Volume 4, Issue 1, January 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Multi-Objective Optimization Using Differential Evolution

Er. Anuj Kumar Parashar
Computer Science& Engineering
Agra College, Agra, India

Dr. BDK Patro
Computer Science& Engineering
FET,RBS College, Agra, India

Dr. K Srinivas
Electrical Engineering
DEI, Agra, India

Abstract—In most real world multi-objective optimization problems the objectives are conflicting and therefore, they do not lend themselves to a single solution but result in a set of non-dominating solutions. Several issues arise in Multi-objective Optimization. Firstly, the entire search space has to be searched (in order to find all the good non-dominated solutions) without getting stuck in local optima, secondly, the search should approach the global Pareto-optimal front as closely as possible, thirdly, the search should also ensure a good spread of solutions along the obtained pareto-optimal front and fourthly, it should achieve convergence in a reasonable time. Therefore, the search algorithm has to be carefully designed to address the above mentioned issues. In this work, Elitist Multi-objective Differential Evolution (E-MODE) a new Multi-objective Optimization algorithm is designed and implemented for the solution of real parameter multi-objective function optimization.

Keywords—Differential Evolution, Optimization, multi-Objective

I. INTRODUCTION

This Optimization is the process of finding solutions that minimize or maximize a set of objective functions subject to constraints. When multiple objectives are present, the optimization problem is called a multi-objective optimization (MO) problem. An MO problem can be formally defined as finding a vector of decision variables that satisfies constraints and optimizes a vector function whose elements represent the objective functions [1]. . By using suitably adapted stochastic optimization methods it is possible to reveal the trade- off surface of a multi-objective optimization problem in a single run.[2]. The fact that EAs search from population to population rather than from one individual solution to another, makes them very well suited to performing multi-objective optimization. It is easy to conceive of a population being evolved onto the trade-off surface by a suitably configured EA. In fact, with an appropriate archiving scheme in place, the only modification required to a single objective EA, in order to perform multi-objective optimization, is in the *selection* scheme. As with single objective EAs, a wide variety of multi-objective selection schemes have been devised. Three of the most widely used (and most easily implemented) will be described here.[3]

II. DIFFERENTIAL EVOLUTION

Differential Evolution (DE) is a population-based and directed search method [4], [5]. Like many other evolutionary algorithms, it starts with an initial population vector, which is randomly generated when no a priori knowledge about the solution space is available. Let us assume that $X_{i,G}(i = 1, 2, ...,N_p)$ are candidate solution vectors in the generation G (N_p: population size). Successive populations are generated by adding the weighted difference of two randomly selected vectors to a third randomly selected vector.

Mutation

For each vector $X_{i,G}$ in generation G a mutant vector $V_{i,G}$ is defined by

$$V_{i,G} = X_{a,G} + F(X_{b,G} - X_{c,G}),$$

where $i = \{1, 2, ...,N_p\}$ and a, b, and c are mutually different random integer indices selected from $\{1, 2, ...,N_p\}$. Further, i, a, b, and c are different so that $N_p >= 4$ is required. $F \in [0, 2]$ is a real constant which determines the amplification of the added differential variation of $(X_{b,G} - X_{c,G})$. Larger values for F result higher diversity in the generated population and lower values cause faster convergence.

Crossover

DE utilizes the crossover operation to increase the diversity of the population. It defines the following trial vector:

$$U_{i,G} = (U_{1i},G,U_{2i},G, ...,U_{Di,G}),$$

where D is the problem dimension and

$$U_{ji,G} = \begin{cases} V_{ji,G} \text{ if randj}(0, 1) <= Cr, \\ X_{ji,G} \text{ otherwise.} \end{cases}$$

$C_r \in (0, 1)$ is the predefined crossover rate constant, and randj(0, 1) is the jth valuation of uniform random number generator. Most popular values for C_r are in the range of (0.4, 1) [5].

Selection

The approach that must decide which vector ($U_{i,G}$ or $X_{i,G}$) should be a member of next (new) generation, G + 1. For a maximization problem, the vector with the higher fitness value is chosen. There are other variants based on different mutation strategies [6].

III. PROPOSED ALGORITHM

1. Initialize parameters *t*=0, *NP* =100. (N is number of individuals in population), C and F. Where
   $C \in (0,1)$ is crossover constant
   $F \in (0,2)$ is mutant constant
2. Initialize target population X^t
3. Evaluate each individual *i* in the population and obtain the fitness values corresponding to the respective objective functions.
4. Obtain the best_front_so_far from the current population.
5. Obtain the mutant population (a mutant individual, $V_i^{t+1} = [v_{i1}^{t+1}, v_{i2}^{t+1}, ..., v_{in}^{t+1}]$ is determined such that $V_i^{t+1} = Ceil(X_{best}^t + F(X_{bi}^t - X_{ci}^t))$ where X_{best}^t _front_individual is an individual randomly selected from the best_front_so_far so far and b_i, and c_i are two randomly chosen individuals from the population such that $(b_i \neq c_i)$.
6. Obtain the trial population (For each mutant individual, $V_i^{t+1} = [v_{i1}^{t+1}, v_{i2}^{t+1}, ..., v_{in}^{t+1}]$ an integer random number between 1 and *n*, i.e., $D_i \in (1,2,...,n)$, is chosen, and a trial individual, $U^{t+1} = [U_1^{t+1}, U_2^{t+1}, ..., U_{NP}^{t+1}]$ is generated such that:
   $$u_{ij}^{t+1} = \begin{cases} v_{ij}^{t+1}, if \ r_{ij}^{t+1} \leq CR \ \ or \ \ j = D_i \\ x_{ij}^t, \ \ otherwise \end{cases}$$
   where the index D refers to a randomly chosen dimension (*j=1,2,..,n*))
7. Evaluate trial population
8. Obtain the best_front_so_far from the combined population of Parent population and trial population.
9. Selection procedure for evolving the new parent population P(t+1)
   i. If the number of solutions in the best_front_so_far is equal to N then put all these solutions in the new population P(t+1).
   ii. If the number of solutions in the best_front_so_far is greater than N then select N solutions from this set using Crowding Distance Metric.
   iii. If the number of solutions in the best_front_so_far is less than N then put all the solutions of this set in the new population P(t+1) and select remaining members of the new population using the following procedure:
   (i) Mark those parents who have not made it to the new population P(t+1).
   For each parent marked in step (i).The selection is based on the survival of the fittest among the trial population and target population such that:
   $$X_i^{t+1} = \begin{cases} U_i^{t+1}, \ \ if \ (f_k(U_i^{t+1}) \geq f_k(X_i^t)) \ for k=1 to m \\ X_i^t, \ \ otherwise \end{cases}$$
10. Update the external archive of non-dominated solutions using the current best_front_so_far.
11. Repeat steps 3 to 9 while termination condition not reached.

IV. TEST FUNCTIONS

The performance of E-MODE was evaluated on six benchmark problems proposed by Zitzler *et al.* [7] namely ZDT1, ZDT2and ZDT3.

ZDT1

$F_1(X) = X_1$
$F_2(X) = G(X) [1- \sqrt{} (X_1/G(X))]$
$G(X) = 1+ 9/ (NP-1) \sum X_i$

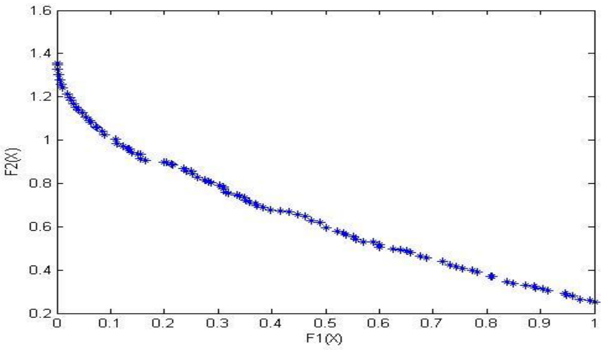
ZDT2

$F_1(X) = X_1$
$F_2(X) = G(X) [1- (X_1/G(X))^2]$
$G(X) = 1+ 9/ (NP-1) \sum X_i$

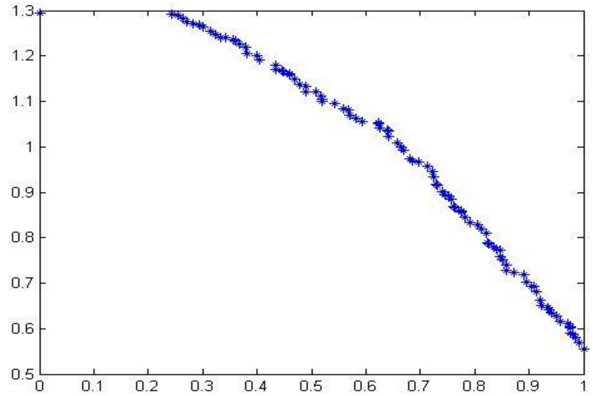
ZDT3

$F_1(X) = X_1$
$F_2(X) = G(X) [(1-\sqrt{} (X_1/G(X))) – (X_1/G(X)) \text{ Sin } (10\pi X_1)]$ $G(X) = 1+ 9/ (NP-1) \sum X_i$

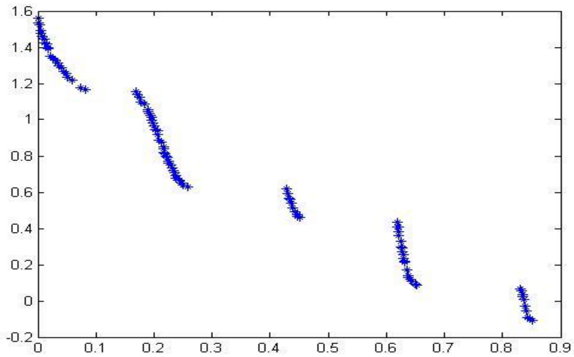
V. RESULTS



Simulation Results of ZDT1



Simulation Results of ZDT2



Simulation Results of ZDT3

VI. CONCLUSION AND FUTURE WORK

For the optimization of real-world problems it is important that the applied algorithm is capable of handling multiple objective functions and several constraints. In this work Differentia Evolution algorithm is extended to a new version of Multi-objective Differential Evolution (E-Mode) based on a non-dominant sorting and crowding distance.

In this work, Elitist Multi-objective Differential Evolution (E-MODE) a new Multi-objective Optimization algorithm is designed and implemented for the solution of real parameter multi-objective function optimization. E-MODE extends Differential Evolution to deal with multi-objective optimization. It incorporates many important novel features like Pareto dominance, elitism, and crowded distance metric for ensuring a good spread of solutions and reducing the number of non-dominated solutions in the external archive of non-dominated solutions.. E-MODE algorithm is implemented in Matlab 7.0. The performance of the algorithm is tested on some standard multi-objective test functions. The obtained results indicate good performance of the algorithm in terms of obtaining quality solutions (global Pareto-optimal solutions with good spread along the Pareto-optimal curve), and fast convergence.

REFERENCES

1. Deb, K., (2001) Multi-objective Optimization Using Evolutionary Algorithms John Wiley & Sons Ltd., New York, NY.
2. Engrand, P., (1997) "A Multi-Objective Approach Based on Simulated Annealing and its Application to Nuclear Fuel Management", publication Proc. 5th Int. Conf. Nuclear Engineering , Nice, ICONE5-2523.

3. Baker, J.E., (1985) "Adaptive Selection Methods for Genetic Algorithms", 101-111 in Pro- Proceedings of an International Conference on Genetic Algorithms and their Applications proceedings (J.J. Grefenstette, editor), Lawrence Erlbaum Associates, Hillsdale, NJ.
4. R. Storn and K. Price, Differential Evolution- A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, Journal of Global Optimization 11, pp. 341-359, 1997
5. G.C. Onwubolu and B.V. Babu, New Optimization Techniques in Engineering, Berlin ; New York : Springer, 2004.
6. 4. S. Das, A. Konar, U.K. Chakraborty, Two Improved Differential Evolution Schemes for Faster Global Search, Proceedings of the 2005 conference on Genetic and evolutionary computation, pp. 991-998, Washington, USA, 2005.
7. E. Zitzler, K. Deb, L. Thiele, Comparison of multi-objective evolutionary algorithms: Empirical results. Evolutionary Computation (2000) 173 -195.

