} x_i^2 - 10\cos(4\pi x_i)$
ZDT6 (n=10) $x_i \in$ [0,1] *i=1,...,n*	$f_1(x_1) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x) = g(x)[1 - (f_1 / g(x))^2]$ $g(x) = 1 + 9[(\sum_{i=2}^{n} x_i)/(n-1)]^{0.25}$
DLZT (n=3) $x_i \in$ [0,1] *i=1,...,n*	$f_1(x) = x_1$ $f_2(x) = x_2$ $f_3(x) = 3.5 - \sum_{i=1}^{n} 2x_i \sin(n\pi x_i)$

Fig.s 2-8 illustrate the nondominated Pareto fronts achieved by the proposed and the other well-known algorithms used for comparisons. In Fig. 2, non-dominated solution sets provided by the proposed algorithm MOCE, MOPSO and NSGA II algorithms when applied to Fonseca and Fleming's nonconvex test function FON with 3 real-valued parameters are presented. According to these results MOCE returned almost equally with MPSO and NSGA II.

In Fig. 3, non-dominated solution sets provided by the proposed algorithm MOCE and selected MOEA algorithms when applied to the convex test function ZDT1 with 30 real parameters are presented. According to Fig. 3, MOCE's

Pareto boundary dominates most of those found by MOPSO and NSGA II and clearly outperformed all the other MOEAs.

Non-dominated solution sets provided by the proposed algorithm MOCE and selected MOEA algorithms when applied to nonconvex test function ZDT2 with 30 real parameters are presented in Fig. 4. As can be seen clearly from this figure, MOCE outperformed all of MOEAs used for comparison in terms of the density along the Pareto front and the solution quality.

Fig. 5 visualizes the comparative results for the discrete test function ZDT3 with 30 real parameters. As seen on the figure, MOCE returned significantly better results than all the other MOEAs. It is found that the solutions found by MOCE dominate all of the solutions found by all the other algorithms.

In Fig. 6, non-dominated solution sets provided by the proposed algorithm MOCE, selected MOEA algorithms when applied to the multimodal test function ZDT4 with 10 real parameters are presented. According to the experimental results, MOCE is the best performing algorithm for this problem instance.

In Fig. 7, non-dominated solution sets provided by the proposed algorithm MOCE and selected MOEA algorithms when applied to non-uniform test function ZDT6 with 10 real parameters are presented. MOCE beated its competitors in terms of solution density and the solution quality.

In Fig. 8, a 3-objective function DLZT is considered and the set of nondominated solutions found by the MOCE presented [25]. Comparing the Pareto boundaries of [27] and MOCE, MOCE clearly performed much better in terms of the density of solutions and solution quality. This example further validates our cluster-based approach for guiding the CE method for multiobjective optimization.

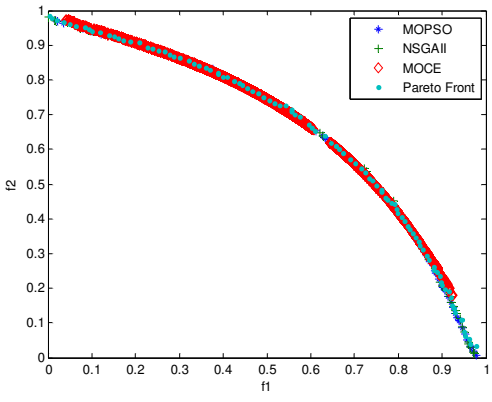
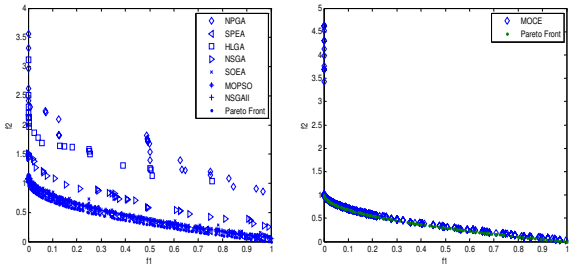
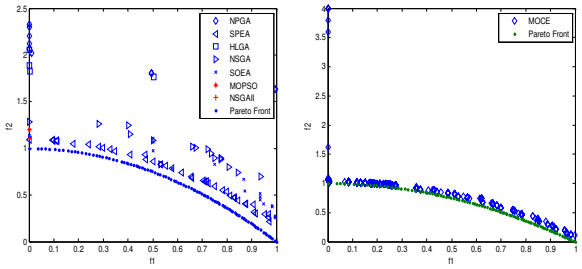


Fig. 2. Pareto fronts for test problem 1 (FON).



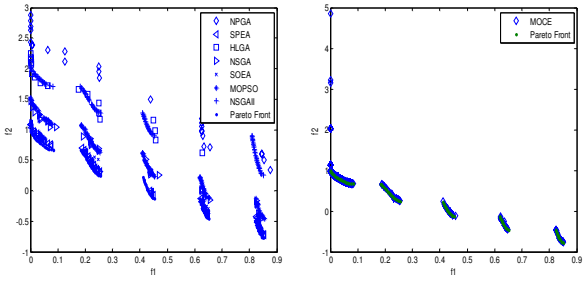
(a) (b)

Fig. 3. Pareto fronts for test problem 2 (ZDT1).



(a) (b)

Fig. 4. Pareto fronts for test problem 3 (ZDT2).



(a) (b)

Fig. 5. Pareto fronts for test problem 4 (ZDT3).

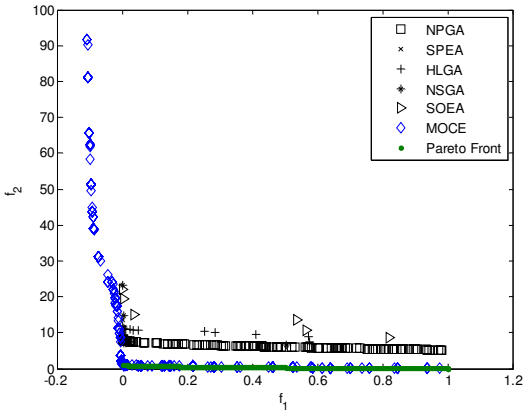
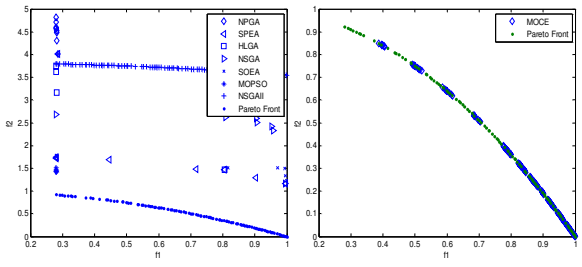
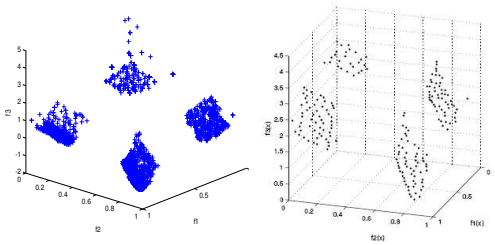


Fig. 6. Pareto fronts for test problem 5 (ZDT4).



(a) (b)

Fig. 7. Pareto fronts for test problem 6 (ZDT6).



(a) (b)

Fig. 8. Pareto fronts for test problem 7 (DLZT).

Tables II to VII illustrates the performance indices for the achieved experimental results in terms of the convergence metric, generational distance and the diversity metric for MOCE and its competitors under consideration.

In terms of the convergence metric *Cm,* MOCE performs much better than all of its competitors for the difficult benchmark problems ZDT4 and ZDT6. For the ZDT1, ZDT2 and ZDT3 problem instances, SPEA performs better than MOCE whereas MOCE achieves better scores than all of the other MOO methods. Finally, MOCE performs better than NSGA for FON benchmark; however MOPSO is slightly better than MOCE in this case.

For the difficult benchmark instance ZDT6, MOCE is still the best performing algorithm in terms of the generational distance *GD*. NPGA is better than MOCE for ZDT3 and ZDT4 problem instances while the other MOO approaches exhibiting comparably poor performance scores. In case of ZDT1 and ZDT2 problems, SPEA exhibits the best performance scores whereas MOCE is the third best performing algorithm. MOCE achieves significantly better results for the FON benchmark.

MOCE is superior over all the other MOO algorithms for all benchmark instances, except ZDT6, in terms of the divergence metric *Dm*. Its score is far below than those of other approaches. In case of ZDT6 benchmark, MOCE and all other MOO algorithms performed almost equivalently.

TABLE II
COMPARISONS USING CONVERGENCE METRIC, GENERATIONAL DISTANCE AND DIVERSITY METRIC FOR PROBLEM FON

	FON		
	Cm	*GD*	*Dm*
MOPSO	**0.0021**	2.7086×10^{-4}	0.5815
NSGAII	0.0036	4.38×10^{-4}	0.4605
MOCE	**0.0026**	$\mathbf{2.7556 \times 10^{-5}}$	**0.2504**

TABLE III
COMPARISONS USIING CONVERGENCE METRIC, GENERATIONAL DISTANCE AND DIVERSITY METRIC FOR PROBLEM ZDT1

	ZDT1		
	Cm	*GD*	*Dm*
NPGA	1.084	0.2299	0.8104
SPEA	**0.039**	**0.0047**	**0.5860**
HLGA	0.8588	0.1994	0.9528
NSGA	0.2762	0.0468	0.7740
SOEA	0.0763	0.0116	0.7455
MOPSO	**0.0640**	**0.0066**	**0.556**
NSGAII	0.0563	**0.0058**	0.4885
MOCE	**0.0434**	**0.0077**	**0.0921**

TABLE IV
COMPARISONS USIING CONVERGENCE METRIC, GENERATIONAL DISTANCE AND DIVERSITY METRIC FOR PROBLEM ZDT2

	ZDT2		
	Cm	*GD*	*Dm*
NPGA	1.0713	0.3259	0.9778
SPEA	**0.0871**	**0.0134**	0.6332
HLGA	0.7429	0.3645	0.8988
NSGA	0.2799	0.0690	0.7925
SOEA	0.1731	0.0414	0.8958
MOPSO	0.2043	0.0681	0.9983
NSGAII	**0.1093**	**0.0109**	1
MOCE	**0.1002**	**0.0160**	**0.1777**

TABLE V
COMPARISONS USIING CONVERGENCE METRIC, GENERATIONAL DISTANCE AND DIVERSITY METRIC FOR PROBLEM ZDT3

	ZDT3		
	Cm	*GD*	*Dm*
NPGA	0.9630	0.2094	0.8133
SPEA	**0.0151**	**0.0023**	0.8031
HLGA	0.7226	0.1750	0.9187
NSGA	0.1449	0.0290	0.8188
SOEA	0.0418	**0.0090**	0.9649
MOPSO	0.1837	0.0211	0.7640
NSGAII	0.5864	0.0613	0.7660
MOCE	**0.0245**	**0.0042**	**0.1686**

TABLE VI
COMPARISONS USIING CONVERGENCE METRIC, GENERATIONAL DISTANCE AND DIVERSITY METRIC FOR PROBLEM ZDT4

	ZDT4		
	Cm	*GD*	*Dm*
NPGA	5.2562	**0.4674**	0.9480
SPEA	11.4482	8.1384	0.9099
HLGA	9.1384	2.5605	0.9936
NSGA	9.4761	3.8618	0.9911
SOEA	13.9416	5.9932	0.7832
MOCE	**1.2056**	**0.5985**	**0.5697**

TABLE VII
COMPARISONS USIING CONVERGENCE METRIC, GENERATIONAL DISTANCE AND DIVERSITY METRIC FOR PROBLEM ZDT6

	ZDT6		
	Cm	*GD*	*Dm*
NPGA	3.5812	1.2688	**0.9613**
SPEA	1.0339	0.4581	**1.0930**
HLGA	2.1547	1.0191	**1.0017**
NSGA	1.5558	0.6054	**0.9897**
SOEA	1.0998	0.4719	1.1751
MOPSO	0.5274	0.1860	**0.9735**
NSGAII	2.8047	0.2805	**0.9376**
MOCE	**0.0029**	**0.0015**	**1.0459**

IV. CONCLUSIONS

This work presents a novel CE strategy for multiobjective optimization. Due to the nature of the CE method, this strategy can be applied for numerical and combinatorial optimization equally well.

The presented strategy clusters the set of nondominated solutions on the Pareto boundary to better guide the CE method and lead uniform distribution of nondominated solutions.

Experimental evaluations using well known benchmark instances and comparisons with well known MOEA's demonstrate that MOCE performs better than its competitors for most of the test cases and it achieved equal success for a few of the others.

Feature research is planed to use this approach for difficult multiobjective combinatorial optimization problems.

REFERENCES

[1] C. A. C. Coello, "A comprehensive survey of evolutionary-based multiobjective optimization", Knowledge and Information Systems, 1(3):269–308, 1999.

[2] K. Deb, and D.E. Goldberg, "An investigation of niche and species formation in genetic function optimization", In Schaffer, J. D., editor, Proceedings of the Third International Conference on Genetic Algorithms, pages 42–50, Morgan Kaufmann, San Mateo, California, 1989.

[3] K. Deb, "Multi-objective genetic algorithms: Problem difficulties and construction of test problems", Evolutionary Computation, 7(3):205–230, 1999.

[4] N. Srinivas and K. Deb. "Multiobjective optimization using nondominated sorting in genetic algorithms", Evolutionary Computation, 2(3):221-248, 1995.

[5] D.E. Goldberg, "Genetic algorithms in search, optimization, and machine learning", Addison-Wesley, Reading, Massachusetts, 1989.

[6] J. Horn, and N. Nafpliotis, "Multiobjective optimization using the niched pareto genetic algorithm", IlliGAL Technical Report 93005, IllinoisGenetic Algorithms Laboratory, University of Illinois, Urbana, Illinois, 1993.

[7] J. Horn, N. Nafpliotis, and D. E. Goldberg," A niched pareto genetic algorithm for multiobjective optimization", In Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Computation, Volume 1, pages 82–87, IEEE Press, Piscataway, New Jersey, 1994.

[8] S. Mostaghim, and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)", Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE, Page(s):26 – 33, 24-26 April 2003.

[9] M. A. Villalobos-Arias, G. T. Pulido, C. A. C. Coello, "A proposal to use stripes to maintain diversity in a multi-objective particle swarm optimizer", Swarm Intelligence Symposium, SIS 2005. Proceedings 2005 IEEE, 2005.

[10] L. B. Zhang, C. G. Zhou, X. H. Liu, Z. Q. Ma, M. Ma, Y. C. Liang, "Solving multi objective optimization problems using particle swarm optimization", Evolutionary Computation, 2003. CEC '03. The 2003 Congress on Volume 4, Page(s):2400 - 2405 Vol.4, 8-12 Dec. 2003.

[11] Hajela, P. and C. Lin, "Genetic search strategies in multicriterion optimal design", Structural Optimization 4, 99–107, 1992.

[12] F. Lin, G. He, "An improved genetic algorithm for multi-objective optimization", Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005. Sixth International Conference on 05-08 Dec. 2005 Page(s):938 – 940, 2005.

[13] G. L. Nemhauser, A. H. G. Rinnooy Kan and M. J. Todd, "Optimization", Chap 10, North Holland, 1989.

[14] X. Hu and R. Eberhart, "Multiobjective optimization using dynamic neighborhood particle swarm optimization", In IEEE Proceedings, World Congress on Computotionol Intelligence (CEC2002), pages 1677-1681, 2002.

[15] C. A. Coello Coello and M. S. Lechuga. "MOPSO: A proposal for multiple objective particle swarm optimization", In IEEE Proceedings, World Congress on Computational Infelligence (CEC2002), pages 1051-1056, 2002.

[16] J. E. Fieldsend and S. Singh, "A multi-objective algorithm based upon particle swarm optimization, an efficient data structure and turbulence", In The 2002 U.K. Workshop on Computational Intelligence, pages 34-44, U.K., 2002.

[17] Rubinstein, R.Y., Kroese, D.P. ,"The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning", Springer-Verlag, New York, 2004.

[18] Rubinstein, R.Y. , "Optimization of computer simulation models with Rare Events", European Journal of Operations Research, **99**, 89-112, 1997

[19] De Boer, P-T., Kroese, D.P, Mannor, S. and Rubinstein, R.Y. , "A tutorial on the cross-entropy method", Annals of Operations Research, **134** (1), 19—67, 2005.

[20] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern classification", John Wiley & Sons, New York, 2001.

[21] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results", Evolutionary Computation 8(2): 173-195, Volume 8, Number 2, 2000.

[22] E. Zitzler, Marco Laumanns, and Lothar Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm", Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001.

[23] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal and T. Meyarivan, "A fast elitist multi-objective genetic algorithm: NSGA-II", Proceedings of the Parallel Problem Solving from Nature VI Conference, pages 849-858, Springer, 2000.

[24] K. Deb, A. Anand, and D. Joshi. "A computationally efficient evolutionary algorithm for real-parameter optimization", Evolutionary Computation Journal, 10(4):371-395, 2002.

[25] K. Deb and D. Saxena. "On finding Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems", Technical Report KanGAL Report No. 2005011, Department of Mechanical Engineering, IIT Kanpur, India, 2005.

[26] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. "Scalable test problems for evolutionary multi-objective optimization", In A. Abraham, L. Jain, and R. Goldberg, editors, Evolutionary Multiobjective Optimization, pages 105-145. London: Springer-Verlag, 2005.

[27] S. Mostaghim and J. Teich. "A new approach on many objective diversity measure". In Dagstuhl Proceedings number 04461, Dagstuhl, Germany, 2005.

[28] Tatsuya Okabe, Yaochu Jin, and Bernhard Sendhoff, "A critical survey of performance indices for multi-objective optimization", Proceedings of IEEE Congress on Evolutionary Computation (CEC-2003). IEEE Press, pages 878-885, 2003.