2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Fuzzy Multiobjective Differential Evolution Using Performance Metrics Feedback

Chatkaew Jariyatantiwait Gary G. Yen
Oklahoma State University
School of Electrical and Computer Engineering

***Abstract*—Differential evolution is regarded as one of the most efficient evolutionary algorithms to tackle multiobjective optimization problems. The key to success of any multiobjective evolutionary algorithms (MOEAs) is maintaining a delicate balance between exploration and exploitation throughout the evolution process. In this paper, we propose a Fuzzy-based Multiobjective Differential Evolution (FMDE) that uses performance metrics, specifically, hypervolume, spacing, and maximum spread, to measure the state of the evolution process. We apply the inference rules to these metrics in order to dynamically adjust the associated control parameters of a chosen mutation strategy used in this algorithm. One parameter controls the degree of greedy or exploitation, while another regulates the degree of diversity or exploration of the reproduction phase. Therefore, we can appropriately adjust the degree of exploration and exploitation through performance feedback. The performance of FMDE is evaluated on well-known ZDT and DTLZ test suites in addition two representative functions in WFG. The results show that the proposed algorithm is competitive with respect to chosen state-of-the-art MOEAs.**

***Keywords*—Multiobjective differential evolution, fuzzy logic, performance metrics, hypervolume, spacing, maximum spread.**

I. INTRODUCTION

Differential Evolution (DE) was proposed by Storn and Price in 1995 as a new evolutionary algorithm (EA) [1-2]. It is a stochastic population-based search approach for optimization over the continuous space. DE is one of the most powerful tools for solving optimization problems. DE can handle mixed-type variables, constraints, multimodality and also multiple-objective. Implementing DE is easier than other EAs such as genetic algorithm (GA) even for the beginners in the optimization field. In addition, number of control parameters is very few. DE is similar to other evolutionary algorithms, as it starts with the randomly initializing population in the search space. Then the population enters the evolution loop: mutation, crossover, and selection operations. These three operations will be repeated until the stopping criterion is met. However, the difference between DE and other EAs is the powerful use of differences between individuals realized by differential mutation which makes DE unique.

The mutation strategy and the control parameters, namely, scaling factor (F), crossover rate (CR), and population size (NP), play the major roles in the success of DE. Choosing the appropriate mutation operator and parameter values for a particular problem is a difficult task because it is problem dependent and time-consuming trial and error process. Therefore, among multiobjective DE algorithms references [3-6] proposed the adaptive control parameter setting during the search process.

In addition, balancing the exploration and exploitation throughout the search is the key to the success of an EA. During the evolution process we may need different mutation strategy and parameter values. In the beginning of the evolution we need higher degree of exploration than exploitation in order to search larger regions in the space. We may choose the mutation operator that possesses high exploration ability and the control parameter that promote the diversity. However, near the end of the evolution we need to emphasize on the local search that is exploitation. The mutation operator that favors local search is chosen along with the control parameters that emphasize the exploitation. If we know the state of the evolution process, we may decide whether we should emphasize on exploration or exploitation, and choose suitable parameter values or the mutation strategies. One possible way that we can observe the status of the evolving process is utilizing the performance metrics. Most of the performance metrics are calculated at the end of the evolution in order to assess the quality of the obtained nondominated front. For instance, generational distance needs the complete knowledge about the true Pareto front in order for calculation. We cannot assume the true Pareto front is available during the evolution search. The quality of the population can be measured by three properties of the obtained nondominated front [7], namely, the convergence, uniform distribution, and extensiveness. Although there are some proposed running performance metrics [8] to measure the quality of the population, there are very few choices to allow us to measure the convergence, uniform distribution, and extensiveness of the population. Hence, we exploit three performance metrics, namely, hypervolume, spacing, and maximum spread to measure the three properties of the obtained nondominated solutions. The proposed multiobjective differential evolution utilizes hypervolume, spacing, and maximum spread as the input to the fuzzy rules that adaptively adjust the control parameters for the mutation scheme which is the greedy factor and the diversity factor every generation in order to balance the exploration and exploitation abilities of the population during the search process.

The rest of this paper is organized as follows. Section II describes the background knowledge of DE and gathers some related works presenting the adaptive multiobjective DE and some state-of-the-art MOEAs. Then, Section III introduces the proposed fuzzy multiobjective DE using performance metrics feedback. Section IV presents the experimental setups and results. Finally, Section V states the conclusion of our work and the future research.

II. RELATED WORKS

A. Background

For single objective optimization problem, we search for the best possible solution. However, for multiobjective optimization, some objective functions conflict with the others. We then search for a set of Pareto optimal solutions, not a single optimal one. The concepts of "*Pareto Dominance*" and "*Pareto Optimality*" are employed in order to obtain the set of optimal solutions. Without loss of generosity, consider the multiobjective minimization problem (MOPs):

$$\min_{\mathbf{x}\in\Re^D} \mathbf{F}(\mathbf{x}) = [F_1(\mathbf{x}), F_2(\mathbf{x}), ..., F_k(\mathbf{x})]. \qquad (1)$$

A decision variable $\mathbf{x}_a$ is said to dominate another decision vector $\mathbf{x}_b$, denoted by $\mathbf{x}_a \prec \mathbf{x}_b$ *iff* $F_i(\mathbf{x}_a) \le F_i(\mathbf{x}_b)$ for all i = 1,2, 3,…,k and $F_i(\mathbf{x}_a) < F_i(\mathbf{x}_b)$ for at least one $i \in \{1,2,3,…,k\}$. A decision variable $\mathbf{x}^*$ is said to be the Pareto optimal *iff* there exist no $\mathbf{x}$ in the decision space such that $\mathbf{x} \prec \mathbf{x}^*$. The set of such Pareto optimal vectors is known as the *Pareto optimal set* and the corresponding curve in the objective space that obtained from the Pareto optimal set is called the *Pareto front*.

The DE population at generation G contains NP D-dimensional parameter vector $\mathbf{P}_{\mathbf{x,G}} = (\mathbf{x}_{\mathbf{i,G}})$, $i = 1,..., NP$ and $\mathbf{x}_{\mathbf{i,G}} = [x_{1,i,G}, x_{2,i,G} ..., x_{D,i,G}]$ generated by

$$x_{j,i} = rand(0,1]\times\left(b_{j,U} - b_{j,L}\right) + b_{j,L} \qquad (2)$$

where $b_{j,L}$ and $b_{j,U}$ are the lower and upper bounds of the vectors $\mathbf{x}$ in dimension j, and $rand$[0,1) is the uniform random number and $j = 1,2...,D$. After population initialization, DE uses the mutation to generate a mutant vector $\mathbf{v}_{i,G}$ with respect to the target vector $\mathbf{x}_{i,G}$ by adding the base vector to weighted difference vectors:

$$\text{DE/rand/1: } \mathbf{v}_{i,G} = \mathbf{x}_{r1,G} + F(\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}). \qquad (3)$$

The index *i, r1, r2, and r3* are randomly generated and mutually exclusive within the range [1, *NP*]. F is a scaling factor. The original mutation operator is called DE/rand/1. There are many mutation strategy proposed by many researchers recently [9]. After we obtain the mutant vector we perform the crossover operation in order to increase the potential diversity of the population. Crossover is applied to each pair of the target vector $\mathbf{x}_{i,G}$ and it's mutant vector $\mathbf{v}_{i,G}$ then we obtain a trial vector $\mathbf{u}_{i,G} = [u_{1,i,G}, u_{2,i,G} ..., u_{D,i,G}]$. The DE family algorithms can use two types of crossover schemes: the binomial (uniform) crossover and the exponential crossover. The binomial crossover is the discrete recombination operator by

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if } rand_j(0,1) \le CR \text{ or } j = j_{rand} \\ x_{j,i,G} & \text{otherwise} \end{cases}. \qquad (4)$$

The crossover rate or so called crossover probability $CR \in [0,1]$ is a user-defined constant that controls the fraction of parameter values copied from the mutant vector.

The exponential crossover is similar to the two-point crossover used in genetic algorithms. Thus the trial vector $\mathbf{u}_{i,G}$ inherits components from its corresponding mutant vector $\mathbf{v}_{i,G}$ by randomly chosen starting point until the first time that $rand(0,1)<CR$. The remaining parameter inherits components from its corresponding mutant vector. The classical DE utilized the binomial crossover, and most of DE variants nowadays employ binomial crossover.

In order to select the next generation population $\mathbf{x}_{i,G+1}$, the target vector $\mathbf{x}_{i,G}$ competes with the corresponding trial vector $\mathbf{u}_{i,G}$ by objective value. If $\mathbf{u}_{i,G} \prec \mathbf{x}_{i,G}$, then $\mathbf{u}_{i,G}$ replaces $\mathbf{x}_{i,G}$ otherwise, $\mathbf{x}_{i,G}$ survive to the next generation.

B. Previuos Works in Adaptive Multiobjective DE

The performance of multiobjective DE is affected by balance of the exploration and exploitation during the evolution process. Balancing these two abilities can be based on choosing control parameters such as the scaling factor F, the crossover rate CR, and the population size NP, and the mutation strategy. There are some researchers that introduced the methods to adaptively adjust the control parameters and choosing the mutation strategy. For example, Huang *et al.* [3] extended the SaDE [10] to solve MOPs. They named the algorithm as the multi-objective SaDE algorithm (MOSaDE). The algorithm automatically adapts the trial vector generation strategies and their associated parameters according to their previous experience of generating promising solutions as same as SaDE. However, MOSaDE uses non-domination sorting and crowding in evaluation process. Later, Huang *et al.* [6] modified MOSaDE in order to learn the suitable crossover rate and mutation strategies for each objective separately in MOPs. Zamuda *et al.* [4] proposed differential evolution for multiobjective optimization with self-adaptation (DEMOSA). They extended the DEMO by incorporating the self-adaptive control parameters F and CR. F and CR will be encoded to the decision variables and simultaneously evolved with the population. Zhang and Sanderson [5] proposed the self-adaptive multiobjective DE with direction information provided by archived inferior solutions (JADE2) which is extended from JADE. JADE2 incorporated the self-adaption of F and CR and selection scheme based on Pareto dominance and crowding density. Adaptation of F and CR is based on the principle that the better values of control parameters tend to generate individuals that are more likely to survive and should propagate to the next generation.

First fuzzy adaptive parameters for DE proposed by Liu and Lampinen [11] for single objective optimization. They use fuzzy logic controllers to adapt the control parameters F and CR. The proposed algorithm is called a fuzzy adaptive DE (FADE). The inputs of the fuzzy controllers incorporate the relative objective function values and individuals of the successive generations. Xue *et al.* [12] introduced the fuzzy logic controlled multiobjective differential evolution (FLC-MODE). They use fuzzy logic controller to dynamically adapt the parameters of their previous multiobjective differential evolution version [13]. Population diversity and the percentage of generation are used as inputs for the fuzzy logic controller that dynamically controls the greediness and perturbation factor associated with the reproduction operator.

C. State-of-The-Art Multiobjective Optimization Algorithms

The nondominated sorting genetic algorithm-II (NSGA-II) [14] was improved from its first version. A fast nondominated sorting method is employed to Pareto rank individuals and a crowding distance measurement is the density estimation for each individual. In fitness assignment, NSGA-II prefers the one with the lower rank, or the one that located in a less crowded region if both points are in the same front. The crowding comparison method preserve the diversity of the population and no sharing parameter is required. The elitism mechanism does not allow an already found nondominated solution be deleted. Therefore, NSGA-II combines a fast nondominated sorting approach, a parameterless sharing method, and an elitism scheme in order to produce a better spread of solutions in some test functions. However, the nondominated sorting needs to be performed on a population size of 2NP.

A multiobjective evolutionary algorithm based on decomposition (MOEA/D) [15] decomposes a multiobjective optimization problem into a number of scalar single optimization subproblems and optimizes them simultaneously. Later, the new version of MOEA/D so called MOEA/D-DE [16] was introduced. MOEA/D-DE employs a DE operator and polynomial mutation. The simulation study of MOEA/D-DE shows that it is less sensitive to F and CR setting.

III. Fuzzy Multiobjective Differential Evolution Using Performance Metrics Feedback

The quality of the multiobjective optimization algorithms consists of three design goals [7]. First, the distance of the resulting nondominated set to the true Pareto-optimal front should be minimized. Second, a good (in most cases uniform) distribution of the solutions found is desirable. Last, the extent of the obtained nondominated front should be maximized, i.e., for each objective, a wide range of values should be covered by the nondominated solutions. This understanding motivated the idea that we used the performance metrics, specifically hypervolume, spacing, and maximum spread which match the three optimization goals as inputs to fuzzy rules to adapt the control parameters of the proposed FMDE algorithm in order to smoothly adapt to the emphasis on the convergence or the diversity and exploration and exploitation during the evolution process. The flowchart of the proposed FMDE is shown in Fig. 1.

The FMDE begins with randomly generated population and associated control parameters. The population will undergo the mutation and crossover processes. Afterward we combine the offspring and parent population together and identify the nondominated solutions of the combined population. The obtained nondominated front will be measured by all three performance metrics. These three values are inputs to the fuzzy rule based system. Outputs of the fuzzy rule based system are control parameters of DE, namely, scaling factor F and greedy factor γ for the mutation strategy that is used in FMDE. The fuzzy rules will be implemented every generation in order to adaptively adjust the parameter of mutation strategy for the next generation. The combined population size $2NP$ will be truncated to size NP. Then we update the archive by adding the nondominated solution found from the combined population. The truncation method used in maintaining external archive in FMDE follows the same approach from NSGA-II [14]. The new population undergoes the whole process until the stopping criterion is met. The stopping criterion is the preset maximum number of generation.

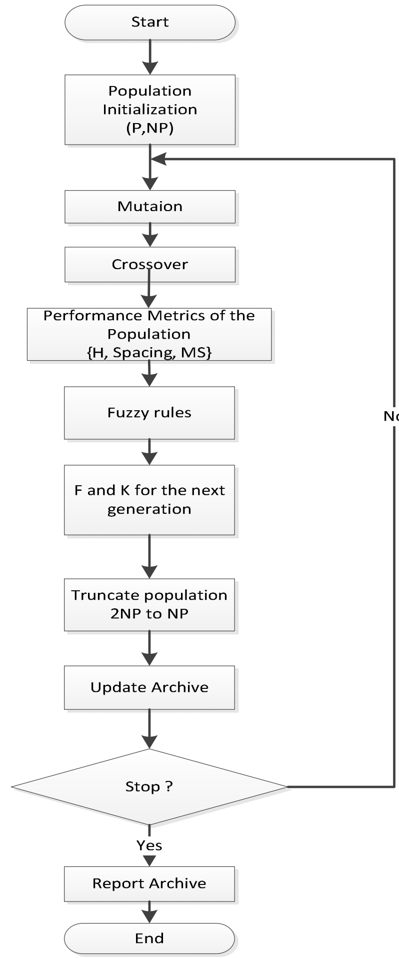

Fig. 1. Fuzzy Multiobjective Differential Evolution flowchart

A. Mutation Strategy

Joshi and Sanderson [17] proposed the mutation operator as

$$\mathbf{v}_{i,G} = \gamma \mathbf{x}_{best,G} + (1-\gamma)\mathbf{x}_{i,G} + F\sum_{k=1}^{K}(\mathbf{x}_{i_a^k} - \mathbf{x}_{i_b^k}), \qquad (5)$$

where $\gamma \in [0,1]$ is the greediness of the operator, $F \in [0,2]$ is the scaling factor, $\mathbf{x}_{best}$ is the best individual in the parent population, and K is the number of differentials used to generate the perturbation. The control parameter γ represents the degree of exploitation and greediness of the mutation operator. If γ value is large, the mutation strategy is greedier. Consequently, mutant vectors will be generated near the best vectors in the parent population and emphasize the exploitation ability of the algorithm. The scaling factor F controls the diversity and exploration ability of the mutation. If F value is larger, the degree of exploration is higher, and more diversity among the mutant vectors will be favored. Choosing the appropriate values for γ and F is often a trial-and-error, time-consuming, and problem-dependent task. Knowing the state of the current population through performance metrics, we can adjust the value of these parameters without the prior knowledge of the considered problem.

Even though Joshi and Sanderson suggested that F should be between 0 and 2, and it is the maximum scaling range for the DE research. According to a more recent study by Wang *et al.* [18], F should be set even tighter between 0.4 and 1. Given this consideration, our algorithm sets the range of F between 0.4 and 1.

B. Crossover Strategy

As stated in Section II, there are two types of crossover strategies employed in DE community: binomial and exponential crossovers. Zaharie [19] analyzed the influence of crossover type on the behavior of the DE algorithms, and found that the exponential crossover was more sensitive to the problem size than the binomial crossover does. Wang *et al.* [18] suggested that CR should be a low value near 0, or high value near 1. We therefore choose the binomial crossover and set $CR = 0.3$.

C. Performance Metrics

Since during the evolution process, we do not know the location of the true Pareto front. We choose the performance metrics that can measure the three design goals of optimization without the knowledge of the true Pareto front. Although there are proposed running performance metrics by Deb and Jain [8], there are very few choices to allow us to measure the convergence, uniform distribution, and extensiveness of the population. Hence, we choose three performance metrics in order to measure the three properties of the obtained nondominated solutions as the following.

In order to measure convergence, the metric we choose is hyperarea Ratio (hypervolume indicator) [20]. It calculates the size of the hypervolume enclosed by the obtained nondominated front PF_{known} and a reference point. For instance, an individual x_i in PF_{known} for a two-dimensional MOP defines a rectangle area, $a(x_i)$, bounded by an origin and $f(x_i)$. The union of such rectangle areas is referred to as hyperarea of PF_{known},

$$H(PF_{known}) = \left\{ \bigcup_i a(x_i) \middle| \forall x_i \in PF_{known} \right\}. \quad (6)$$

It measures both convergence and distribution of a nondominated set, and reference points are set as discussed in [21]. If hypervolume value is larger, we can interpret the status of the population as is converging and/or with good distribution. However, it is not clear that the increased value is due to converging, or better distribution, or both. Therefore, we need another metric that can measure the degree of uniform distribution, i.e., spacing.

Spacing (S) [22] is a metric measuring how the obtained nondominated solutions are evenly distributed:

$$S = \sqrt{\frac{1}{\bar{n}} \sum_{i=1}^{\bar{n}} (d_i - \bar{d})^2}, \bar{n} \to |PF_{known}| \quad (7)$$

where d_i is the Euclidean distance in the objective space between individual x_i and the nearest solution of the true Pareto front, and $\bar{n}$ is the number of solutions in the obtained nondominated front. If S is zero, it indicates that all solutions of the nondominated front are equally spaced.

Maximum spread (MS) [7] measures the length of diagonal hyperbox formed by the extreme solutions observed in the nondominated sets. But it does not reveal the distribution of solutions. A normalized version of MS [23] is

$$MS = \sqrt{\frac{1}{M} \sum_{m=1}^{M} \left(\frac{\max_{i=1}^{Q} f_m^i - \min_{i=1}^{Q} f_m^i}{F_m^{\max} - F_m^{\min}} \right)^2} \quad (8)$$

where $F_m^{\max}$ and $F_m^{\min}$ are the maximum and minimum values of the m-th objective in the chosen set of Pareto optimal solutions. Q is a set of the obtained nondominated solutions. f_m^i is the value of the m-th objective function of the i-th member of Q and M is the dimension of the objective function.

D. Fuzzy Membership Functions and Fuzzy Rules

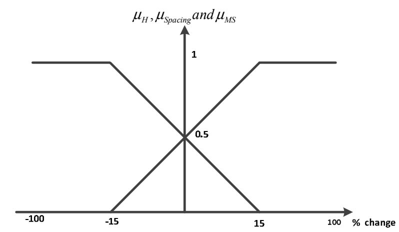

Fig. 2. Input membership functions

The membership functions of hypervolume, spacing, and maximum spread are shown in Fig. 2. All three performance metrics use the same shape of membership functions for simplicity. The input is the percent change of performance metrics calculated every two successive generation and fuzzified to the "decreasing" and "increasing" membership values. The fuzzification method is "and" method. The output membership function for γ and F are the same shape. There are three status, namely, "decrease," "no change," and "increase" for γ and F value. The "centroid" defuzzification is used, then we get the percent change of γ and F value. The output membership function is shown in Fig. 3.

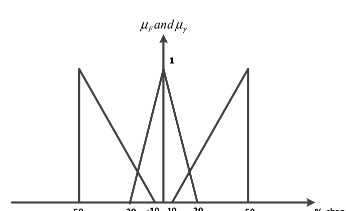

Fig. 3. Output membership functions

The fuzzy inference rules are shown in Table 1. These rules are used for adjusting the value of γ and F in order to emphasize the exploitation (greedy) or exploration (diversity) of the mutation strategy for the next generation.

Once we receive the quality feedback of the nondominated front through all performance metrics, we can design how we emphasize the exploitation or exploration abilities of the proposed algorithm. If we need to put strong emphasis on exploitation, we increase γ and decrease F. If we need to place strong emphasis on exploration, we decrease γ and increase F. However, if we need to place a mild emphasis on the exploitation, we can do it by two methods: increase γ and keep F unchanged, or keep γ unchanged and decrease F. To place the mild emphasis on the exploration, we can have two options as the same manner as exploitation: keep γ unchanged and increase F or decrease γ and keep F unchanged.

TABLE I. FUZZY RULES

Rules	Inputs			Outputs	
	Hyper Volume	***Spacing***	***MS***	**γ**	***F***
1	Increase	Increase	Increase	Increase	No Change
2	Increase	Increase	Decrease	Decrease	No Change
3	Increase	Decrease	Increase	No Change	No Change
4	Increase	Decrease	Decrease	No Change	Increase
5	Decrease	Increase	Increase	Decrease	No Change
6	Decrease	Increase	Decrease	Decrease	Increase
7	Decrease	Decrease	Increase	No Change	Increase
8	Decrease	Decrease	Decrease	No Change	Increase

Rule number 3 is the best status of the population because hypervolume is increasing and spacing is decreasing, that implies the population is converging and more uniformly distributed. The extensiveness of the obtained front is larger by increasing MS value. Thus, we will not change γ and F because they should remain the suitable values for the current status. Compared to Rule 3, in Rule number 4 we need mild exploration in order to increase the extensiveness of the population. So, we do nothing with γ, but increase F slightly.

Rule number 6 is the worst case scenario, because all three metrics states that the obtained front is diverging and losing diversity and extensiveness in the obtained nondominated front. We need strong exploration so to decrease γ and increase F.

In the case of increasing hypervolume, it means the population is converging but we do not know whether the nondominated solutions are uniformly distributed or not. Thus, we consider spacing metric. If it is increasing, so the solutions are not well distributed and we need a mild exploitation. Rules 1 and 2 are under this case, but the maximum spread for Rule 1 is increasing then F is kept unchanged but decreased the γ value. Maximum spread for rule number 2 is decreasing, the stage of population is converging but not well distributed and the searched area is shrinking. Thus we need to increase the mild degree of exploration. Since the spacing is worsen, then we decrease γ, and keep F unchanged.

If the hypervolume is decreasing, it implies that the search direction of the population is incorrect; we will increase the exploration ability. Rules 5 to 8 are under this case. Rule 5 and 7 need to increase the mild degree of exploration. Spacing of rule 5 is decreasing, it implies that the solutions are crowded then we decrease γ, while keep γ unchanged for rule 7.

Rule 8 states that the population is not converging and the search area of the population is shrinking even though the distribution is good. We increase the mild degree of exploration by keep γ unchanged and increase F.

IV. EXPERIMENTAL RESULTS

The proposed MODE, FMDE, is tested on the ZDT test suit [10], i.e., ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6. All test functions are bi-objective optimization problems. DTLZ1 to DTLZ7 [28] in three objectives are tested as well. Also WFG1 and WFG2 [29] in two objectives are tested.

ZDT1 is a convex Pareto-optimal front, while ZDT2 is having the nonconvex counterpart to ZDT1. ZDT3 represents the discontinuous Pareto-optimal front. It consists of several discontinuous convex parts. ZDT4 contains many local Pareto-optimal fronts. The search space of ZDT6 is non-uniformity. Thus, it causes two difficulties: first, the Pareto-optimal front is non-uniformly distributed, and the density of the solutions is lowest near the Pareto front and highest away from the front. ZDT5 is not included in the experiment because the decision variable is a binary string.

DTLZ1 is a linear hyper-plane and have many local Pareto fronts where an MOEA can be attracted to them before reaching the global Pareto front. DTLZ2 is the spherical Pareto optimal front. DTLZ3 is the concave and multimodality. The local optimal fronts are parallel to the global Pareto optimal front. DTLZ4 is the modification of DTLZ2 to allow the solutions to be crowded near the $f_M - f_1$ plane. DTLZ5 is to test the ability of an MOEA to converge to a curve. DTLZ6 is modified from DTLZ5 to make the problem harder. DTLZ7 has four disconnected Pareto optimal regions in the search space. WFG1 is a convex, mixed shape Pareto front where an MOEA should have the ability to coping with bias. WFG2 is a convex disconnected front.

A. Experimental Setup

The proposed algorithm is compared with state-of-the-art MOEA/D-DE [16], NSGA-II [14], and SPEA2 [24]. We compare FMDE with JADE2 [5] which is the self-adaptive MODE as well. Each algorithm is tested on the two-objective, ZDT test functions, three-objective DTLZ test functions and bi-objective WFG1 and WFG2 with 30 independent runs.

For each trial, an algorithm will stop if it reaches 250 generations for bi-objective and three objective problems. The population size is 100 for bi-objective and 300 for three objective problems.

The FMDE use the external archive and is updated as in [14]. γ and F are 0.5 for the first generation. K is 1 which implies we use only one difference vector. Even though CR is the diversity control mechanism for DE as well, however if we adaptively adjust γ, F, and CR, it can be overly used for the diversity effect, we set CR = 0.3 and F should be limit in [0.4, 1] for the whole experiments [18].

The other parameter settings for MOEA/D-DE, NSGA-II, SPEA2 and JADE2 are the same as what suggested in the original papers. The reference points we set for calculating hypervolume indicator for all test function are the same as [21]. We choose the reference point (3,100) for ZDT1 and the other points for ZDT2, ZDT3, ZDT4, ZDT6 are (3/2, 4/3), (100, 5.446), (1.497, 4/3). DTLZ1 is (1, 1, 1). The reference point (1.180, 1.180, 1.180) for DTLZ2, DTLZ3 and DTLZ4 is the same. DTLZ7 is (13.3725, 5.3054, 5.3054). Since [21] does not state the reference points for DTLZ5 and DTLZ6, we choose the reference point as (5, 5, 5) for both functions and the reference point for both WFG1 and WFG2 are (7,9) .

B. Performance Metrics

In comparing the performance between three algorithms, the performance metric used in this experiment is the inverted generational distance (*IGD*) [25]. Let PF_{true} be the uniformly distributed true Pareto front. Let P_A be the obtained approximated one. *IGD* is defined as

$$IGD(P_A, PF_{true}) = \frac{\sum_{v \in PF_{true}} d(v, P_A)}{|PF_{true}|}, \quad (9)$$

where $d(v, P_A)$ is the minimal Euclidean distance between every $v \in PF_{true}$ and set P_A. *IGD* measures both the convergence and diversity of the obtained approximation front. If $IGD = 0$, it means that all the approximation solutions are in the true Pareto solutions and they cover all the extension of the true Pareto front.

The uniformly distributed true Pareto fronts for calculating *IGD* for all test problems are taken from [26]. $|PF_{true}|$ for ZDT1, ZDT2, ZDT3 and ZDT6 is 1001 whereas ZDT4 is 269. Also $|PF_{true}|$ for DTLZ1 and DTLZ2 is 10,000, DTLZ3 and DTLZ4 is 4,000, DTLZ5 is 166,500, DTLZ6 is 28,000 and DTLZ7 is 676. $|PF_{true}|$ for WFG1 is 1,113 and WFG2 is 119.

C. Experimental Results

An example of the obtained nondominated front and the associated performance metrics, and control parameters are shown in Fig. 4 for ZDT1. At the beginning of the evolutionary process, γ and F starts at 0.5. At 20 generation, the spacing is the highest while MS is the lowest, this means the obtained front is crowed and the distance between extreme solutions are shorter, even though the hypervolume is increasing. This shows that the population can find more nondominated solutions but they are crowded together in some part of the approximation front. Hypervolume is continuously improved and go to steady around 100 generation. After that the hypervolume has very small fluctuation which means FMDE continuously improves its convergence and distribution. When the algorithm converges, we can see that the spacing went to near zero which is the ideal value for evenly distribution. The maximum spread of the algorithm is glowing to approximately one which stated that the algorithm reach its maximum extent of the extreme solutions. Later γ is decreasing while F is increasing according to the search process detect the promising region and then fast converge toward that direction which make γ value high in order to use the exploitation ability. When the population converges, the exploration becomes prominent because every individual will be near or at the true Pareto front and we need to do the local search. As can be seen, the FMDE is continuously improves its performance: hypervolume is increasing which demonstrate that the algorithm is converging but we observe that the distribution is not good due to spacing values are fluctuating. Meanwhile the fluctuation of the maximum spread indicates that our extreme nondominated solutions occupy smaller space. After 120 generations, γ decreased to the lowest value near zero, but F is closed to 1, means that the degree of exploration is higher that exploitation. What it means is that the algorithm converged, the number of the nondominated solutions found are high, the algorithm tried to do the local search to be evenly distributed.

The mean value and standard deviation of IGD for FMDE, MOEA/D-DE, NSGA-II, SPEA2 and JADE2 are shown in Table 2. We compare the performance between any two algorithms in terms of statistics by utilizing the *t-test* on IGD with 95% of confidence level.

Table II shows that the proposed FMDE outperforms MOEA/D-DE 10 out of 14 functions. In case of bi-objective, FMDE is competitive with MOEA/D-DE. The performance of FMDE is statistically better than MOEA/D-DE in ZDT1, ZDT2, ZDT3, WFG1 and WFG2 but worse for ZDT4 and ZDT6. However, FMDE outperforms MOEA/D-DE 5 out of 7 three objective functions. FMDE outperforms NSGA-II 12 out of 14 functions. It outperform NSGA-II on all ZDT and WFG test functions, but underperforms on DTLZ2. Whereas FMDE outperforms SPEA2 on ZDT1 and ZDT4 but the performance on ZDT6 is not statistically difference. In comparison with SPEA2, FMDE is competitive with SPEA2 for both bi-objective and three objective problems. FMDE outperform JADE2 for all test problems.

Overall, FMDE is competitive with the other algorithms for bi-objective benchmark functions, and outperforms all the others on ZDT and WFG. It is superior for the three objective benchmark functions especially DTLZ2, DTLZ4 and DTLZ5. The most difficult problem for FMDE is DTLZ1.

FMDE perform good on the convex, nonconvex, and discontinuous problems. But it faces difficulties on the multimodality problem such as ZDT4 and DTLZ1. The preservation of diversity in FMDE is not enough for solving the multimodality. It should be improved by adaptively adjust *CR*.

V. Conclusion

This paper presents a MODE which utilizes hypervolume, spacing, and maximum spread to indicate the stage of evolution in order to dynamically adapt the greedy and distribution parameters of a mutation strategy used in DE. The direction of change for each parameter is determined by fuzzy rules. The effect of dynamically adjust these parameters is that we can emphasize the exploitation or exploration ability due to the status of the search process.

The experimental results show that FMDE is better than two chosen state-of-the-art and one of the adaptive MODEs. This research demonstrates that we can combine performance metrics and human knowledge of optimization process together by fuzzy rules. Therefore, it is one possible method to automatically adjust the control parameter values without a prior knowledge on the problem. Our future research is to adaptively adjust *CR* in concurrent with *F*, and use the performance metrics as the maintaining mechanism for the external archive, and stopping criteria.

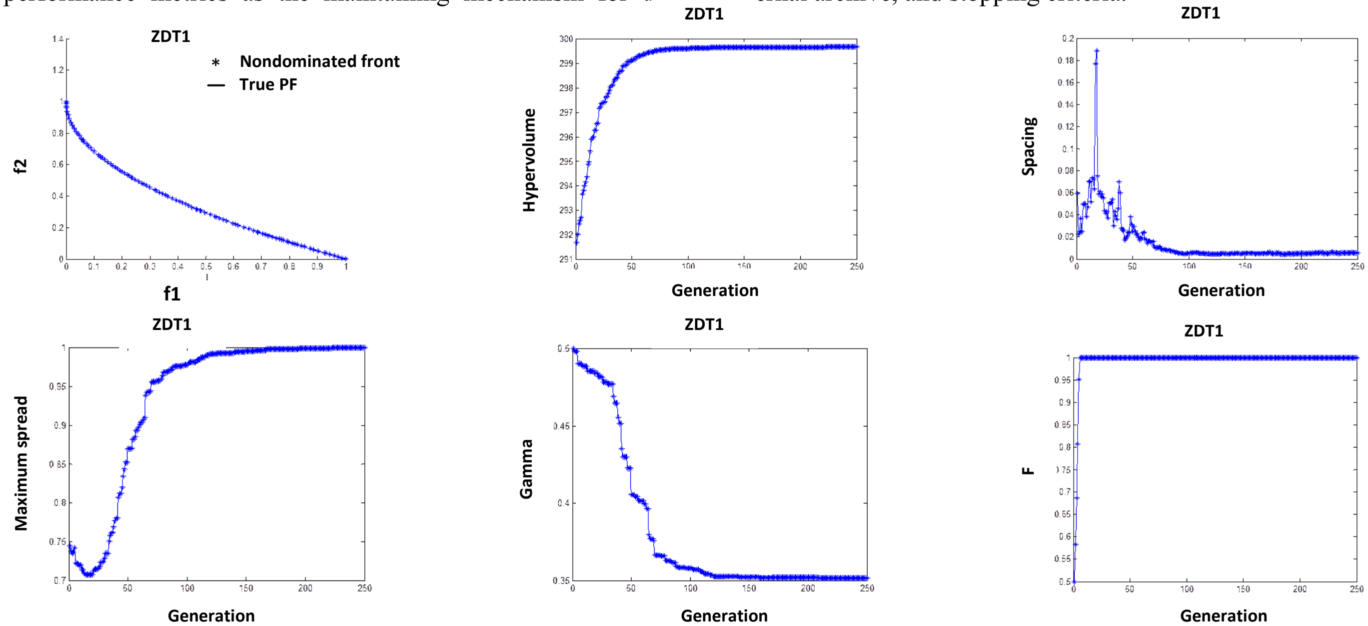

Fig. 4. An example result of ZDT1. The approximated front, true Pareto front, and performance metrics, and control parameters.

TABLE II. COMPARISON OF IGD FOR FMDE, OTHER MODEs AND MOEAs ON T-TESTS

Functions		FMDE	MOEA/D-DE	NSGA-II	SPEA2	JADE2
ZDT1	Mean	**0.0048**	0.1460	0.6174	3.14684e+1	0.2385
	Std.	**2.5595e-4**	5.2931e-18	0.2605	1.084e-14	0.0328
	t-test		-3.0216e+3	-1.28804e+1	-6.7331e+5	-3.9024e+1
			+	+	+	+
ZDT2	Mean	0.0047	0.0064	0.9636	**0.0042**	0.1591
	Std.	1.9826e-4	8.8219e-19	0.7273	**0.0000**	0.0281
	t-test		-4.6956e+1	-7.2214	1.38132e+1	-3.0095e+1
			+	+	-	+
ZDT3	Mean	0.0035	0.0123	0.3730	**0.0033**	0.2269
	Std.	2.3176e-4	7.0575e-18	0.1805	**1.3233e-18**	0.0297
	t-test		-2.07972e+2	-1.12124e+1	4.7266	-4.1198e+1
			+	+	-	+
ZDT4	Mean	0.6949	**0.0070**	1.6299	30.2537	3.1201e+1
	Std.	0.6792	**0.0025**	0.8700	1.4454e-14	7.3325
	t-test		5.5473	-4.6399	-2.3837e+2	-2.2690e+1
			-	+	+	+
ZDT6	Mean	0.0037	**0.0021**	5.1635	0.0037	0.0799
	Std.	6.545e-4	**0.0000**	1.4989	8.8219e-19	0.0093
	t-test		1.33897e+1	-1.88548e+1	0.0000	-4.4767e+1
			-	+	=	+
DTLZ1	Mean	2.3507e+2	**0.5108**	8.7177	32.4673	2.4554e+2
	Std.	0.7330	**6.0532e-4**	3.4200	1.4454e-14	4.5334
	t-test		1.7527e+3	3.5446e+2	1.5139e+3	-1.2485e+1
			-	-	-	+
DTLZ2	Mean	**0.0398**	0.6241	0.0714	0.0609	0.524
	Std.	**0.0010**	1.1292e-16	0.0034	3.5288e-17	0.0710
	t-test		-3.2003e+3	-4.8837e+1	-1.1557e+2	-3.7357e+1
			+	+	+	+
DTLZ3	Mean	0.9965	0.5569	**0.3509**	9.2031	1.8621e+2
	Std.	1.5138	3.3876e-16	**0.2640**	3.6134e-15	1.4556e+1
	t-test		1.5906	2.3012	-2.9693e+1	-6.9320e+1
			-	-	+	+
DTLZ4	Mean	**0.0442**	0.2940	0.0532	0.0490	0.5241
	Std.	**0.0033**	1.6938e-16	0.0070	0.0000	0.0710
	t-test		-4.14609e+2	-6.3698	-7.9669	-3.6982e+1
			+	+	+	+
DTLZ5	Mean	0.6066	0.9385	0.6270	**0.6061**	0.6387
	Std.	0.0086	2.2584e-16	0.0057	**4.5168e-16**	0.0310
	t-test		-2.1138e+2	-1.08297e+1	0.3184	-5.4652

Functions		FMDE	MOEA/D-DE	NSGA-II	SPEA2	JADE2
			+	+	-	+
DTLZ6	Mean Std. t-test	**0.6015** **0.0105**	3.4282 2.2584e-15 -1.4745e+3 +	4.1792 1.1628 -1.68516e+1 +	1.9353 6.7752e-16 -6.9576e+2 +	1.3176 0.6640 -5.9063 +
DTLZ7	Mean Std. t-test	0.0285 8.0856e-4	0.6550 0.0000 -4.2439e+3 +	0.1351 0.0740 -7.8897 +	**0.0276** **1.0586e-17** 6.0966 -	0.2834 0.0394 -3.5428e+1 +
WFG1	Mean Std. t-test	**0.5243** **5.2681e-5**	0.5355 0.0000 -1.1645e+3 +	0.5346 1.3103e-4 -3.9945e+3 +	0.5774 1.1292e-16 -5.5208e+3 +	0.5341 0.0041 -1.3091e+1 +
WFG2	Mean Std. t-test	**0.4594** **2.5406e-5**	0.5584 1.1292e-16 -2.1343e+4 +	0.5597 0.0014 -3.9234e+2 +	0.5584 1.1292e-16 -2.1343e+4 +	0.4617 3.9464e-4 -3.1856e+1 +
	Better (+) Same (=) Worse (-) Score		10 0 (4) 6	12 0 (2) 10	8 1 (5) 3	14 0 (0) 14

REFERENCES

[1] R. Storn and K. Price, "Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces," *Technical Report TR-95-012*, March, 1995 [online]. Available: http://icsi.berkeley.edu/~litera.html.

[2] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997.

[3] V. L. Huang, A. K. Qin, P. N. Suganthan, and M. F.Tasgetiren, "Multi-objective optimization based on self-adaptive differential evolution algorithm," in *Proc. of Congress on Evolutionary Computation*, 2007, pp. 3601-3608.

[4] A. Zamuda, J. Brest, B. Boskovic, and V. Zumer, "Differential evolution for multiobjective optimization with self adaptation," in *Proc. of IEEE Congress on Evolutionary Computation*, 2007, pp. 3617-3624.

[5] J. Zhang and A. C. Sanderson, "Self-Adaptive multi-objective differential evolution with direction information provided by archived inferior solutions," in *Proc. of IEEE Congress on Evolutionary Computation*, 2008, pp. 2801-2810.

[6] V. L. Huang, S. Z. Zhao, R. Mallipeddi, and P. N. Suganthan, "Multi-objective optimization using self-adaptive differential evolution algorithm," in *Proc. of IEEE Congress on Evolutionary Computation*, 2009, pp. 190-194.

[7] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary Computation,* vol. 8, pp. 173-195, 2000.

[8] K. Deb and S. Jain, "Running performance metrics for evolutionary multi-objective optimization," *KanGal Technical Report No. 2002004*, 2002.

[9] R. Mallipeddi and P. N. Suganthan, "Differential evolution algorithm with ensemble of populations for global numerical optimization," *OPSEARCH,* vol. 46, pp. 184-213, 2009

[10] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. on Evolutionary Computation,* vol. 13, pp. 398-417, 2009.

[11] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Computing - A Fusion of Foundations, Methodologies and Applications,* vol. 9, pp. 448-462, 2005.

[12] F. Xue, A. C. Sanderson, P. P. Bonissone, and R. J. Graves, "Fuzzy Logic Controlled Multi-Objective Differential Evolution," in *Fuzzy Systems, 2005. FUZZ '05. The 14th IEEE International Conference on*, 2005, pp. 720-725.

[13] F. Xue, A. C. Sanderson, and R. J. Graves, "Pareto-based multi-objective differential evolution," in *The 2003 Congress on Evolutionary Computation, CEC '03.*, 2003, Vol.2, pp. 862-869.

[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation, ,* vol. 6, pp. 182-197, 2002.

[15] Q. Zhang, and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Trans. on Evolutionary Computation,* vol. 11, pp. 712-731, 2007.

[16] H. Li and Q. Zhang, "Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II," *IEEE Trans. on Evolutionary Computation*, , vol. 11, pp. 712-731, 2007.

[17] R. Joshi and A. C. Sanderson, "Minimal Representation Multisensor Fusion Using Differ ential Evolution," *IEEE trans. on Systems, Man, and Cybrnatics---Part A*, No. 29, Vol. 1, pp. 63-76, 1999.

[18] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. on Evolutionary Computation*, vol. 15, pp. 55-66, 2011.

[19] D. Zaharie, "Influence of crossover on the behavior of differential evolution algorithms," *Appl. Soft Comput.,* vol. 9, pp. 1126-1138, 2009.

[20] D. V. Veldhuizen, "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations," PhD Thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering., Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1999

[21] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point," in *Proceedings of ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, Orlando, FL, 2009, pp. 87-102.

[22] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," MS Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge Massachusetts, 1995.

[23] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*: Wiley, 2009.

[24] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength pareto evolutionary algorithm," Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, Technical report TIK-Report 103, May 2011.

[25] C. A. C. Coello and N. C. Cortes, "Solving multiobjective optimization problems using an artificial immune system," *Genetic Programming and Evolvable Machines,* vol. 6, pp. 163-190, 2005.

[26] J. J. Durillo and A. J. Nebro, "jMetal: A Java framework for multi-objective optimization," *Advances in Engineering Software,* vol. 42, pp. 760-771, 2011.

[27] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. on Evolutionary Computation*, vol. 10, pp. 477-506, 2006.