A New Framework for Self-adapting Control Parameters in Multi-objective Optimization

Xin Qiu
NUS Graduate School for Integrative Sciences and Engineering
National University of Singapore, Singapore
qiuxin@u.nus.edu

Weinan Xu
Department of Electrical and Computer Engineering
National University of Singapore, Singapore
pumpkinxu@163.com

Jian-Xin Xu
Department of Electrical and Computer Engineering
National University of Singapore, Singapore
elexujx@nus.edu.sg

Kay Chen Tan
Department of Electrical and Computer Engineering
National University of Singapore, Singapore
eletankc@nus.edu.sg

ABSTRACT

Proper tuning of control parameters is critical to the performance of a multi-objective evolutionary algorithm (MOEA). However, the developments of tuning methods for multi-objective optimization are insufficient compared to single-objective optimization. To circumvent this issue, this paper proposes a novel framework that can self-adapt the parameter values from an objective-based perspective. Optimal parametric setups for each objective will be efficiently estimated by combining single-objective tuning methods with a grouping mechanism. Subsequently, the position information of individuals in objective space is utilized to achieve a more efficient adaptation among multiple objectives. The new framework is implemented into two classical Differential-Evolution-based MOEAs to help to adapt the scaling factor F in an objective-wise manner. Three state-of-the-art single-objective tuning methods are applied respectively to validate the robustness of the proposed mechanisms. Experimental results demonstrate that the new framework is effective and robust in solving multi-objective optimization problems.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; G.1.6 [**Numerical Analysis**]: Optimization

GECCO '15, July 11 - 15, 2015, Madrid, Spain
© 2015 ACM. ISBN 978-1-4503-3472-3/15/07…$15.00
DOI: http://dx.doi.org/10.1145/2739480.2754714

Keywords

Multi-objective optimization, Parameter tuning, Differential Evolution

1. INTRODUCTION

Multi-objective Evolutionary Algorithms (MOEAs) are favored approaches for solving optimization problems with multiple conflicting objectives. A good volume of work has been undertaken to extend powerful Single-objective evolutionary algorithms, including Differential Evolution (DE) [21] and Particle Swarm Optimization (PSO) [12], into Multi-objective Optimization (MO) [16, 5]. According to previous empirical studies [6, 3, 20, 17], the success of DE and PSO in solving a particular problem is very closely related to the fine-tuning of their control parameters. To enhance the optimization performance, many methods were proposed to help adapt these parameters intelligently [3, 19, 27, 10, 11, 25, 23]. However, most of the efficient tuning mechanisms are restricted to Single-objective Optimization (SO), and little attention has been devoted to the adaptation of control parameters in MO. Actually, the parametric sensitivity issue may become more challenging since multiple objectives may have disparate requirements for the parametric setup. Self-adaptation of control parameters from the perspective of objective space is still a research gap. To fill the above niche, a new parameter adaptation framework for multi-objective optimization is developed in this paper. The proposed framework is able to make use of existing single-objective tuning methods during multi-objective optimization. The optimal parametric setups will be estimated in an objective-wise manner, and the positions of individuals in objective space will be considered in order to achieve a more efficient adaptation. Implementations with three state-of-the-art SO adaptation methods and two classical MOEAs demonstrate that the proposed framework is effective and robust in solving MO problems.

The rest of this paper is organized as follows. Section 2 reviews the previous work in parameter adaptation for DE briefly and introduces the basic ideas of three adaptation methods involved in later experiments. Section 3 describes the technical details of the proposed framework and explains the underlying rationales. Section 4 presents the experimental results for evaluation of the new mechanisms. Concluding remarks and future work are summarized in Section 5.

2. BACKGROUND

2.1 Parameter adaptation for DE

Typically, the self-adaptation methods for control parameters in DE can be classified into three main categories:

1) Randomization Strategy

A randomization model is pre-defined, and new parameters will be generated based on the probability in which this randomization model is distributed. This kind of strategies do not take any feedback from evolutionary search into account [27]. Two successful examples are jDE [3], and SaNSDE [24].

2) Statistical Strategy

These approaches will update parameter value as a statistical model such as normal distribution or Cauchy distribution. With the information acquired from evolutionary search, the mean value of the distributions will be updated so that the control parameters will be real-time tuned. JADE [27], MDE_pBX [10], and SaDE [19] are representative algorithms in this class.

3) Coevolution Strategy

Control parameters are combined with the individuals as the chromosome, and undergo mutation and crossover altogether with individuals. Since the process of evolution is to search the optimal solution, control parameters will also evolve to better values, and these values can be propagated to more offspring [27]. SPDE [1] and DESAP [22] belong to this category.

Among all the existing adaptive DE variants, the tuning methods applied in jDE [3], JADE [27], and MDE_pBX [10] present overall good performances [6]. In this paper, all of them will be implemented into the proposed framework independently. There are two control parameters in classical DE, namely, scaling factor F and crossover probability Cr. Compared to Cr, the overall performance of DE depends more heavily on the selection of F [26, 7], which has a big role in controlling the searching step of the optimization process. Therefore, the empirical study in this paper will focus on adapting the scaling factor F using the new framework. The following subsections introduce the adaptation mechanisms of F in jDE, JADE, MDE_pBX, respectively.

2.2 jDE

In jDE, all the control parameters are applied at the individual level. The new control parameters for each individual $X_{i,G}$, where i is the index of the solution in population and G is the number of current generation, are generated as

$$F_{i,G+1} = \begin{cases} F_l + rand_1 \times F_u & \text{rand}_2 < \tau_1 \\ F_{i,G} & \text{rand}_2 \geq \tau_1 \end{cases} \quad (1)$$

where $rand1$ and $rand2$ are uniform random values $\in [0, 1]$. F_l and F_u are the lower bound and upper bound for the new generated parameter values. τ_1 represent the probability to adjust scaling factor F.

The contribution of parameter tuning in jDE is that the self-tuning formula is quite simple and easy to manipulate. It outperforms traditional DE and some other self-tuning methods like FADE [15] in handling single-objective problems [3].

2.3 JADE

Similar to jDE, the principle for adaptation mechanism in JADE is that better control parameters tend to generate individuals which are more likely to survive, and better individuals help to propagate their associated paremeters. In JADE, F is encoded into each individual, and the associated F for each solution is updated as follows:

$$mean_L\left(S_F\right) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F} \quad (2)$$

$$\mu_F = (1 - c) \times \mu_F + c \times mean_L\left(S_F\right) \quad (3)$$

$$F_i = randc_i\left(\mu_F, 0.1\right) \quad (4)$$

where S_F is the set of successful F values in previous generation. More specifically, for any F_i, if the child solution $X_{i,G+1}$ replaces its parent solution $X_{i,G}$, this F_i will be stored in the set S_F. $randc_i\left(\mu_F, 0.1\right)$ refers to a Cauchy distribution generator with location parameter μ_F and scale parameter 0.1. The location parameter μ_F of the Cauchy distribution is initialized as 0.5 and then updated at the end of each generation via the above formula. c is a positive constant between 0 and 1.

Cauchy distribution is applied instead of normal distribution or uniform distribution because it is more helpful to diversify the F values in current population, thereby avoiding premature convergence which often occurs with highly concentrated parameter values. According to [27], the tuning method in JADE is very effective in solving single-objective problems.

2.4 MDE_pBX

The parameter adaptation schemes in MDE_pBX are inspired by those used in JADE algorithm [10], but the former ones have their own distinctive characteristics. At every generation, the scaling factor F_i associated with each individual is independently generated via the following process:

$$\omega_F = 0.8 + 0.2 \times rand(0, 1) \quad (5)$$

$$mean_{Pow}\left(F_{success}\right) = \sum_{x \in F_{success}} \left(x^n / \left|F_{success}\right|\right)^{1/n} \quad (6)$$

$$F_m = \omega_F \times F_m + (1 - \omega_F) \times mean_{Pow}\left(F_{success}\right) \quad (7)$$

$$F_i = Cauchy(F_m, 0.1) \quad (8)$$

where $F_{success}$ stores the successful F values in previous generation and is formed via the same way as S_F in JADE. $Cauchy(F_m, 0.1)$ is a random number sampled from a Cauchy distribution with location parameter F_m and scale parameter 0.1. $rand(0, 1)$ stands for a uniformly distributed random number in (0,1) and $meanPow$ stands for the power mean. ω_F is a weight factor varies randomly between 0.8 to 1. n is fixed as 1.5.

The main differences between adaptation mechanisms in MDE_pBX and JADE are the dynamic weight and power mean calculation. The experimental results in [10] show that the tuning approach of MDE_pBX performs better than those in jDE and JADE over a wide variety of single-objective problems.

3. PROPOSED FRAMEWORK

3.1 Overview of the framework

Based on our above review, many researchers have already investigated the adaptation of control parameters in SO and proposed numerous efficient schemes. In contrast, parameter adaptation in MO received less attention and the development of related methods is insufficient. When solving MO problems, more than one objective needs to be optimized simultaneously and each of them may have disparate parametric requirements. This intrinsic difference between SO and MO causes the difficulty in directly applying SO adaptation approaches into MO problems. To overcome this issue, we propose a new framework that can extend existing single-objective adaptation approaches into MO. The aim of the proposed framework is to effectively adapt the control parameters from an objective-based perspective.

During implementation of the new framework, optimal control parameters for each objective will be estimated via corresponding estimation groups (formed by selected individuals), where SO adaptation schemes are applied. Based on the estimation results, the parameter values for remaining individuals will be produced according to their current positions in objective space. Considering the different structures of MOEAs, two versions are designed for more efficient and convenient implementation of the proposed framework. One is the general version, which can be implemented in dominance-based MOEAs [8, 9, 13], indicator-based MOEAs [2] and other population-based MOEAs. Another is the decomposition-based version, which shows high efficiency when used with decomposition-based MOEAs [28, 14]. Detailed descriptions of the proposed framework will be provided in the next subsections.

3.2 General Version

3.2.1 Basic Structure

In the proposed framework, control parameters are applied at the individual level. Each solution has its own associated parameters. The introduced mechanisms will be performed after survival selection, which is always at the end of each generation. For a better interpretation, we divide the framework into three phases.

The first phase is grouping, during which particular individuals need to be selected to form the estimation groups. The target of estimation groups is to estimate the optimal parametric setups for each objective. Thus, those individuals which place obviously more weight on optimizing one of the objectives are preferred. In order to picking up the individuals with above-mentioned properties, a sorting procedure is performed among the population based on their fitness values on each separate objective. Since the different objectives are commonly conflicting to each other, higher ranking in one objective indicates higher weight on optimizing this objective. Based on our preliminary tests for group size, the individuals that rank among top 10% within one objective will be recorded as the estimation group for this objective.

The second phase is adaptation. In this phase, the parameter values of the individuals in estimation groups will be adapted using single-objective adaptation schemes. Depending on the procedures of the utilized schemes, additional modifications may be needed for a more efficient implementation. For instance, the adaption mechanisms in JADE and MDE_pBX involve the selections of successful F values, which are supposed to have different definitions in SO and MO. During implementations of the proposed framework, separate sets are created for each objective to store their corresponding successful F values. More specifically, if the offspring outperforms the parent in kth objective, the F value associated with the parent will be recorded as a successful F value for kth objective. After that, adaptation mechanisms are performed for the estimation groups of each objective using the corresponding successful parameter sets.

Algorithm 1 Procedure of General Version

Random Initialization
for $g = 1, 2, \cdots, G_{\max}$ **do**
  Reproduction operations (e.g., Crossover, Mutation)
  Update of successful F sets if needed
  Survival Selection
  for each objective k, $k = 1, 2, \cdots, M$ **do**
    Clear corresponding estimation group.
    Sorting the population w.r.t the fitness values in kth objective.
    for each individual i, $i = 1, 2, \cdots, N$ **do**
      The rank of individual $X_{i,g}$ is recorded as $r^k_{i,g}$.
      if $r^k_{i,g} \leq 10\% \cdot N$ **then**
        $X_{i,g}$ is added into the estimation group for kth objective.
      end if
    end for
  end for
  for each individual i, $i = 1, 2, \cdots, N$ **do**
    if individual $X_{i,g}$ is among any estimation groups **then**
      Perform selected single-objective adaptation method to adapt $F_{i,g}$
    end if
  end for
  for each objective k, $k = 1, 2, \cdots, M$ **do**
    Calculate the arithmetic mean of associated F values in estimation group for kth objective. Denote it as F^{ref}_k.
  end for
  for each individual i, $i = 1, 2, \cdots, N$ **do**
    if individual $X_{i,g}$ is not within estimation groups **then**
      $S_i = 1/r^1_{i,g} + 1/r^2_{i,g} + \cdots + 1/r^M_{i,g}$.
      $F_{i,g} = \frac{(F^{ref}_1/r^1_{i,g} + F^{ref}_2/r^2_{i,g} + \cdots + F^{ref}_M/r^M_{i,g})}{S_i}$.
    end if
  end for
end for

The last phase is hybridization. Arithmetic mean of all the control parameter values associated with the members in one estimation group is computed as the estimated optimal parameter value for corresponding objective. Suppose we are adapting the scaling factor F in DE, then the estimated values are symbolized by F^{ref}_k, where $k = 1, 2, \ldots, M$ and M is the total number of objectives. With the estimated optimal parametric setups for each objective, the parameter values for remaining individuals can be generated according to their positions in objective space, which may reveal the

Algorithm 2 Procedure of Decomposition-based Version

Random Initialization
for each individual i, $i = 1, 2, \cdots, N$ **do**
  Assignment of weight vector $\lambda^i = (\lambda_1^i, \lambda_2^i, \ldots, \lambda_M^i)$
end for
for $g = 1, 2, \cdots, G_{\max}$ **do**
  Reproduction operations (e.g., Crossover, Mutation)
  Update of successful F sets if needed
  Survival Selection
  for each objective k, $k = 1, 2, \cdots, M$ **do**
    Clear corresponding estimation group.
    Sorting the population w.r.t the assigned weights on kth objective.
    for each individual i, $i = 1, 2, \cdots, N$ **do**
      The rank of individual $X_{i,g}$ is recorded as $rw_{i,g}^k$.
      if $rw_{i,g}^k \leq 10\% \cdot N$ **then**
        $X_{i,g}$ is added into the estimation group for kth objective.
      end if
    end for
  end for
  for each individual i, $i = 1, 2, \cdots, N$ **do**
    if individual $X_{i,g}$ is among any estimation groups **then**
      Perform selected single-objective adaptation method to adapt $F_{i,g}$
    end if
  end for
  for each objective k, $k = 1, 2, \cdots, M$ **do**
    Calculate the arithmetic mean of associated F values in estimation group for kth objective. Denote it as F_k^{ref}.
  end for
  for each individual i, $i = 1, 2, \cdots, N$ **do**
    if individual $X_{i,g}$ is not within estimation groups **then**
      $F_{i,g} = \lambda_1^i \times F_1^{ref} + \lambda_2^i \times F_2^{ref} + \cdots + \lambda_M^i \times F_M^{ref}$.
    end if
  end for
end for

weights on optimizing each objective. In order to extract the position information in objective space, the concept of "sub-rank" is employed. Sub-rank is defined as a vector comprising the ranks of an individual in each seperate objective among current population. For example, if the sub-rank of one individual is (1, 1) in a two-objective problem, it means that this individual has the best fitness value for both objectives. Now we symbolize the sub-rank for individual $X_{i,g}$ by $R_{i,g} = (r_{i,g}^1, r_{i,g}^2, \ldots, r_{i,g}^M)$, where i is the index for this individual and g is the number of current generation. Following the assumption that we are adapting F in DE, the $F_{i,g}$ associated with $X_{i,g}$ will be updated as below:

$$S_i = 1/r_{i,g}^1 + 1/r_{i,g}^2 + \cdots + 1/r_{i,g}^M \tag{9}$$

$$F_{i,g} = \frac{(F_1^{ref}/r_{i,g}^1 + F_2^{ref}/r_{i,g}^2 + \cdots + F_M^{ref}/r_{i,g}^M)}{S_i} \tag{10}$$

It is notable that the parameter values associated with parents are always used in reproduction operations (e.g., crossover and mutation), and the offspring will inherit parents' associated control parameters before conducting survival selection.

To present a clear picture of how proposed framework operates, pseudo-codes of the implementation procedure (adapting F) are provided in Algorithm 1.

3.2.2 *Underlying Rationale*

The fundamental principle for individual-level parameter adaptation is that suitable control parameters tend to make their associated solutions more likely to survive and better solutions help to propagate their associated parameters [3, 27]. Therefore, the distribution of parameter values for one estimation group, which emphasize more on optimizing one particular objective, may reveal the parametric requirement of this objective. Calculation of the arithmetic mean within one estimation group is a reasonable way to approximate the optimal parameter values for corresponding objective. The advantages of considering all the parameter values within one group include making the estimation more reliable and alleviating the influence of possible outliers.

The role of adaptation phase is to help our framework exploring effective control parameters. Since the members in estimation groups focus much more on optimizing one particular objective, employing a single-objective tuning method that is proved to be effective will facilitate a more efficient adaptation for this objective. The parametric sensitivity issue for a MO problem has be converted into several single-objective tuning tasks. The adapted control parameters reveal the parametric requirements for each objective, thereby providing useful information for further analysis.

During the hybridization phase, all the individuals except for those in estimation groups will be assigned with new control parameters based on their sub-ranks. As per (10), a relatively higher rank in kth objective will lead to a larger weight on corresponding F_k^{ref}, which reflects the preferred parametric setup of kth objective, during the calculation of new parameter values. In other words, the different weights on optimizing each objective are reflected during the parametric trade-off among different objectives. Each individual will utilize the most proper parametric setup for its current region in objective space and keep exploring with the existing weights on optimizing each objective. As a result, the diversity of the population is better maintained and the efficiency of exploration is improved.

3.3 Decomposition-based Version

The implementation of the general version will introduce additional sorting procedures, which has $O(MNlogN)$ computational complexity. M is the total number of objectives and N is the population size. The purpose of these sortings is to calculate the sub-ranks so that the weights on optimizing each objective can be decided. Actually in decomposition-based MOEAs, each individual has already be assigned with a weight vector that decides its optimization weights. To reduce the unnecessary time complexity of the framework, a simplified version for decomposition-based MOEAs is developed. Similar to the general version, the decomposition-based version is also divided into three phases.

In the grouping phase, the members of estimation groups are decided based on their weight vectors. The individuals with the highest 10% weights on one particular objective are selected to form the estimation group for this objective. The adaptation phase in decomposition-based version is identical with that in general version.

Table 1: Mean and Standard Deviation of the IGD Values for General Version (30 runs)

Problems	NSGA-II -jDE	NSGA-II JADE	NSGA-II MDE_pBX	NSGA-II -DE
	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)
UF1	0.0619	0.0696	0.0662	0.0603
(2-obj)	(0.0082)	(0.0104)	(0.0077)	(0.0162)
UF2	**0.0391**	**0.0405**	**0.0375**	0.0429
(2-obj)	**(0.0041)**	**(0.0035)**	**(0.0029)**	(0.0047)
UF3	**0.0957**	**0.121**	**0.1053**	0.1515
(2-obj)	**(0.0213)**	**(0.0333)**	**(0.0261)**	(0.0271)
UF4	**0.0694**	**0.0698**	**0.0673**	0.0723
(2-obj)	**(0.0065)**	**(0.0068)**	**(0.0066)**	(0.0078)
UF5	1.141	1.11	1.0741	0.8494
(2-obj)	(0.2105)	(0.2243)	(0.2038)	(0.1698)
UF6	**0.3808**	**0.3993**	**0.393**	0.4181
(2-obj)	**(0.0674)**	**(0.0841)**	**(0.091)**	(0.0819)
UF7	**0.0291**	**0.0344**	**0.028**	0.0389
(2-obj)	**(0.0032)**	**(0.0058)**	**(0.0035)**	(0.0422)
UF8	**0.131**	**0.1483**	**0.1443**	0.152
(3-obj)	**(0.0324)**	**(0.0373)**	**(0.0248)**	(0.03)
UF9	**0.1891**	**0.1931**	0.1995	0.1938
(3-obj)	**(0.0762)**	**(0.0653)**	0.0598	(0.0646)
UF10	**0.8013**	**0.8238**	**0.9876**	2.4308
(3-obj)	**(0.0954)**	**(0.1469)**	**(0.1704)**	(0.1848)

During the hybridization phase, the sub-ranks in the general version are replaced by weight vectors. Suppose that the weight vector assigned to individual $X_{i,g}$ (outside estimation group) is $(\lambda_i^1, \lambda_i^2, \ldots, \lambda_i^M)$, then the associated parameter (e.g., $F_{i,g}$) will be generated as below:

$$F_{i,g} = \lambda_1^i \times F_1^{ref} + \lambda_2^i \times F_2^{ref} + \cdots + \lambda_M^i \times F_M^{ref} \qquad (11)$$

The detailed structure of the decomposition-based version (adapting F) is demonstrated in Algorithm 2.

4. EMPIRICAL STUDY

4.1 Numerical Benchmarks

10 representative benchmark problems from CEC-09 Special Session and Competition [29] are selected to evaluate the performance of the proposed framework. Numerous types of Pareto Optimal Front are covered including convex, non-convex, disconnected points, disconnected lines, straight line, discontinuous surfaces and continuous surface. With these benchmarks, whether the framework is capable to handle different kinds of problems will be validated.

4.2 Experimental Setup

In the following experiments, the proposed general version and decomposition-based version are implemented into NSGA-II-DE [14] and MOEA/D-DE [14], respectively. The scaling factor F in these two MOEAs will be adapted via the new framework. The single-objective adaptation methods in jDE, JADE, MDE_pBX are employed respectively in the adaption phase as stated in Section.3. Thus, totally 6 variants of the proposed framework are tested, namely, NSGA-II-jDE, NSGA-II-JADE, NSGA-II-MDE_pBX, MOEA/D-jDE, MOEA/D-JADE, MOEA/D-MDE_pBX. The performances of the proposed variants are compared with those of original NSGA-II-DE and MOEA/D-DE. Recommended parametric setups in the original literatures are utilized for all the above algorithmic components (F values are fixed as 0.5 in original NSGA-II-DE and MOEA/D-DE). For all the testing algorithms, the population size is fixed as 100 for 2-objective benchmarks, 300 for 3-objective benchmarks, and the maximum number of function evaluations is set to 5×10^4 for 2-objective problems, 15×10^4 for 3-objective problems. Inverted Generational Distance (IGD) [4] is selected as the quantitative metric to evaluate the performance of algorithms in terms of both convergence and diversity. All of the simulations were done on an Intel (R) Core (TM) i7 machine with 16-GB RAM and 3.40-GHz speed.

Table 2: Mean and Standard Deviation of the IGD Values for Decomposition-based Version (30 runs)

Problems	MOEA/D -jDE	MOEA/D JADE	MOEA/D MDE_pBX	MOEA/D -DE
	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)
UF1	**0.042**	**0.0431**	**0.0422**	0.0475
(2-obj)	**(0.0078)**	**(0.0073)**	**(0.0099)**	(0.0372)
UF2	**0.0245**	**0.0236**	**0.0239**	0.0426
(2-obj)	**(0.0044)**	**(0.0061)**	**(0.0033)**	(0.0316)
UF3	**0.0943**	**0.0994**	**0.0706**	0.1513
(2-obj)	**(0.0667)**	**(0.0626)**	**(0.0402)**	(0.0688)
UF4	**0.0758**	**0.0825**	**0.0746**	0.0866
(2-obj)	**(0.0114)**	**(0.0171)**	**(0.0104)**	(0.0104)
UF5	**0.6891**	**0.6867**	**0.6577**	0.7643
(2-obj)	**(0.1603)**	**(0.1719)**	**(0.1747)**	(0.1307)
UF6	**0.3791**	**0.3712**	**0.3641**	0.4386
(2-obj)	**(0.2551)**	**(0.2275)**	**(0.2048)**	(0.2206)
UF7	**0.0267**	**0.0671**	**0.0409**	0.1018
(2-obj)	**(0.0398)**	**(0.1177)**	**(0.0747)**	(0.1648)
UF8	0.0964	0.0926	**0.0902**	0.0911
(3-obj)	(0.0193)	0.0183	**(0.0183)**	(0.0124)
UF9	0.1335	0.1085	0.1174	0.1065
(3-obj)	(0.0445)	(0.0484)	(0.0472)	(0.0452)
UF10	0.5899	0.5917	0.6447	0.5826
(3-obj)	(0.1273)	(0.1236)	(0.1181)	(0.0716)

4.3 Effectiveness of the Proposed Framework

Table 1 and Table 2 show the mean and standard deviation of the IGD values over 30 independent runs of each algorithm on each benchmark. The entries that outperform original NSGA-II-DE or MOEA/D-DE in terms of mean value are marked in boldface.

From Table 1, the general version of the proposed framework is able to improve the optimization performance of original NSGA-II-DE in most problems except for UF1 and UF5. For UF1, actually the recommended F value 0.5 in original NSGA-II-DE is already close to the optimal setting. The adaption in our framework may employ some less efficient parameter values, which lead to the relatively slower convergence speed compared to original NSGA-II-DE. The shape of POF in UF5 is disconnected points, which will lead to clustering of the population. As a result, the sub-ranks utilized in the general version cannot reveal the true positions of individuals in objective space. The efficiency of the proposed mechanisms is thus deteriorated. When the performances among three proposed variants are compared, no obvious difference is observed in most benchmarks and improvement over original NSGA-II-DE is achieved by all of them. This implies that the proposed framework is not sensitive to the selection of single-objective adaptation methods and is able to successfully apply them in solving multi-objective problems.

Based on the results in Table 2, the decomposition-based version presents better performance than original MOEA/D-DE in all the 2-objective problems. More specifically, all the three variants of decomposition-based version achieved superior results over original MOEA/D-DE in solving UF1-UF7. There is a significant difference between decomposition-based version and general version in optimizing UF5. It is because the individuals in decomposition-based version have been assigned with fixed weight vectors, which will decrease the chance of clustering during optimization. Another notable pattern is that no significant improvement is achieved by the decomposition-based version in solving 3-objective problems. This may result from the unevenly assignment of weight vectors for 3-objective problems in MOEA/D as described in [18]. The effectiveness of parameters generated in hybridization phase will be alleviated because of this issue.

4.4 Behaviors of the Proposed Framework

In order to further investigate the detailed behaviors of the proposed framework, the movement of F_k^{ref} and the number of successful F values during optimization are presented in this subsection.

Figure 1 plots the F_k^{ref} values over generations for each objective on UF3 and UF10. All the 6 variants are tested for comparison. As stated before, F_k^{ref} actually represents the estimated optimal F values for kth objective. From Figure 1, the estimated results show different patterns among three single-objective adaptation methods. The distributions of F_k^{ref} values in jDE-based variants are obviously more widespread compared to those of JADE-based and MDE_pBX-based variants. This is because the tuning mechanisms in jDE involves a uniform randomization process, which will generate a wide variety of values. The F_k^{ref} values for JADE-based variants will converge more easily. This implies that the adaptation mechanisms in JADE is relatively weak in exploring new parameter values. The MDE_pBX-based variants presents a good trade-off between exploitation and exploration during adaptation of parameter values. From the perspective of MOEAs, the behaviors of the general version and the decomposition-based version provide similar patterns, which result from the same underlying rationales of these two versions. From the perspective of problems, all the objectives in UF3 and UF10 have the similar parametric requirements. It is difficult to observe the difference between multiple objectives. Actually this is common for the existing MO benchmark problems, which do not intentionally set disparate parametric requirements for each objective.

Figure 2 plots the number of successful F values over generations for each objective on UF3 and UF10. An F value is defined to be "successful" for kth objective when the associated offspring outperforms its parent in kth objective. The total number of successful F values for each objective in one generation will thus reveal the efficiency of adopted adaptation mechanisms on optimizing corresponding objectives. From Figure 2, the general version presents higher number of successful F values than the decomposition-based version in most cases. This indicates that the employment of sub-ranks help increase the efficiency of the adaptation. The patterns among different objectives are almost the same within one problem. It reflects that there is no bias in the proposed framework during the optimization of multiple objectives. A decreasing trend can be observed in most plots. This is because the improvement of population quality during later optimization stage becomes more difficult compared with earlier stages. All the three single-objective adaptation methods present similar results when implemented in the general version. However, the MOEA/D-MDE_pBX variant shows significantly higher overall success rate on UF3 compared with the other two decomposition-based variants. The previous results in Table 2 also demonstrate that MOEA/D-MDE_pBX performs significantly better than the jDE-based and JADE-based variants in solving UF3. This implies that the successful tuning of F values plays an very important role in improving the performance of algorithms.

5. CONCLUSIONS

This paper has proposed a new framework for self-adapting control parameters in multi-objective optimization. The proposed mechanisms are able to employ existing SO adaptation methods for optimizing multiple objectives. Efficient adaption from an objective-based perspective is achieved by utilizing position information in objective space. Successful implementations with two types of MOEAs and three exiting SO tuning approaches validate the effectiveness of the proposed framework.

In future work, designing a mechanism for dynamically switching among different SO adaptation methods is expected to further enhance the robustness of the framework. Introduction of more tuning approaches and implementation into more MOEAs will also be considered.

6. ACKNOWLEDGMENTS

This work was supported by the Singapore Ministry of Education Academic Research Fund Tier 1 under the project R-263-000-A12-112.

7. REFERENCES

[1] H. A. Abbass. The self-adaptive pareto differential evolution algorithm. *Congress on Evolutionary Computation 2002*, 1:831–836, 2002.

[2] J. Bader and E. Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, March 2011.

[3] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *Evolutionary Computation, IEEE Transactions on*, 10(6):646–657, Dec 2006.

[4] C. A. C. Coello and N. C. Cortes. Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*, 6(2):163–190, 2005.

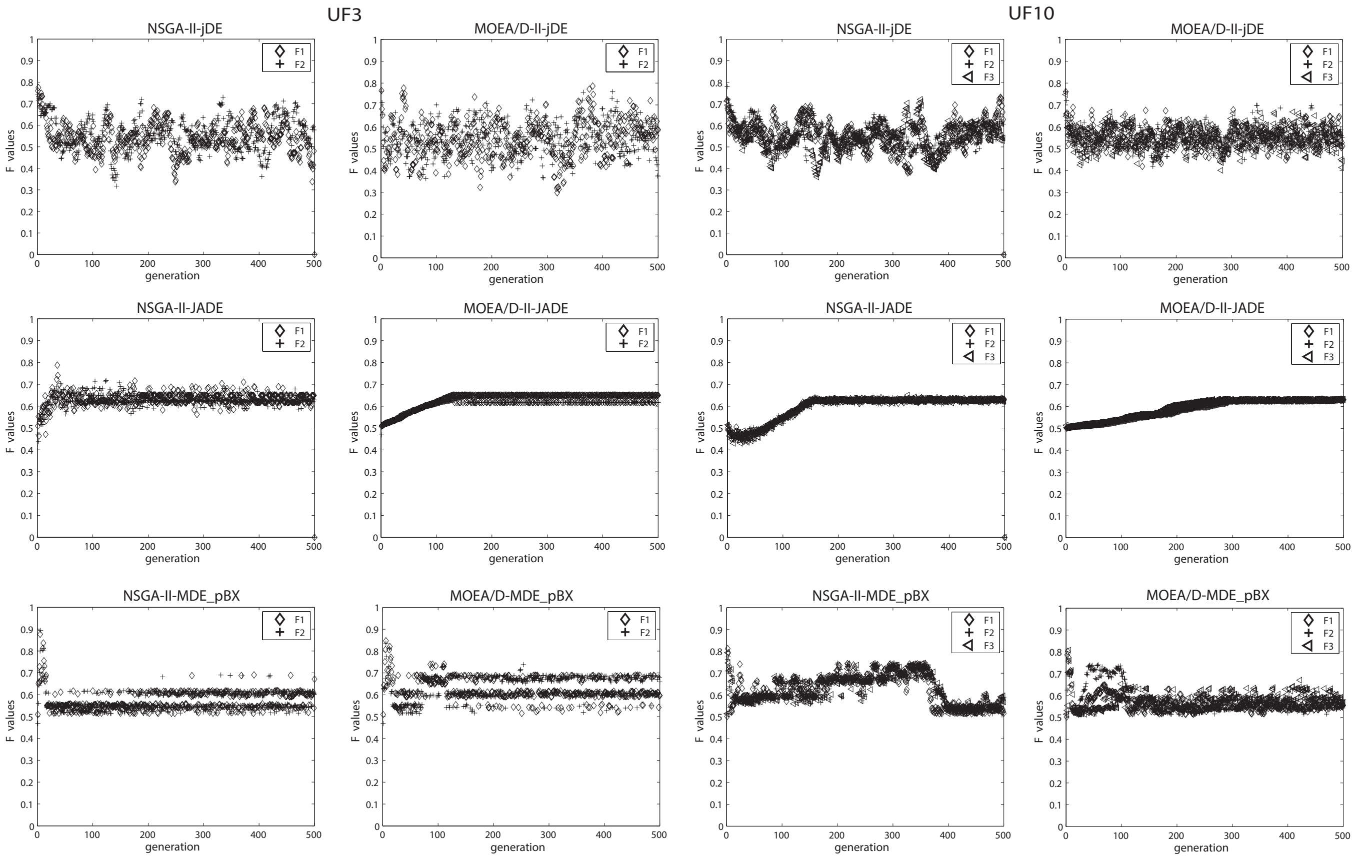

Figure 1: This figure plots the F_k^{ref} values over generations for each objective on UF3 (2-objective) and UF10 (3-objective). Vertical axis indicates the values for each F_k^{ref}, and horizontal axis represents the number of current generation. Different symbols are utilized to differentiate the F_k^{ref} for each objective.

[5] C. A. C. Coello and M. Reyes-Sierra. Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research*, 2, 2006.

[6] S. Das and P. N. Suganthan. Differential evolution: A survey of the state-of-the-art. *Evolutionary Computation, IEEE Transactions on*, 15(1):4 –31, feb. 2011.

[7] S. Dasgupta, S. Das, A. Biswas, and A. Abraham. On stability and convergence of the population-dynamics in differential evolution. *AI Commun.*, 22(1):1–20, 2009.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[9] A. W. Iorio and X. Li. Solving rotated multi-objective optimization problems using differential evolution. *Advances in Artificial Intelli-gence, Proceedings*, 3339:861–872, 2004.

[10] S. Islam, S. Das, S. Ghosh, S. Roy, and P. Suganthan. An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(2):482–500, April 2012.

[11] N. Iwasaki, K. Yasuda, and G. Ueno. Dynamic parameter tuning of particle swarm optimization. *IEEJ Transactions on Electrical and Electronic Engineering*, 1(4):353–363, 2006.

[12] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4, Nov 1995.

[13] S. Kukkonen and J. Lampinen. GDE3: The third evolution step of generalized differential evolution. *IEEE Congress on Evolutionary Computation (CEC'2005)*, 1:443–450, 2005.

[14] H. Li and Q. Zhang. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 13(2):284–302, 2009.

[15] J. Liu and J. Lampinen. A fuzzy adaptive differential evolution algorithm. In *TENCON '02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering*, volume 1, pages 606–611, Oct 2002.

[16] E. Mezura-Montes, M. Reyes-Sierra, and C. A. C. Coello. Multi-objective optimization using differential evolution: A survey of the state-of-the-art. *Advances in Differential Evolution , Studies in Computational Intelligence*, 143:173–196, 2008.

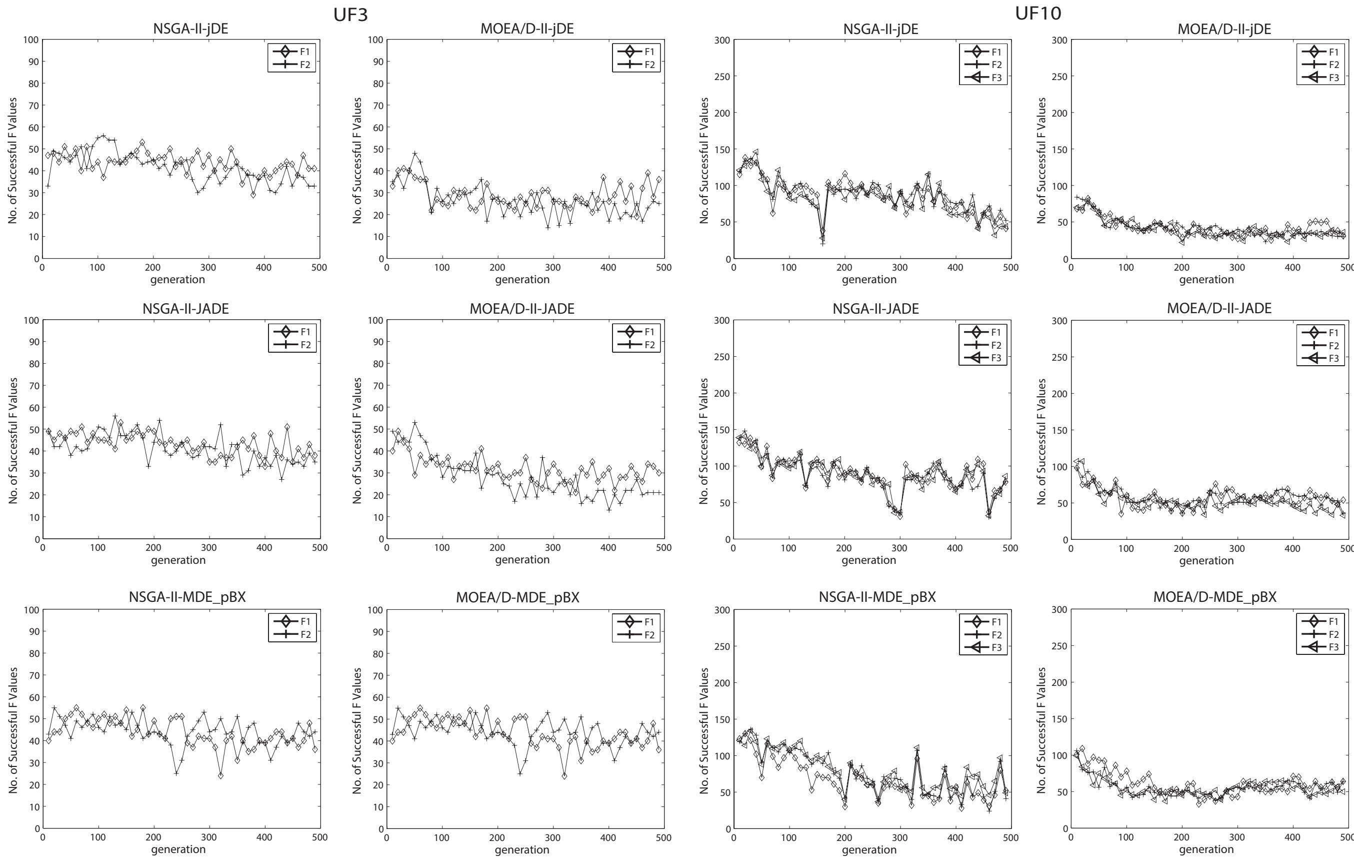

Figure 2: This figure shows the number of successful F values over generation for each objective on UF3 (2-objective) and UF10 (3-objective). Vertical axis indicates the number of successful F values for corresponding objectives, and horizontal axis represents the number of current generation. The different scales of vertical axes are due to the different population size for UF3 and UF10.

[17] M. E. H. Pedersen and H. Laboratories. Good parameters for particle swarm optimization. In *Technical Report*, 2010.

[18] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu. Moea/d with adaptive weight adjustment. *Evolutionary Computation*, 22(2):231–264, 2014.

[19] A. K. Qin, V. L. Huang, and P. Suganthan. Differential evolution algorithm with strategy adaptation for global numerical optimization. *Evolutionary Computation, IEEE Transactions on*, 13(2):398–417, April 2009.

[20] Y. Shi and R. Eberhart. Fuzzy adaptive particle swarm optimization. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 1, pages 101–106 vol. 1, 2001.

[21] R. Storn and K. V. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Golbal Optimization, Journal of*, 11, no. 4:341–359, 1997.

[22] J. Teo. Exploring dynamic self-adaptive populations in differential evolution. *Soft Computing*, 10(8):673–686, 2006.

[23] G. Xu. An adaptive parameter tuning of particle swarm optimization algorithm. *Applied Mathematics and Computation*, 219(9):4560 – 4569, 2013.

[24] Z. Yang, K. Tang, and X. Yao. Self-adaptive differential evolution with neighborhood search. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 1110–1116, June 2008.

[25] K. Yasuda, K. Yazawa, and M. Motoki. Particle swarm optimization with parameter self-adjusting mechanism. *IEEJ Transactions on Electrical and Electronic Engineering*, 5(2):256–257, 2010.

[26] D. Zaharie. On the explorative power of differential evolution. In *Proc. 3rd Int. Workshop Symbol. Numer. Algorithms Sci. Comput.*, Timisoara, Romania, Oct. 2-4 2001.

[27] J. Zhang and A. Sanderson. Jade: Adaptive differential evolution with optional external archive. *Evolutionary Computation, IEEE Transactions on*, 13(5):945–958, Oct 2009.

[28] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on*, 11(6):712–731, 2007.

[29] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari. Multiobjective optimization test instances for the CEC 2009 special session and competition. *Technical Report*, 2008.