A Novel Hybrid Fuzzy Multi-Objective Evolutionary Algorithm: HFMOEA

Amit Saraswat* and Ashish Saini

Department of Electrical Engineering,
Dayalbagh Educational Institute, Agra 282110, Uttar Pradesh, India
amitsaras@gmail.com, ashish_711@rediffmail.com

Abstract. This paper presents a development of a new hybrid fuzzy multi-objective evolutionary algorithm (HFMOEA) for solving complex multi-objective optimization problems. In this proposed algorithm, two significant parameters such as crossover probability (P_C) and mutation probability (P_M) are dynamically varied during optimization based on the output of a fuzzy controller for improving its convergence performance by guiding the direction of stochastic search to reach near the true pareto-optimal solution effectively. The performance of HFMOEA is examined and compared with NSGA-II on three benchmark test problems such as ZDT1, ZDT2 and ZDT3.

Keywords: Multi-objective evolutionary algorithms, fuzzy logic controller, global optimal solution, pareto-optimal front.

1 Introduction

In last couple of decades, a number of multi-objective evolutionary algorithms (MOEAs) have been suggested for solving complex multi-objective problems [1]-[3]. The main purpose behind the development of the MOEA approach is that it has ability to find multiple Pareto-optimal solutions in one single simulation run. The non-dominated sorting genetic algorithm (NSGA) proposed in [1] was one of the first such EAs. Over the years, NSGA was criticized in [2] on the basis of some aspects such as high computational complexity of non-dominated sorting, lack of elitism and need for specifying the sharing parameter. In reference [3], an improved version of NSGA called as NSGA-II. Two other contemporary MOEAs: Pareto-archived evolution strategy (PAES) [4] and strength Pareto EA (SPEA) [5] were also reported in the literature. In reference [6], a survey of different multi-objective evolutionary and real coded genetic algorithms is presented.

In present paper, a new Hybrid Fuzzy Multi-Objective Evolutionary Algorithm (HFMOEA) has been proposed for solving complex multi-objective problems. In proposed HFMOEA, a fuzzy logic controller (*FLC_HMOEA*) is developed, which would cause variation in two HFMOEA parameters such as crossover probability (P_C) and mutation probability (P_M) dynamically during optimization process after each k

* Corresponding author.

N. Meghanathan et al. (Eds.): CCSIT 2012, Part III, LNICST 86, pp. 168–177, 2012.

number of iterations. These parameter variations provide HFMOEA a kind of adaptability to the nature of targeted optimization problem and help to reach the near global optimal solutions and hence arrive near to *true pareto-optimal front*. The performance of HFMOEA is examined on three benchmark test problems such as ZDT1, ZDT2 and ZDT3.

2 Proposed HFMOEA for Solving Multi-Objective Problems

The flowchart of proposed HFMOEA for solution of complex multi-objective problems is outlined in Fig.1. Details of proposed algorithm are discussed as below:

Initialization: HFMOEA based optimization starts with initialization of various input parameters of HFMOEA such as population size (*popsize*), maximum numbers of iterations (*max_iteration*), number of control variables, system constraints limits, crossover probability (P_C), mutation probability (P_M) etc.

Generation of Initial population: it is generated randomly according to following procedural steps:

- Step 1: Generate a string of real valued random numbers within their given variable limits to form a single individual;
- Step 2: Place the individual as valid individual in initial population;
- Step 3: Evaluate fitness value for valid individual;
- Step 4: Check if the initial population has not completed then go to step 1;

Non-Domination Sorting: The generated initial population is sorted on the basis on non-domination sorting algorithm proposed by Deb [2].

For producing the new population for next iteration, the following operators are applied to parent population:

Selection: The Binary Tournament selection as proposed in reference [2] is used as a selection operator for reproducing the mating pool of parent individuals for crossover and mutation operations.

Crossover: The BLX-α crossover as proposed in reference is applied on randomly selected pairs of parent individuals $\left(x_i^{(1,t)}, x_i^{(2,t)}\right)$ with a crossover probability $\left(P_C\right)$ which is a combination of an extrapolation/interpolation method.

Mutation: The PCA based Mutation as proposed in reference [7] with mutation probability $\left(P_m\right)$ is applied to generate the offspring population.

Criterion to prepare population for next iteration: After the execution of above genetic operators, offspring population is checked to prepare new population for next iteration by going through following procedural step:

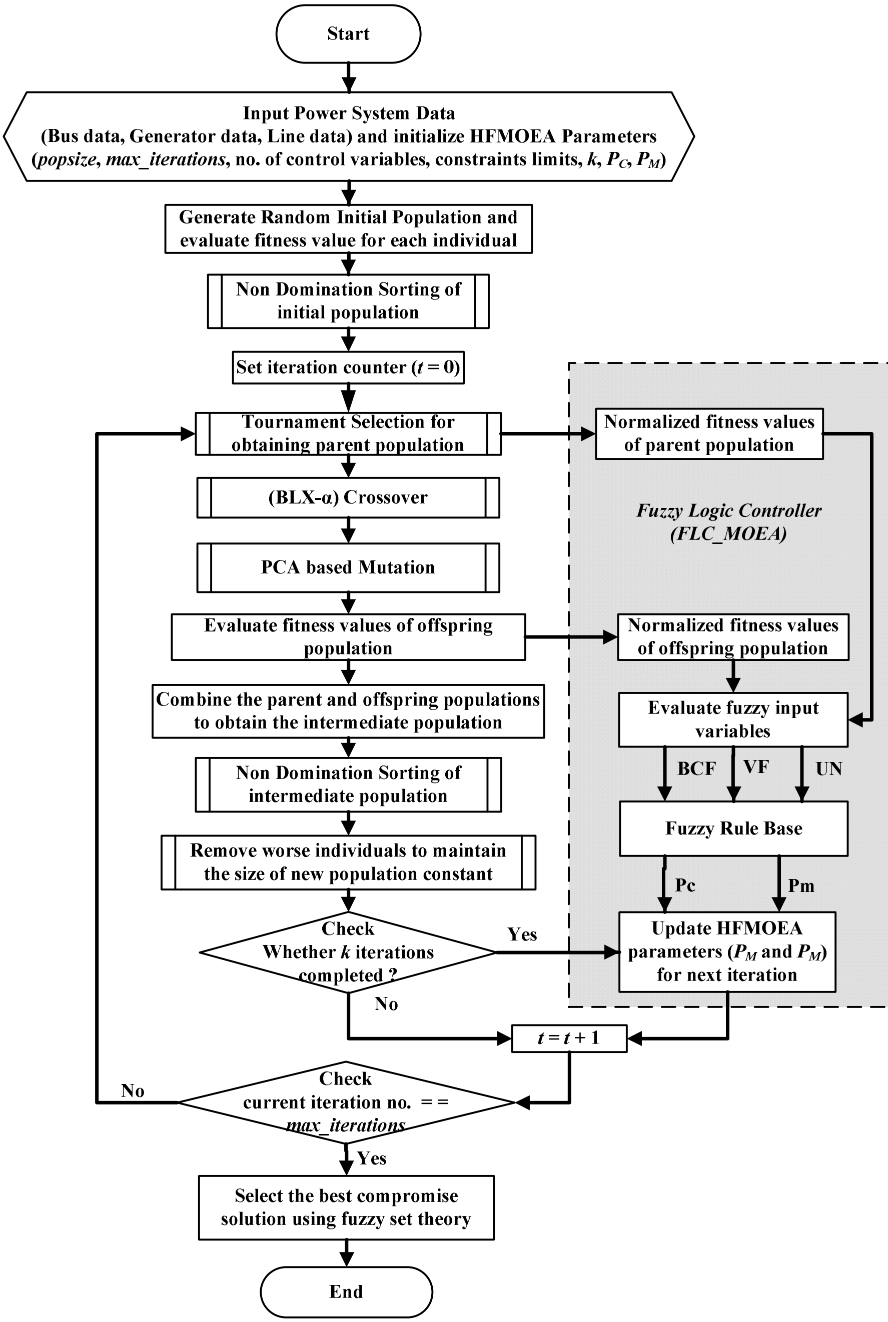


Fig. 1. Flowchart for Hybrid Fuzzy Multi-Objective Evolutionary Algorithm (HFMOEA)

Step 1: Evaluate the fitness values for each individual in offspring population;
Step 2: Combine the parent and offspring population to obtain the intermediate population;
Step 3: Perform the non-domination sorting algorithm on intermediate population;
Step 4: Remove the worse individuals to maintain the new population size constant. Here the new population for next iteration is prepared;
Step 5: Check if k^{th} iterations (let k = 10) has completed go to next step 6 otherwise go to step 7.
Step 6: Update HFMOEA parameters (i.e. P_C and P_M) by using fuzzy logic controller (*FLC_HMOEA*).
Step 7: Check the termination condition of HFMOEA. i.e. if the current iteration number is equal to *max_iterations*, terminate the iteration process otherwise go to next iteration.
Step 8: Select the best compromise solution using fuzzy set theory.

Best compromise solution: Upon having the Pareto-optimal set of non-dominated solution using proposed HFMOEA, an approach proposed in [8] selects one solution to the decision maker as the best compromise solution [9]. This approach suggests that due to imprecise nature of the decision maker's judgment, the i^{th} objective function F_i is represented by a membership function μ_i defined as in reference [8]:

$$\mu_i = \begin{cases} 1 & F_i \le F_i^{\min} \\ \dfrac{F_i^{\max} - F_i}{F_i^{\max} - F_i^{\min}} & F_i^{\min} < F_i < F_i^{\max} \\ 0 & F_i \ge F_i^{\min} \end{cases} \tag{1}$$

Where $F_i^{\min}$ and $F_i^{\max}$ are the minimum and maximum values of the i^{th} objective function among all non-dominated solutions, respectively. For each j^{th} non-dominated solution, the normalized membership function μ^j is calculated as:

$$\mu^j = \frac{\sum_{i=1}^{N_{obj}} \mu_i^j}{\sum_{j=1}^{N_{dom}} \sum_{i=1}^{N_{obj}} \mu_i^j} \tag{2}$$

Where N_{dom} is the number of non-dominated solutions. The best compromise solution is that having the maximum value of μ^j .

Fitness function: The fitness function corresponding to each individual in the population is assigned based on their respective generalized augmented functions as evaluated in equation (3). Thus the fitness function (H_m) for m^{th} objective is evaluated as:

$$H_m = \frac{K_m}{1 + f_{obj,m}}; \qquad \forall m = 1 : N_{obj} \tag{3}$$

Where N_{obj} is the total number of objectives and K_m is the appropriate constant corresponding to m^{th} objective, in this paper.

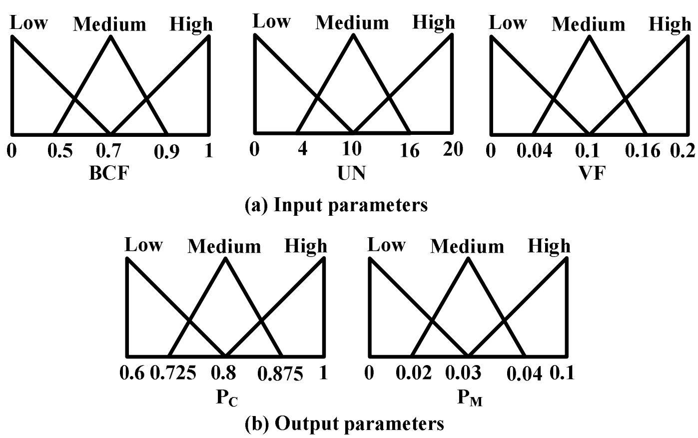


Fig. 2. Input and Output membership functions for Fuzzy Logic Controller

Fuzzy logic controller: It has been experienced that after few iterations, the fitness values of each of the individuals are becoming equal to other individuals in same population and hence the effect of crossover operator beyond that stage becomes insignificant due to lack of diversity. Therefore, the increased mutation probability (P_M) remains the only alternative to produce the better offspring for achieving a more diversified population. A fuzzy logic controller (*FLC_HMOEA*) is designed to vary P_C and P_M dynamically during the optimization process. These parameters (P_C and P_M) are varied based on the fitness function values as per following logic given in reference [10]:

- Ideally the best compromized fitness (BCF) using (1)-(3) should change for each iteration, but if no significant change take places over a number of iterations (UN) then the values of both P_C and P_M must be modified.
- In multi-objecteive problems, the diversity in population supports the stochastic search to reach the perato-optimal fronts. The variance of the fitness values of objective function (VF) of a population is a measure of diversity in population. Hence, it is considered as another factor on which both P_C and P_M may be changed.

Thus the ranges of three input fuzzy parameters such BCF, UN and VF and also two output fuzzy parameters such as P_C and P_M are repersented by three lingustic terms as LOW, MEDIUM and HIGH. The details of membership functions for input and output variables of *FLC_HMOEA* are shown in Fig. 2.

Fuzzy Rule Base for HFMOEA

1. If (BCF is Low) then (Pc is High) (Pm is Low) (1)
2. If (BCF is Medium) and (UN is Low) then (Pc is High) (Pm is Low) (1)
3. If (BCF is High) and (UN is Low) then (Pc is High) (Pm is Low) (1)
4. If (BCF is Medium) and (UN is Medium) then (Pc is Medium) (Pm is Medium) (1)
5. If (BCF is High) and (UN is Medium) then (Pc is Medium) (1)
6. If (UN is High) and (VF is Low) then (Pc is Low) (Pm is High) (1)
7. If (UN is High) and (VF is Medium) then (Pc is Low) (1)
8. If (UN is High) and (VF is High) then (Pc is Medium) (1)
9. If (BCF is High) and (VF is Medium) then (Pm is Low) (1)
10. If (BCF is High) and (VF is High) then (Pm is Low) (1)
11. If (VF is High) then (Pc is High) (Pm is Low) (1)
12. If (VF is Medium) then (Pc is High) (Pm is Low) (1)
13. If (BCF is High) and (VF is Low) then (Pc is High) (Pm is Low) (1)
14. If (BCF is Medium) and (VF is Medium) then (Pc is Low) (Pm is High) (1)
15. If (BCF is Low) and (UN is Low) and (VF is Low) then (Pc is High) (Pm is Low)

3 Simulation Results

The proposed HFMOEA is implemented according to the procedure explained in previous sections and all the simulations are carried out in MATLAB 7.0 programming environment on Pentium IV 2.27 GHz, 2.0 GB RAM computer system. In present case study, the proposed HFMOEA is examined and compared with a popular multi-objective evolutionary algorithm i.e. NSGA-II presented in reference [2]. The detailed specifications of both NSGA-II and HFMOEA are summarized in Table 1. The NSGA-II comprises a simulated binary crossover (SBX) operator and a polynomial mutation [11] like real coded GAs. For real-coded NSGA-II, distribution indexes [11] as $\eta_c = 20, \textit{and } \eta_m = 20$ are used for crossover and mutation operators respectively (see Table 1). Whereas in HFMOEA, a BLX-α crossover and PCA-mutation [7] operators are used with dynamically varying after each 10 iterations with probabilities (P_C and P_M) based on fuzzy logic controller (*FLC_HMOEA*) as described in section 2.

Three benchmark test problems such as ZDT1, ZDT2 and ZDT3 out of six as suggested by Zitzler, Deb and Thiele [12] are taken for testing and comparison of proposed HFMOEA. In this paper, the whole simulation is divided into four cases such that in each case, both the algorithms (NSGA-II and HFMOEA) are evaluated for deferent population sizes and number of maximum iterations. Thus, the Population size and number of maximum iterations are taken as (100 and 300), (100 and 500), (200 and 500) and (300 and 500) in Case: 1, Case: 2, Case: 3 and Case: 4 respectively (see Table 2). For all three test problems, the best compromised solutions obtained after optimization using NSGA-II and HFMOEA are summarized in Table 2. The best compromised solution is calculated according to (1) and (2) described in previous section.

The pareto-optimal fronts obtained by NSGA-II and HFMOEA for ZDT1 test problem in all four cases are depicted in Fig.3. It has been observed that NSGA-II could not be fully converged in case 1 and Case: 2 when the population size is 100 and maximum numbers of iterations are 300 and 500 respectively. While, proposed HFMOEA has been converged and able to achieve near global pareto-optimal front even in case: 1 (see Fig.3). Similar investigations are conducted on other to benchmark test problems such as ZDT2 and ZDT3 for Case: 1, Case: 2 and Case: 3, the pareto-optimal fronts are obtained shown in Fig.4 and Fig.5, respectively. During the execution of optimization based on proposed HFMOEA, it's two parameters such as crossover probability (P_C) and mutation probability (P_M) are varied dynamically after each ten iterations. These variations are taken place based on the output of fuzzy controller (FLC_HMOEA) as described in section 2. The variations in P_C and P_M for all three test problems (ZDT1, ZDT2 and ZDT3) in Case: 3 are shown in Fig.6. It has been observed that the variations in crossover and mutation probabilities are such that if P_C is going to reduce, P_M will increase (see Fig.6.). These variations in parameters are helping the HFMOEA in searching the global optimal solutions. Therefore, this property will enhance the capability of HFMOEA to achieve the near global pareto-optimal front.

Table 1. Specifications of optimization algorithms

Algorithm Parameters	**NSGA-II**	**HFMOEA**
Selection operator	Tournament	Tournament
Crossover operator	Simulated Binary (SBX)	BLX-α crossover
Mutation operator	polynomial mutation	PCA mutation
Crossover probability (P_C)	0.9	varying based on FLC output
Mutation probability (P_M)	$\eta_c = 20, \textit{and } \eta_m = 20$	varying based on FLC output

Table 2. Best Compromised solutions obtained by NSGA-II and HFMOEA

Test Case	**Pop. Size**	**Max. Iterations**	**optimization Algorithm**	**Best Compromised Solution after optimization**					
				Test Problem : ZDT1		**Test Problem : ZDT2**		**Test Problem : ZDT3**	
				f1(x)	**f2(x)**	**f1(x)**	**f2(x)**	**f1(x)**	**f2(x)**
Case:1	100	300	NSGA-II	0.1557	0.9129	0	1.2949	0.2505	0.6376
			HFMOEA	0.2838	0.4693	1	0.002	0.2507	0.2579
Case:2	100	500	NSGA-II	0.2593	0.6503	1	0.2403	0.2503	0.4431
			HFMOEA	0.2543	0.4967	1	0.0011	0.2485	0.2565
Case:3	200	500	NSGA-II	0.2236	0.5951	0	1.053	0.2485	0.3706
			HFMOEA	0.259	0.4921	1	0.0007	0.2507	0.2519
Case:4	300	500	NSGA-II	0.2354	0.5543	1	0.0803	0.2498	0.3071
			HFMOEA	0.2614	0.4893	0	1.0001	0.2495	0.2529

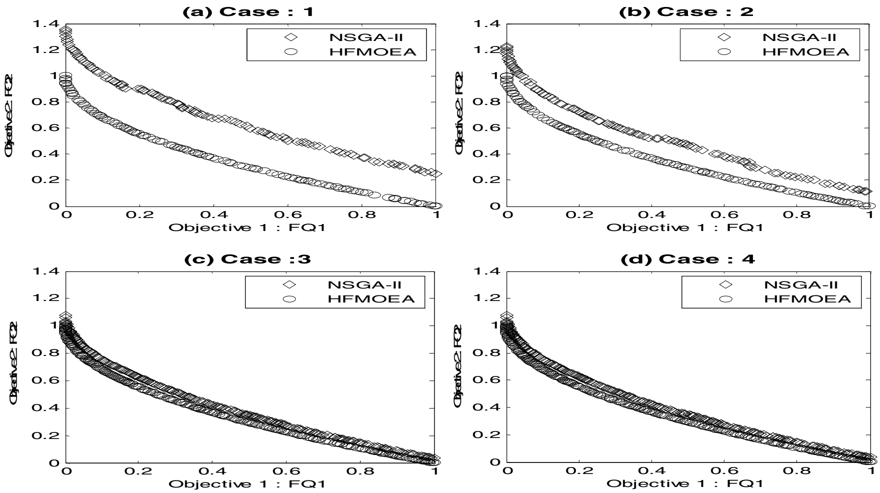


Fig. 3. Comparison of Pareto-optimal fronts obtained using NSGA-II and HFMOEA for ZDT1 test problem for different population sizes after various iterations

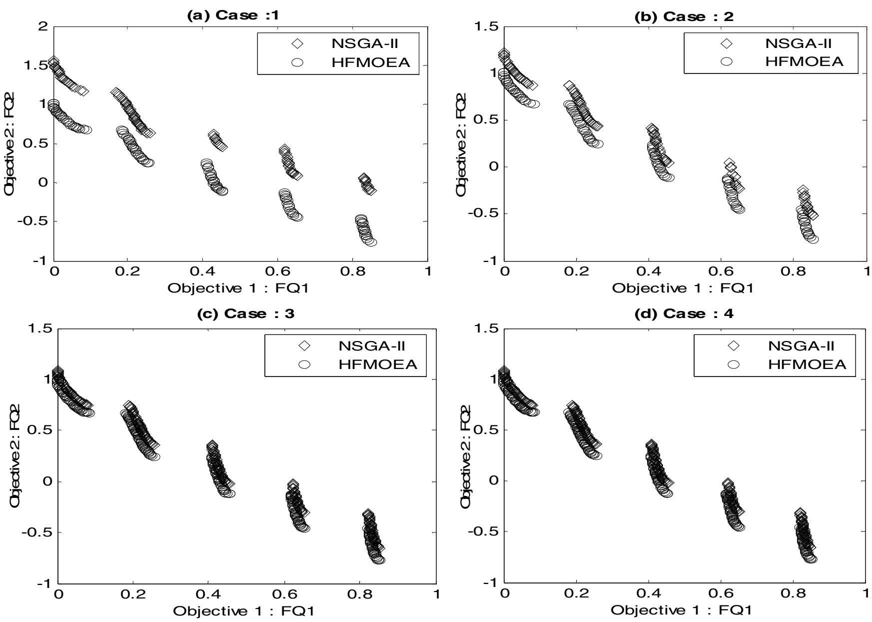


Fig. 4. Comparison of Pareto-optimal fronts obtained using NSGA-II and HFMOEA for ZDT2 test problem for different population sizes after various iterations

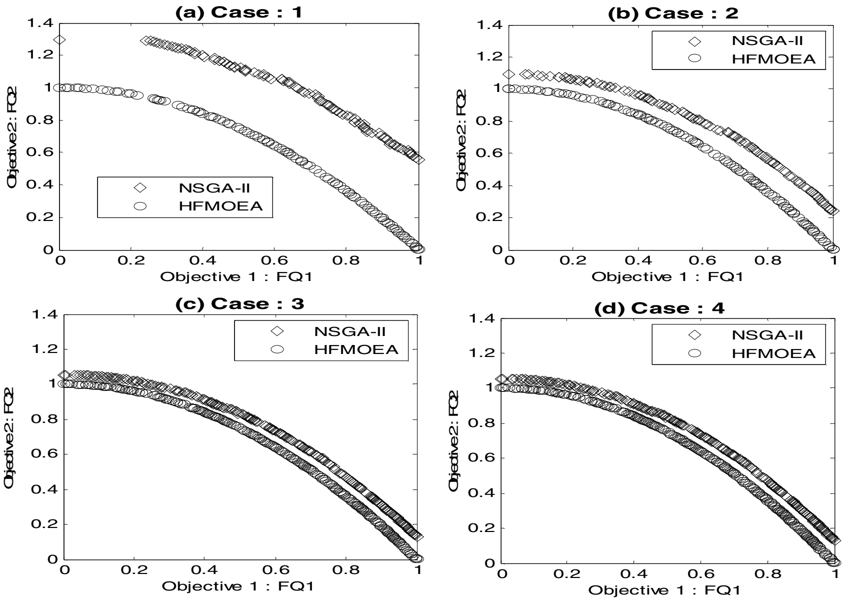
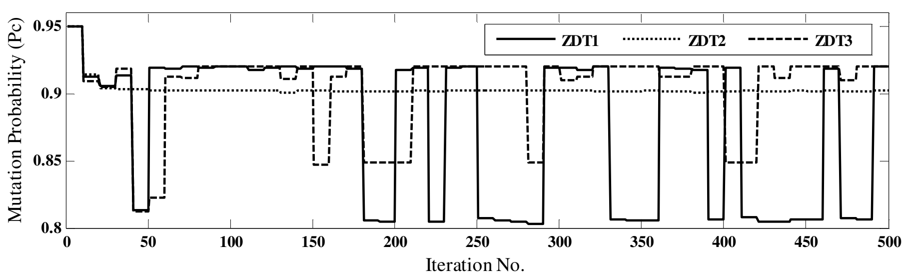
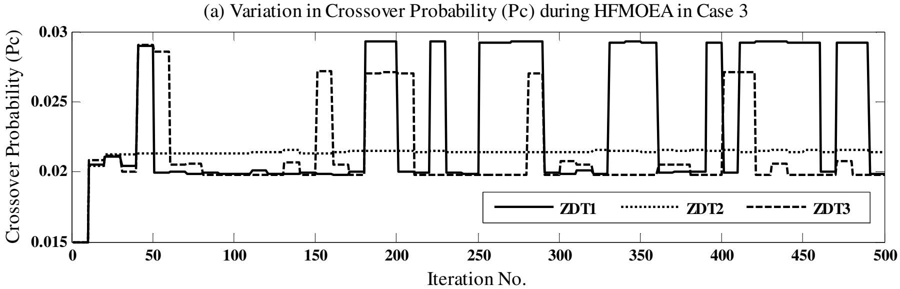


Fig. 5. Comparison of Pareto-optimal fronts obtained using NSGA-II and HFMOEA for ZDT3 test problem for different population sizes after various iterations



(a) Variation in Crossover Probability (Pc) during HFMOEA in Case 3



(b) Variation in Mutation Probability (Pm) during HFMOEA in Case 3

Fig. 6. Variations P_C and P_M during optimization based on HFMOEA for three test problems

4 Conclusion

A fuzzy logic controller called as FLC_HMOEA has been developed and successfully applied in a proposed multi-objective optimization algorithm i.e. HFMOEA. This implementation returns the advantage in terms of improvement in the performance of HFMOEA i.e. good convergence with better quality of the pareto-optimal solutions and consequently arrives to a near pareto-optimal front. FLC_HMOEA helps in guiding the direction of stochastic search to reach the near global optimal solution effectively. HFMOEA has been tested on three benchmark test problems such as ZDT1, ZDT2 and ZDT3 and compared with NSGA-II. The simulation results revealed the effectiveness of HFMOEA for solving multi-objective problems.

References

1. Srinivas, N., Deb, K.: Multi-objective Optimization Using Non-dominated Sorting in Genetic Algorithms. Evolutionary Computation 2(3), 221–248 (1994)
2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast Elitist Multi-objective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
3. Zitzler, E., Thiele, L.: Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)
4. Knowles, J., Corne, D.: The Pareto archived evolution strategy: A new baseline algorithm for multi-objective optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation, pp. 98–105. IEEE Press, Piscataway (1999)
5. Zitzler, E.: Evolutionary algorithms for multiobjective optimization: Methods and applications, Doctoral dissertation ETH 13398. Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (1999)
6. Raghuwanshi, M.M., Kakde, O.G.: Survey on multi-objective evolutionary and real coded genetic algorithms. In: Proceedings of the 8th Asia Pacific Symposium on Intelligent and Evolutionary Systems, pp. 150–161 (2004)
7. Saraswat, A., Saini, A.: Optimal reactive power dispatch by an improved real coded genetic algorithm with PCA mutation. In: Proceedings of Second International Conference on Sustainable Energy and Intelligent System (IET SEISCON 2011), vol. 2, pp. 310–315 (2011)
8. Dhillon, J.S., Parti, S.C., Kothari, D.P.: Stochastic economic emission load dispatch. Electric Power Systems Research 26, 179–186 (1993)
9. Abido, M.A., Bakhashwain, J.M.: Optimal VAR dispatch using a multi-objective evolutionary algorithm. Electric Power and Energy Systems 27(1), 13–20 (2005)
10. Saini, A., Chaturvedi, D.K., Saxena, A.K.: Optimal power flow solution: a GA-Fuzzy system approach. International Journal of Emerging Electric Power Systems 5(2) (2006)
11. Deb, K., Agarwal, R.B.: Simulated Binary Crossover for Continuous Search Space. Complex Systems 9, 115–148 (1995)
12. Zitzler, E., Deb, K., Thiele, L.: Comparison of multi-objective evolutionary algorithms: Empirical results. Evolutionary Computing 8(2), 173–195 (2000)