A Multiagent Evolutionary Framework based on Trust for Multiobjective Optimization

Siwei Jiang Jie Zhang Yew Soon Ong
School of Computer Engineering
Nanyang Technological University, Singapore
{sjiang1, zhangj, asysong}@ntu.edu.sg

ABSTRACT

In an Evolutionary Algorithm (EA) for optimization problems, candidate solutions to the problems are individuals in a population. They produce offsprings by taking evolutionary operators with user-specific control parameters. The challenge is then how to effectively select evolutionary operators and adjust control parameters from generation to generation and on different problems. We propose a novel multiagent evolutionary framework based on trust where each solution is represented as an intelligent agent, and evolutionary operators and control parameters are represented as services. Agents select services in each generation based on trust that measures the competency or suitability of the services for solving particular problems. Multiobjective Optimization Problems (MOPs) are used to showcase the value of our framework. Experimental studies on 35 benchmark MOPs show that our framework significantly improves the performance of the state-of-the-art EAs.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents; Multiagent systems

General Terms

Design; Algorithms

Keywords

Evolutionary Algorithm; Multiagent Systems; Trust and Reputation; Multiobjective Optimization

1. INTRODUCTION

Evolutionary Algorithms (EAs) [1] are generic population-based stochastic search techniques inspired by biological evolution of nature selection for solving optimization problems. In EAs, candidate solutions to the problems play the role of individuals in a population. They produce offsprings by taking *evolutionary operators* (such as crossover and mutation) with user-specific *control parameters*. EAs are well known by its generality and simplicity that they often perform well approximating solutions to all types of problems in many fields such as engineering, economics, robotics, etc. However, evolutionary operators and control parameters may vary for different problems. It is time-consuming to determine the operators and parameters by the trial-and-error procedure. In addition, the competency of operators may vary with generations. For example, crossover is often powerful in the earlier stage of EAs, but mutation is effective when the solutions are similar with each other in the later stage of EAs. The challenge of EAs is thus how to effectively select evolutionary operators and adjust control parameters from generation to generation and on different problems.

Different techniques have been proposed to select evolutionary operators and control parameters in EAs. Igel et al. [3] propose CMA-ES as an evolution strategy that automatically adjusts control parameters, including the step-size, matrix mean and covariance. The methods of SaDE [5] and CoDE [8] not only adjust control parameters but also select evolutionary operators. However, both SaDE and CoDE introduce some other parameters whose values are predefined according to previous studies, which limits their generality.

In this paper, we propose a novel multiagent evolutionary framework based on trust where each solution is represented as an intelligent agent, and the pairs of evolutionary operators and control parameters are represented as services. In our framework, the agents model the trustworthiness of the services, based on whether the agents' offsprings produced by using the services survive to the next generations, which represents the dynamic competency or suitability of the services from generation to generation and on particular optimization problems. The agents will then select the services with the probabilities correlated to the trustworthiness of the services. To demonstrate the value of our framework, we consider the challenges of Multiobjective Optimization Problems (MOPs) as a case study, whereas it is generally applicable to other optimization problems. Experimental results on 35 benchmark MOPs confirm that our framework significantly improves the performance of the state-of-the-art EAs. The present work thus represents a promising step towards the use of multiagent based paradigms in the design of novel EAs as composing of intelligent agents that adopt trust modeling techniques for selecting evolutionary operators and control parameters in multiagent-based EAs.

2. RELATED WORK

CMA-ES [3] is a representative evolution strategy (ES) to adaptively adjust control parameters for solving MOPs. In CMA-ES, only one type of evolutionary operator is used to produce offsprings by a Gaussian mutation. The mean

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

and variance of the Gaussian distribution are adjusted with an iteration procedure. Our framework not only adjusts control parameters but also selects evolutionary operators. SaDE [5] and CoDE [8] are the two representative algorithms proposed to select evolutionary operators and adjust control parameters for solving single objective optimization problems. In SaDE, operators and parameters are gradually self-adapted by learning from their previous experience in generating promising solutions. In each generation, operators and parameters are assigned to different individuals in the current population according to the selection probabilities learned from the previous generations. However, SaDE computes simple statistics on the experience only after each 50 generations. In our framework, the competency or suitability of evolutionary operators and control parameters (referred to as the trustworthiness of services) is modeled by cumulating all previous experience based on well established probabilistic modeling and in a dynamic manner. Also, SaDE adjusts control parameters based on a normal distribution with a predefined mean based on the authors' prior knowledge. The CoDE algorithm randomly combines three DE operators and three predefined parameters to generate offsprings. Although CoDE obtains better performance over SaDE, its setting of the control parameters relies on some prior knowledge. All in all, SaDE and CoDE both introduce some other parameters whose values are predefined based on previous studies on single objective optimization problems, which limits their generality to other problems, e.g. more complex Multiobjective Optimization Problems (MOPs). In contrast, our selection of operators and control parameters does not rely on any prior knowledge about the problems. In addition, we design our framework as a multiagent system where candidate solutions are represented as intelligent agents capable of learning, cooperation and adaptation. This design offers great flexibility and extendability for EAs to employ advanced multiagent technologies for solving complex optimization problems.

Multiagent technologies have recently been widely used to design Evolutionary Algorithms (EAs) for solving complex problems [6]. For example, Stonedahl et al. [7] propose a distributed multiagent-based Genetic Algorithm (GA) to study how the network density of connections and the interactions between agents affect the performance of the GA. In the work of Zhong et al. [11] for solving single objective optimization problems, every solution is considered as an agent and all agents live in a lattice-like environment. The actions of agents are advanced evolutionary operator (such as orthogonal crossover and self-learning operators), but the agents are not autonomous because they select actions only based on predefined probabilities. In our framework, agents autonomously select services by learning the trustworthiness of the services. Trust plays a crucial role in agent-based service selection [10, 9]. It is used by agents to measure the quality of services and select services of high quality. One particularly effective way of modeling trust is to use the collective opinions of all agents about the services. We adopt this method in our framework.

Thus, the contributions of our current work can be summarized as follows: 1) majority of the adaptive EAs have been proposed to work with single objective optimization problems. Our generic framework can also be adopted to solve MOPs and other complex optimization problems; 2) the few existing adaptive EAs for MOPs adjust only control parameters, whereas agents in our framework can also select evolutionary operators; 3) to the best of our knowledge, multiagent technologies have been adopted to design adaptive EAs for solving MOPs for the first time; 4) our framework is also the first attempt to consider the use of trust modeling for measuring the dynamic competency of evolutionary operators and control parameters.

3. BACKGROUND ON MOEA

We demonstrate our framework on solving Multiobjective Optimization Problems (MOPs) [1]. MOPs involve several conflicting objectives to be optimized simultaneously. A minimization of MOPs can be stated as follows:

$$\begin{array}{rcl} min \;\; \mathcal{F}(\vec{x}) & = & (f_1(\vec{x}), \ldots, f_m(\vec{x})) \\ s.t. \;\; g(\vec{x}) & \leq & 0, \; h(\vec{x}) \; = \; 0, \; \vec{x} \; \in \; \Omega \end{array} \tag{1}$$

where $\vec{x} = (x_1, \ldots, x_D)$, Ω is decision (*variable*) *space*, R^m is *objective space*, and $\mathcal{F} : \Omega \rightarrow R^m$ consists of m real-valued objective functions with constraints $g(\vec{x}) \leq 0, h(\vec{x}) = 0$, and the feasible solution space is $\Omega = \Pi_{i=1}^{D}[LB_i, UB_i]$.

The challenge of MOPs is to find a *Pareto set* (PS) including non-dominated solutions which are evenly scattered along *Pareto front* (PF). Multiobjective Evolutionary Algorithms (MOEAs) have been well established as efficient approaches to solve various MOPs [1].

In MOEAs, the first population of solutions is randomly generated as $X_g = \{\vec{x}_{i,g} | i = 1, \ldots, NP, g = 0\}$, where NP is the population size and g is the generation index. The next population is produced by evolutionary operators. We take the "DE/rand/1/bin" operator as an example. At first, the operator generates a vector $\vec{v}_{i,g}$ base on population X_g.

$$\vec{v}_{i,g} = \vec{x}_{r1,g} + F \cdot (\vec{x}_{r2,g} - \vec{x}_{r3,g}) \tag{2}$$

where $r1, r2, r3 \in [1, NP]$ are random integer numbers and $r1 \neq r2 \neq r3 \neq i$. The control parameter F is the scaling factor which amplifies or shrinks the difference vectors.

After that, "DE/rand/1/bin" applies the binomial crossover operation to produce the offspring vectors:.

$$\begin{array}{c} U_g = \{\vec{u}_{i,j,g} | i = 1, \ldots, NP, j = 1, \ldots, D\} \\ \vec{u}_{i,j,g} = \begin{cases} \vec{v}_{i,j,g} & \text{if} \;\; rand_j(0,1) \leq CR \;\; \text{or} \;\; j = j_{rand} \\ \vec{x}_{i,j,g} & \text{otherwise.} \end{cases} \end{array} \tag{3}$$

where $rand_j(0,1) \in [0,1]$ is a uniformly distributed random number, $j_{rand} \in [1, D]$ is a randomly chosen integer. If $\vec{u}_{i,j,g} < LB_j$, it is set to LB_j, if $\vec{u}_{i,j,g} > UB_j$, set to UB_j. The control parameter CR is the probability for crossover.

Then, MOEAs select part of offsprings to enter the next generation ($\vec{u}_{i,g} \rightarrow X_{g+1}$). MOEAs can be generally categorized into two major classes: decomposition-based (called MOEA/D) [4] and Pareto dominance-based MOEAs [2, 12].

- In MOEA/D, $\vec{u}_{i,g} \rightarrow X_{g+1}$ if $\vec{u}_{i,g} \succeq \vec{x}_{j,g+1}$ ($\forall \vec{x}_{j,g+1} \in X_{g+1}$)[1] under, for example, Tchebycheff approach [4].
- In Pareto dominance-based MOEAs, $\vec{u}_{i,g} \rightarrow X_{g+1}$ if $\vec{u}_{i,g} \succeq \vec{x}_{j,g+1}$ ($\forall \vec{x}_{j,g+1} \in X_{g+1}$) under, for example, crowding distance (NSGAII) [2] or neighborhood density estimator (SPEA2) [12].

The performance of MOEAs is determined by the operators and their parameters (i.e. the operator "DE/rand/1/bin", and parameters F and CR in the operator mentioned above). The purpose of our framework is to select proper evolutionary operators and control parameters in EAs (i.e. MOEAs).

[1]"$\succeq$" means "be better than or equal".

4. OUR FRAMEWORK

In MOEAs, solutions in each generation produce offsprings by performing evolutionary operators with some control parameters. A plenty of effective evolutionary operators have been proposed, such as "DE/rand/1/bin", "DE/rand/2/bin", "DE/current-to-rand/1/bin" [8], Simulated Binary Crossover (SBX), and Polynomial mutation [2]. These operators, configured with different control parameters, exhibit distinguishing competence on different MOPs. The offsprings produced by some operators and parameters may be able to survive to the next generation, but some offsprings cannot.

In our multiagent evolutionary framework, each solution is represented as an agent. The pairs of evolutionary operators with corresponding control parameters are represented as services. In each generation, an agent selects a service to produce a new offspring agent (i.e., by Equations 2 and 3), which is also a solution. The new offspring agent competes with other agents in the environment. If the offspring agent can survive to the next generation, it means that the service provides a positive outcome, otherwise, the service provides a negative outcome. The trustworthiness of services can be used to represent the competency of the services in producing positive outcomes. The larger number of outcomes a service can produce, the more suitable the service is to solve the given problem. Thus, agents in our framework model the trustworthiness of the services based on the number of positive and negative outcomes provided by the services in the past generations. The modeling results will be used by the agents to make decisions on which services to consume.

4.1 Probabilistic Modeling of Trustworthiness

The trustworthiness of services is normally modeled based on the number of positive and negative outcomes produced by them in the past. If we define s as the number of positive outcomes and f as the number of negative outcomes provided by a service S, formulated as follows:

$$\begin{cases} s = s + 1 & \text{if } \vec{u}_{i,g} \rightarrow X_{g+1} \\ f = f + 1 & \text{otherwise} \end{cases} \tag{4}$$

where $\vec{u}_{i,g} \rightarrow X_{g+1}$ means that the offspring $\vec{u}_{i,g}$ produced in the generation g by the service can survive to the next generation $g+1$. Whether $\vec{u}_{i,g} \rightarrow X_{g+1}$ is determined based on different methods in MOEAs (see Section 3).

Beta distribution is commonly used to model the distribution of a random variable representing the unknown probability of a binary event. The Beta probability density functions (PDF) of service S can then be formulated as:

$$\text{Beta}(p(S)|\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p(S)^{\alpha-1}(1-p(S))^{\beta-1} \tag{5}$$

where $0 \leq p(S) \leq 1$ and $\alpha, \beta > 0$ with the restriction that $p(S) \neq 0$ if $\alpha < 1$ and $p(S) \neq 1$ if $\beta < 1$.

The trustworthiness of S is then the probability expectation value of the Beta distribution, which represents the relative frequency of positive outcomes in future events [10].

$$T(S) = \frac{\alpha}{\alpha+\beta}, \text{ where } \alpha = s+1, \ \beta = f+1 \tag{6}$$

4.2 Trustworthiness of Service

In this paper, we model the trustworthiness of services by adopting probabilistic modeling introduced in the previous section. One thing to note here is that MOEAs generally involve much randomness. A evolutionary operator configured with the same control parameters may still generate different offsprings because of the random values of $r1$, $r2$, $r3$ and $rand_j(0,1)$ in Equations 2 and 3. Due to this randomness, the trustworthiness of a service cannot be accurately estimated by a small number of outcomes produced by the service for a particular family of agents (a solution and its offsprings). Instead, in our framework, it is modeled based on the outcomes produced by the service for all agents in the past generations, which is referred to as reputation [10].

A service is represented by a evolutionary operator and some control parameters. The evolutionary operator can be any operator from a list of operators $O = \{O_1, O_2, \ldots, O_{|O|}\}$ proposed in MOEAs, where $|O|$ is the number of available evolutionary operators. Given a specific operator $O_k \in O$ in the service, there will be a set of control parameters $C^k = \{C_l^k | l = 1, \ldots, |C^k|\}$ associated with the operator O_k, where $|C^k|$ is the number of control parameters. For example, the operator "DE/rand/1/bin" has two control parameters (CR and F) associated with it (see Section 3). Assume that a parameter C_l^k takes a continuous value in the range as $C_l^k \in [0,1]$. In order to effectively learn the performance of a control parameter, we divide the range $[0,1]$ into a set of q disjoint segments as $L = \{[0, \frac{1}{q}), [\frac{1}{q}, \frac{2}{q}), \cdots, [\frac{q-1}{q}, 1]\}$. Thus, a service can be formally defined as a tuple (O_k, C^k) where $C^k = \{C_l^k | C_l^k = L(C_l^k), l = 1, \ldots, |C^k|\}$ and $L(C_l^k)$ is one of the segments in L for the parameter C_l^k. In another word, a service is a tuple of a evolutionary operator and a set of segments for corresponding control parameters. Here, we do not distinguish a parameter from its segment for simplicity.

For the service (O_k, C^k), we first compute the trustworthiness of the operator O_k. It is modeled based on the number of positive and negative outcomes generated by the agents performing this operator in the past generations. The total number of positive and negative outcomes up to the current generation g is aggregated as follows:

$$\begin{cases} s_g(O_k) &= (1-\eta) \cdot s_{g-1}(O_k) + \eta \cdot N_{g,s}(O_k) \\ f_g(O_k) &= (1-\eta) \cdot f_{g-1}(O_k) + \eta \cdot N_{g,f}(O_k) \end{cases} \tag{7}$$

where $N_{g,s}(O_k)$ and $N_{g,f}(O_k)$ are the number of positive and negative outcomes produced by the agents performing the operator O_k in the current generation g, respectively. The parameter $0 \leq \eta \leq 1$ is to determine how much to consider the current and historical information, where $\eta = 0$ means that only the historical information is considered, whereas $\eta = 1$ only the current information is utilized. After having $s_g(O_k)$ and $f_g(O_k)$, the trustworthiness of the operator O_k in the current generation g, $T_g(O_k)$, can then be computed according to Equation 6.

In general, one operator is suitable for some specific types of problems, but may not work well for other types. Even for the same problem, the competency of the operator may vary in different generations. For example, the operator "DE/ran/1/bin" is suitable to multi-modal problems, which has slow convergency in the earlier stage but exhibits strong exploration in the later stage of EAs. Based on this phenomenon, the trustworthiness of the operator needs to reflect the varying competency of the operator under the condition where trust is hard to build up, but easy to lose.

The aggregation function in Equation 7 is then revised as:

$$\begin{cases} s_g &= (1-T_{g-1}) \cdot s_{g-1} + T_{g-1} \cdot N_{g,s} \\ f_g &= (1-T_{g-1}) \cdot f_{g-1} + T_{g-1} \cdot N_{g,f} \end{cases} \tag{8}$$

where O_k is dropped out for clarity and T_{g-1} is the trustworthiness of operator O_k in generation $g-1$. Equation 8 has two important advantages. It does not have predefined parameters, compared to Equation 7 that has the parameter η. Equation 8 also satisfies the above mentioned condition. When the trustworthiness of the operator in the last generation $g-1$, $T_{g-1}(O_k)$ is low, the operator needs more positive outcomes $N_{g,s}(O_k)$ to build up its trust in the current generation g. When $T_{g-1}(O_k)$ is high, $1-T_{g-1}(O_k)$ is low, meaning that the less consideration will be given to historical information. The trustworthiness of the operator $T_g(O_k)$ will be easy to decline when the number of negative outcomes in the current generation $N_{g,f}(O_k)$ is large.

For the service (O_k, C^k), we then compute the trustworthiness of each parameter in C^k (i.e. the value range segment corresponding to each parameter). When computing the trustworthiness a parameter, we also need to consider the operator the parameter is associated with. Take the parameter C_l^k as an example. The trustworthiness of C_l^k associated with O_k in the current generation g, denoted as $T_g(C_l^k|O_k)$, can be calculated in the similar way as calculating the trustworthiness of the operator O_k (Equation 8), by counting the numbers of positive and negative outcomes produced by the operator O_k with the parameter C_l^k, which are $N_{g,s}(C_l^k|O_k)$ and $N_{g,f}(C_l^k|O_k)$ respectively.

After having the trustworthiness of the evolutionary operator O_k, which is $T_g(O_k)$, and each control parameter C_l^k given O_k, which is $T_g(C_l^k|O_k)$, we can then compute the trustworthiness of the service (O_k, C^k) by assuming the control parameters are independent, as follows:

$$T_g(O_k, C^k) = T_g(O_k) \cdot \prod_{l=1}^{|C^k|} T_g(C_l^k|O_k) \tag{9}$$

4.3 Trust-based Service Selection

In our framework, agents select services based on the computed trust results of the services. In order to balance between exploitation and exploration, services are selected in a probabilistic manner where the probability for a service to be selected is proportional to its trust. More formally, there are $\sum_k^{|O|} |C^k| \cdot m$ services in total because there are $|O|$ evolutionary operators, each operator O_k is associated with $|C^k|$ control parameters, and each parameter is represented by one of the q value range segments. The probability for service (O_k, C^k) with the trust $T_g(O_k, C^k)$ in the current generation g to be selected in the next generation $g+1$ is:

$$p(O_k, C^k) = \frac{T_g(O_k, C^k)}{\sum^{\sum_k^{|O|} |C^k| \cdot m} T_g(O_k, C^k)} \tag{10}$$

Note that after an agent selects a service, e.g. (O_k, C^k), each control parameter in C^k, e.g. C_l^k, is a value range segment in L, not a specific value. In order for the service to be used by the agent to produce an offspring, a specific value for the parameter C_l^k is needed. We assume that the values of the parameter C_l^k follow a normal distribution in the range of $L(C_l^k)$ as $\mathrm{Normal}(\mu_g(C_l^k), \sigma)$ where $\mu_g(C_l^k)$ and $\sigma = \frac{1}{3q}$ are the mean and standard deviation, respectively, and $C_l^k \in [0,1]$. The mean $\mu_g(C_l^k)$ is calculated as follows:

$$\begin{aligned}\mu_g(C_l^k) &= (1 - T_{g-1}(C_l^k|O_k)) \cdot \mu_{g-1}(C_l^k) \\ &\quad + T_{g-1}(C_l^k|O_k) \cdot \mathrm{Mean}(V_g(C_l^k|O_k))\end{aligned} \tag{11}$$

where $V_g(C_l^k|O_k)$ is the set of the values of the parameter C_l^k, which produces positive outcomes for the agent performing the operator O_k in the current generation g. $\mathrm{Mean}(V_g(C_l^k|O_k))$ is the mean of the values in $V_g(C_l^k|O_k)$. The rationale behind Equation 11 is that the effectiveness of the parameter C_l^k measured by $T_{g-1}(C_l^k|O_k)$, reflects the appropriation of its mean $\mu_g(C_l^k)$ up to the generation $g-1$. To cope with the dynamics of the effectiveness of $\mu_g(C_l^k)$, we formulate it in a similar spirit as Equation 8.

5. EXPERIMENTATION

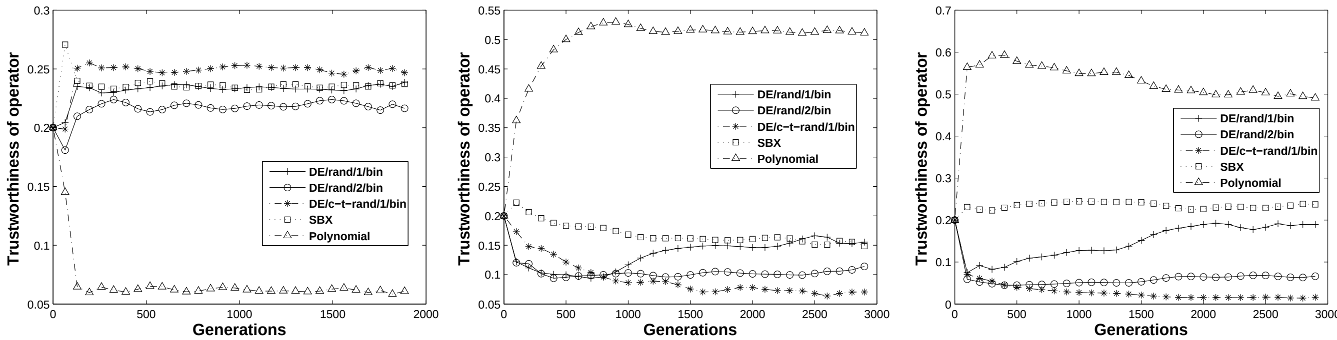
The experiments are carried out on jMetal 3.1[2], a Java-based framework aimed at facilitating the development of metaheuristics for solving MOPs. The benchmark problems include 35 test instances: 5 MOPs in the ZDTx family problems (ZDT1-4 and ZDT6 with 2 objectives), 7 MOPs in the DTLZx family problems (DTLZ1-7 with 3 objectives), and 23 MOPs in the CEC2009 MOEA competition. Among the problems used in the CEC2009 MOEA competition that involves unconstrained functions, UF1-7 have 2 objectives, UF8-10 3 objectives, and UF11-13 5 objectives. In addition, The problems CF1-10 have one constraint except CF6-7 have two constraints. The *decision variables* in the Pareto sets (PSs) of the ZDTx and DTLZx are independent, and those in the CEC2009 MOEA competition are dependent. The 35 MOPs have different geometrical shapes in *objective space* such as concave, convex, linear, discrete, uni-modal and multi-modal Pareto fronts (PFs).

The experimental settings are outlined as follows. The number of decision variables D used in ZDT1-3 is 30, $D=10$ in ZDT4 and ZDT6, $D=7$ in DTLZ1, $D=12$ in DTLZ2-6, $D=22$ in DTLZ7, $D=30$ in UF1-13, and $D=10$ in CF1-10. In MOEA/D, the population size NP is decided by the number of weight vectors C_{H+m-1}^{m-1} (m is the number of objectives, H is a predefined integer). For problems with two objectives, $NP=100$ by setting $H=99$, $NP=153$ for tri-objective problems ($H=16$), $NP=715$ for five-objective problems ($H=9$). The other algorithms have the same population size as MOEA/D on different MOPs. We set the maximum number of function evaluations (FEs) to be $300,000$ and independent run times to be 30.

We compare with classic MOEAs (NSGAII [2], SPEA2 [12] and MOEA/D [4]). We also compare with the other approaches (CMA-ES [3], SaDE [5] and CoDE [8]) that select evolutionary operators and/or control parameters. Five evolutionary operators are considered, including "DE/rand/1/bin", "DE/rand/2/bin", "DE/current-to-rand/1/bin" [8], "SBX", and "Polynomial mutation" [2]. In NSGAII and SPEA2, the control parameters for SBX are set to $\eta_c = 20$ and $p_c = 0.9$, and those for Polynomial mutation are set to $\eta_m = 20$ and $p_m = 1/D$. In MOEA/D, the control parameters of the operator "DE/rand/1/bin" are set to $CR = 1.0$ and $F = 0.5$. The update approach used in decomposition-based MOEAs is the Tchebycheff. In our framework, the number of value range segments for the control parameters is set to $q=3$.

All the algorithms are evaluated by the hypervolume metric, which is strictly monotonic with regard to Pareto dominance [12]. The obtained results are compared using median values and interquartile range (IQR). In order to have statistically sound conclusions, the Wilcoxon rank sum test with 95% confidence level is conducted on the experiment results.

[2]http://jmetal.sourceforge.net



(a) DTLZ1 by NSGAII-T (b) UF1 by NSGAII-T (c) CF1 by NSGAII-T

Figure 1: The Trustworthiness of Operators $T_g(O)$ Derived by NSGAII-T on DTLZ1, UF1 and CF1

5.1 Improving MOEAs by Our Framework

In this experiment, we extend the classic MOEAs (NSGAII [2], SPEA2 [12] and MOEA/D [4]) by our framework that selects the five evolutionary operators and adjusts control parameters based on trust. The extended versions of NSGAII, SPEA2 and MOEA/D are NSGAII-T, SPEA2-T and MOEA/D-T, respectively. The purpose is to evaluate whether they can be improved by our framework.

Table 1: Statistical Comparison Results of Classic MOEAs versus Those Extended by Our Framework

	NSGAII-T	SPEA2	SPEA2-T	MOEA/D	MOEA/D-T
NSGAII	18/13/4	17/11/7	19/8/8	19/7/9	22/5/8
NSGAII-T		14/8/13	15/8/12	17/6/12	20/5/10
SPEA2			19/12/4	14/6/15	17/8/10
SPEA2-T				14/7/14	17/6/12
MOEA/D					22/10/3

Table 3 (in the end of the paper) shows the detailed experimental results, where each tuple reports the median and IQR of hypervolume over 30 independently runs on 35 MOPs with 300,000 FES. Table 1 shows the win/tie/lose ($w/t/l$) statistical results under the Wilcoxon rank sum test with 95% confidence level. Each tuple $w/t/l$ means that the algorithm at the corresponding column wins on w MOPs, ties on t MOPs, and loses on l MOPs, compared to the algorithm at the corresponding row. The results show that the $w/t/l$ values between the extended versions by our framework and the classic MOEAs and are 18/13/4, 19/12/4, 22/10/3, respectively. This indicates that our framework can significantly improve the performance of the classic MOEAs. We also see that MOEA/D-T is the most effective to solve MOPs than the classic MOEAs and the other extended MOEAs (NSGAII-T and SPEA2-T).

5.2 Comparison with Adaptive Approaches

In this experiment, we implement the other adaptive approaches for selecting evolutionary operators and/or control parameters (CMA-ES [3], SaDE [5], and CoDE [8]) to extend MOEA/D for solving MOPs. The purpose is to compare the effectiveness of them with our framework.

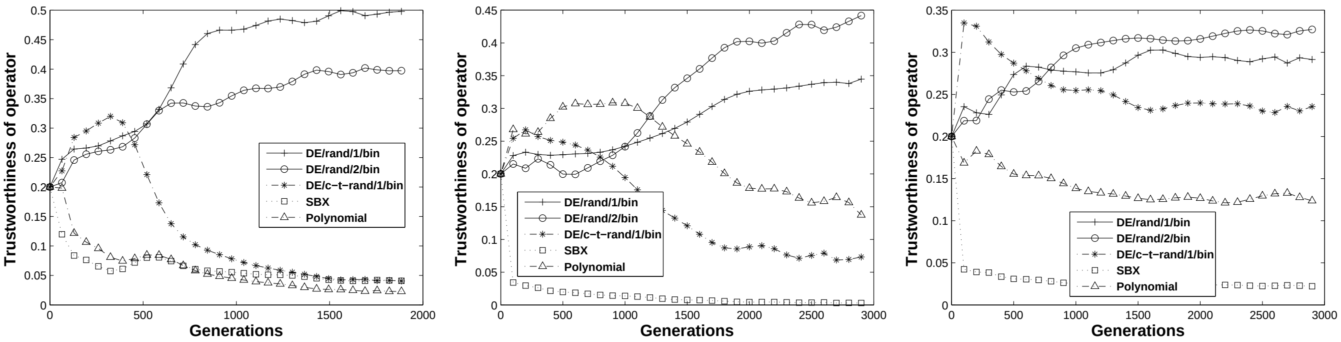
CMA-ES uses only one evolutionary operator (Gaussian mutation), and adjusts the mean and variance of Gaussian distribution in variable space. SaDE and CoDE select operators among "DE/rand/1/bin", "DE/rand/2/bin" and "DE/current-to-rand/1/bin". In SaDE, the control parameters are generated by normal distribution, where $\sigma_{CR} = 0.3$, $\mu_F = 0.5$ and $\sigma_F = 0.1$. SaDE introduces four predefined parameters, including the learning period of 50 generations. CoDE combines the three operators with a set of fixed parameter settings, including $[CR = 0.1, F = 0.1]$, $[CR = 1.0, F = 0.5]$ and $[CR = 0.2, F = 0.8]$. Our framework selects operators among "DE/rand/1/bin", "DE/rand/2/bin", "DE/current-to-rand/1/bin", "SBX" and "Polynomial mutation". To have a fair comparison, we implement MOEA/D-T3 that uses our framework to select among only the first three operators, which is the same as SaDE and CoDE. The algorithm MOEA/D-T' is implemented to use another version of our framework that models the trustworthiness of services with a fixed parameter $\eta = 0.3$ using Equation 7.

Table 4 (in the end of the paper) shows the detailed experimental results of comparing CMA-ES, SaDE and CoDE with MOEA/D-T3, MOEA/D-T' and MOEA/D-T, where each tuple reports the median and IQR of hypervolume over 30 independently runs on 35 MOPs with 300,000 FES. Table 2 shows the win/tie/lose ($w/t/l$) statistical comparison results of the six algorithms under the Wilcoxon rank sum test with 95% confidence level.

Table 2: Comparison Results for CMA-ES, SaDE, CoDE, MOEA/D-T3, MOEA/D-T', MOEA/D-T

	SaDE	CoDE	MOEA/D-T3	MOEA/D-T'	MOEA/D-T
CMA-ES	27/4/4	25/8/2	27/7/1	26/8/1	30/4/1
SaDE		10/17/8	17/13/5	14/17/4	21/8/6
CoDE			11/21/3	10/16/9	22/11/2
MOEA/D-T3				6/20/9	9/24/2
MOEA/D-T'					14/16/5

The $w/t/l$ values between MOEA/D-T3 and CMA-ES is 27/7/1, indicating that selecting evolutionary operators is beneficial for solving MOPs. The $w/t/l$ values between MOEA/D-T3 and SaDE is 17/13/5. Our MOEA/D-T3 not only has less predefined parameters than SaDE, but also is more effective than SaDE. The $w/t/l$ value between MOEA/D-T3 and CoDE is 11/21/3. MOEA/D-T3 is more effective than CoDE. CoDE fixes the two control parameters CR and F before the algorithm starts. MOEA/D-T3 learns the parameters as the algorithm progresses. It demonstrates that parameter learning based on trust in our framework is able to automatically adjust control parameters on different MOPs. The $w/t/l$ values between MOEA/D-T and CMA-ES, SaDE, CoDE, MOEA/D-T3, MOEA/D-T' are 30/4/1, 21/8/6, 22/11/2, 9/24/2 and 14/16/5, respectively. Our



(a) DTLZ1 by MOEA/D-T (b) UF1 by MOEA/D-T (c) CF1 by MOEA/D-T

Figure 2: The Trustworthiness of Operators $T_g(O)$ Derived by MOEA/D-T on DTLZ1, UF1 and CF1

MOEA/D-T is the best among the six algorithms. The selection on the two extra evolutionary operators "SBX" and "Polynomial mutation" does show advantages. MOEA/D-T outperforms MOEA/D-T', confirming that the dynamic modeling of the trustworthiness of services is more effective.

5.3 Effect of PSs on Operator Selection

An important factor that affects the searching ability of Pareto dominance-based MOEAs (e.g. NSGAII) in variable space is the interdependency among the decision variables in Pareto sets (PSs). In this experiment, we want to examine the effect of PSs on the selection of evolutionary operators. We evaluate NSGAII-T on the problems DTLZ1, UF1 and CF1. In DTLZ1, decision variables in PSs are independent, but in UF1 and CF1, the decision variables are interdependent. Figure 1 shows the trustworthiness of operators $T_g(O)$ in different generations. We can see that in Pareto dominance-based MOEAs, the evolutionary operators SBX and Polynomial mutation show great contributions. It is also evident that SBX does better in independent PSs (DTLZ1) than interdependent PSs (UF1 and CF1), whereas Polynomial mutation is more suitable to deal with the nonlinear variable dependencies than independent variables.

5.4 Trustworthiness of Evolutionary Operators

The operators "DE/rand/1/bin" and "DE/rand/2/bin" are quite similar. However, "DE/current-to-rand/1/bin" is different and it is formulated as follows:

$$\vec{v}_{i,g} = \vec{x}_{i,g} + rand() \cdot (\vec{x}_{r1,g} - \vec{x}_{i,g}) + F \cdot (\vec{x}_{r2,g} - \vec{x}_{r3,g}) \qquad (12)$$

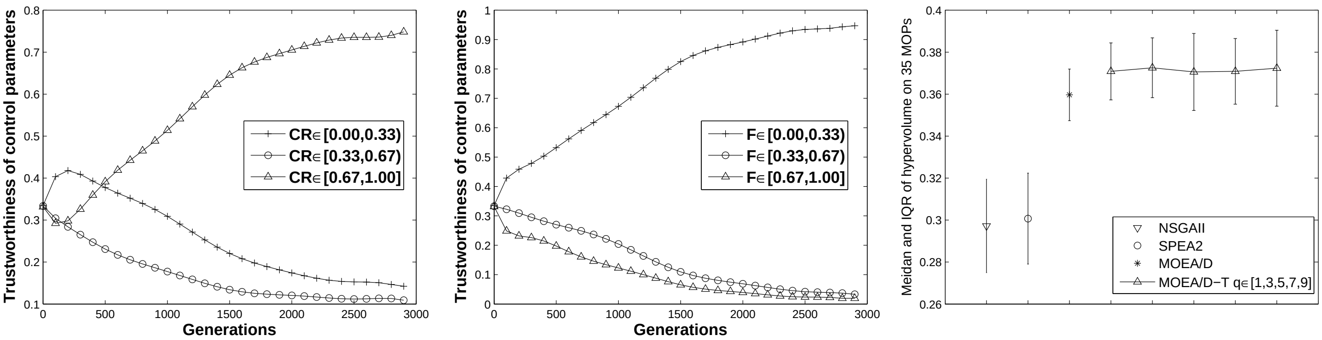
where $r1, r2, r3 \in [1, NP]$ are random integer numbers and $r1 \neq r2 \neq r3 \neq i$, $\vec{x}_i$ is the base vector, and $rand() \in (0, 1)$ is a uniform random value. Operator "DE/current-to-rand/1/bin" generates the new offsprings based on vector $\vec{x}_i$, whereas "DE/rand/1/bin" and "DE/rand/2/bin" search new agent in the global region.

The effectiveness (competency) of evolutionary operators is evaluated by the trustworthiness of them in our MOEA/D-T. In this experiment, we investigate the different competency of the operators in different generations on different MOPs. Figure 2 shows the trustworthiness of operators $T_g(O)$ on DTLZ1, UF1, and CF1 by MOEA/D-T in different generations. All results are means of 30 independent runs. We can see that the trustworthiness (competency) of the operators vary from generations to generations and on different problems. Under the decomposition-based MOEA method, "DE/rand/1/bin", "DE/rand/2/bin" and "DE/current-to-rand/1/bin" are more effective than "SBX" and "Polynomial". The trustworthiness of "DE/current-to-rand/1/bin" increases in the earlier stage then gradually decreases in the later stage, whereas the trustworthiness of "DE/rand/1/bin" and "DE/rand/2/bin" gradually increases as MOEA/D-T progresses. "DE/current-to-rand/1/bin" has the search bias based on the base vector ($\vec{x}_i$) and larger perturbation ($rand()$). In the earlier stage of MOEAs, it has good performance due to the biased search. But its performance gradually deteriorates in the later stage because of the uncertain perturbation. "DE/rand/1/bin" and "DE/rand/2/bin" do not prefer any search direction but they have strong exploration capability. The competency of them is low in the earlier stage because of their unbiased search. But in the later stage, they are more effective than "DE/current-to-rand/1/bin" due to the better exploration. Thus, the trustworthiness of the operators modeled by our framework well reflects their true competency.

5.5 Trustworthiness of Control Parameters

The effectiveness (suitability) of control parameters is also evaluated by the trustworthiness of them in MOEA/D-T. In this experiment, we investigate the different suitability of the control parameters in different generations. Figures 3(a) and 3(b) show the trustworthiness of the control parameters (different value range segments for CR and F respectively), $T_g(CR)$ and $T_g(F)$, on the problem UF1 by MOEA/D-T, where $\{CR, F \in [0, 1]\}$ is divided into three segments. Because of space limitation, we only show the results of the parameters for the operator "DE/rand/2/bin".

The trustworthiness of $CR \in [0.67, 1.00]$ is high in the later stage on UF1. The larger value of CR makes the operator to search in a broad region, and it is beneficial for MOEAs to maintain the population diversity. The trustworthiness of $F \in [0.00, 0.33)$ gradually increases on UF1. As the algorithm progresses, the agents (represent the solutions) spread more evenly. It means that the difference between agents (i.e., $\vec{x}_{r1,g} - \vec{x}_{r_2,g}$ in Equation 2) becomes larger. So, in the later stage, the operator "DE/rand/2/bin" needs to adjust the parameter F to be small for exploitation to search in a neighboring region. Thus, the trustworthiness of the control parameters modeled well reflects the varying competency of them in different generations.



(a) UF1 by MOEA/D-T (b) UF1 by MOEA/D-T (c) Effect of Control Parameters

Figure 3: (a, b) Trustworthiness of Parameters $T_g(CR|O)$ and $T_g(F|O)$ Derived by MOEA/D-T on UF1 and $O =$ "DE/rand/2/bin" in Different Generations; (c) MOEA/D-T with Different Values of Parameter q

5.6 The Effect of Parameter q

Our framework has only one predefined parameter q, which is the number of value segments for control parameters. To investigate the impact of this parameter setting, MOEA/D-T with $q = \{1, 3, 5, 7, 9\}$ are tested on 35 MOPs. Figure 3(c) shows the median and IQR of hypervolume derived from NSGAII, SPEA2, MOEA/D and MOEA/D-T over 30 independent runs. It is evident that MOEA/D-T is not sensitive to the setting of q. For all q values, MOEA/D-T outperforms NSGAII, SPEA2 and MOEA/D.

6. CONCLUSION AND FUTURE WORK

In this paper, a novel multiagent evolutionary framework based on trust is proposed to effectively select evolutionary operators and adjust control parameters (represented as services), for solving complex optimization problems (such as MOPs). In the framework, agents (representing solutions) automatically select services by modeling their trustworthiness based on the number of offsprings produced using them will survive to the next generation. Experiments carried out on 35 benchmark MOPs confirm that our framework significantly improves the performance of the classic MOEAs (NSGAII, SPEA2 and MOEA/D) and outperforms the other three adaptive approaches (CMA-ES, SaDE and CoDE).

For future work, we will examine our framework in a distributed multiagent system where only partial (local and neighboring) information about the outcomes of services is known to agents, towards the development of a distributed framework. We will also investigate the performance of our framework on other complex problems, such as constraint optimization, expensive optimization problems, etc.

Acknowledgment

This work is partially supported by the Media Development Authority of Singapore, Singapore-MIT GAMBIT Game Lab and the Center for Computational Intelligence (C2I) at the Nanyang Technological University, Singapore.

7. REFERENCES

[1] C. A. C. Coello. Evolutionary multi-objective optimization: a historical view of the field. *IEEE Computational Intelligence Magazine*, 1(1):28–36, 2006.

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

[3] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28, 2007.

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

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

[6] R. A. Sarker and T. Ray. *Agent-Based Evolutionary Search*. Springer, 2010.

[7] F. Stonedahly, W. Randyz, and U. Wilensky. Multi-agent learning with a distributed genetic algorithm: Exploring innovation diffusion on networks. In *Proceedings of the AAMAS Workshop on ALAMAS+ALAg*, 2008.

[8] Y. Wang, Z. Cai, and Q. Zhang. Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transaction on Evolutionary Computation*, 15(1):55–66, 2011.

[9] Y. Wang, J. Zhang, and J. Vassileva. Effective web service selection via communities formed by super-agents. In *Proceedings of the IEEE/WIC/ACM Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 2010.

[10] J. Zhang and R. Cohen. A comprehensive approach for sharing semantic web trust ratings. *Computational Intelligence*, 23(3):302–319, 2007.

[11] W. Zhong, J. Liu, M. Xue, and L. Jiao. A multiagent genetic algorithm for global numerical optimization. *IEEE Transactions on System, Man, Cybernetics, Part B: Cybernetics*, 34(2):1128–1141, 2004.

[12] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. *Technical report 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Zurich, Switzerland*, 2002.

Table 3: Hypervolume Median and IQR of NSGAII, NSGAII-T, SPEA2, SPEA2-T, MOEA/D, MOEA/D-T

MOPs	NSGAII	NSGAII-T	SPEA2	SPEA2-T	MOEA/D	MOEA/D-T
ZDT1	6.60E-01±3.15E-04−	6.60E-01±3.66E-04−	6.62E-01±1.66E-04≈	6.62E-01±4.65E-05+	6.61E-01±2.08E-04−	6.62E-01±4.51E-07
ZDT2	3.27E-01±3.07E-04−	3.27E-01±2.91E-04−	3.28E-01±7.83E-05≈	3.29E-01±3.85E-05+	3.28E-01±1.15E-04−	3.28E-01±1.21E-08
ZDT3	5.15E-01±7.64E-05+	5.15E-01±1.12E-04+	5.16E-01±8.76E-05+	5.16E-01±2.71E-05+	5.14E-01±2.95E-05−	5.14E-01±8.61E-06
ZDT4	6.61E-01±1.83E-04−	6.60E-01±2.94E-04−	6.62E-01±5.06E-05+	6.62E-01±1.67E-01≈	6.61E-01±2.66E-04−	6.62E-01±1.05E-04
ZDT6	3.98E-01±3.84E-04−	4.00E-01±3.16E-04−	4.01E-01±2.48E-04−	4.01E-01±2.03E-05+	4.01E-01±2.50E-07−	4.01E-01±6.21E-09
DTLZ1	7.78E-01±4.39E-03+	7.76E-01±3.63E-03+	7.97E-01±5.31E-04+	7.97E-01±2.52E-04+	7.61E-01±5.58E-04−	7.61E-01±1.22E-04
DTLZ2	3.93E-01±4.23E-03≈	3.97E-01±4.28E-03+	4.19E-01±1.42E-03+	4.29E-01±8.00E-04+	3.92E-01±1.30E-03≈	3.93E-01±2.67E-04
DTLZ3	3.99E-01±7.80E-03+	3.96E-01±4.96E-03+	4.28E-01±1.42E-03+	4.29E-01±8.24E-04+	3.91E-01±2.12E-03−	3.93E-01±3.15E-04
DTLZ4	3.95E-01±3.33E-03+	3.94E-01±4.08E-03+	4.14E-01±1.33E-03+	4.22E-01±5.10E-04+	3.95E-01±2.38E-03+	3.88E-01±2.04E-04
DTLZ5	9.41E-02±1.20E-04+	9.41E-02±1.38E-04+	9.45E-02±1.13E-04+	9.47E-02±2.22E-05+	9.15E-02±1.24E-05−	9.15E-02±2.49E-07
DTLZ6	9.51E-02±1.49E-02+	9.46E-02±1.51E-04+	9.56E-02±3.57E-05+	9.56E-02±2.03E-05+	9.24E-02±2.63E-06+	9.23E-02±1.23E-07
DTLZ7	2.98E-01±2.71E-03+	3.04E-01±2.91E-03+	3.07E-01±1.67E-03+	3.12E-01±1.29E-03+	2.17E-01±2.27E-03≈	2.17E-01±2.75E-03
UF1	5.71E-01±1.59E-02−	6.29E-01±1.39E-02−	5.44E-01±2.63E-02−	5.73E-01±3.61E-02−	6.57E-01±2.26E-03≈	6.57E-01±1.34E-03
UF2	6.30E-01±7.40E-03−	6.48E-01±2.05E-03−	6.33E-01±8.49E-03−	6.42E-01±2.93E-03−	6.47E-01±9.69E-03−	6.56E-01±1.15E-03
UF3	4.67E-01±4.35E-02−	6.36E-01±1.18E-02−	4.37E-01±5.63E-02−	4.43E-01±2.93E-02−	6.37E-01±2.72E-02−	6.50E-01±8.98E-03
UF4	2.64E-01±1.17E-03−	2.79E-01±7.45E-03≈	2.71E-01±7.26E-04−	2.80E-01±5.16E-04+	2.27E-01±6.69E-03−	2.78E-01±9.98E-04
UF5	1.65E-01±1.31E-01≈	3.27E-01±4.94E-02+	1.87E-01±7.82E-02≈	1.29E-01±2.03E-01−	9.18E-02±1.52E-01−	2.00E-01±1.12E-01
UF6	2.32E-01±5.92E-02≈	2.33E-01±8.69E-02≈	2.44E-01±1.08E-01≈	2.25E-01±1.05E-01−	1.99E-01±1.35E-01−	2.51E-01±5.94E-02
UF7	4.43E-01±1.27E-01−	4.76E-01±3.68E-03−	4.36E-01±1.51E-01−	4.49E-01±1.26E-02−	4.88E-01±3.94E-03≈	4.88E-01±2.45E-03
UF8	2.05E-01±1.14E-01−	1.35E-01±1.57E-01−	1.56E-01±8.86E-03−	2.59E-01±9.33E-02−	2.84E-01±5.70E-03−	3.02E-01±3.12E-03
UF9	3.82E-01±1.11E-01−	3.58E-01±1.74E-01−	5.44E-01±9.88E-02−	5.98E-01±1.36E-02≈	5.32E-01±1.07E-01−	5.46E-01±1.07E-01
UF10	2.21E-02±4.52E-02−	3.76E-02±8.65E-02−	4.56E-02±3.72E-02−	1.14E-01±2.21E-02≈	4.08E-02±4.72E-02−	1.32E-01±7.59E-02
UF11	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	3.25E-05±9.28E-05−	3.08E-04±2.86E-04
UF12	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00
UF13	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00
CF1	4.66E-01±1.71E-03+	4.71E-01±8.97E-04+	4.62E-01±2.69E-03+	4.42E-01±5.23E-03−	4.39E-01±1.08E-05≈	4.59E-01±2.00E-02
CF2	5.88E-01±3.55E-02−	6.09E-01±1.87E-03−	5.72E-01±2.74E-02−	5.61E-01±3.05E-02−	6.50E-01±1.28E-03−	6.51E-01±1.19E-03
CF3	7.56E-02±6.04E-02−	1.46E-01±9.96E-02≈	1.03E-01±5.43E-02≈	0.00E-00±5.87E-02−	1.14E-01±3.21E-02≈	1.18E-01±5.46E-02
CF4	1.59E-01±9.84E-02−	2.02E-01±5.38E-02−	2.94E-02±1.70E-01−	7.55E-02±1.18E-01−	5.44E-01±7.82E-03−	5.49E-01±4.00E-03
CF5	0.00E-00±9.62E-02−	1.53E-01±2.18E-01−	0.00E-00±4.88E-02−	0.00E-00±0.00E-00−	3.17E-01±7.34E-02≈	3.38E-01±2.00E-01
CF6	3.38E-01±1.48E-01−	4.18E-01±1.28E-01−	2.36E-01±2.50E-01−	1.07E-01±2.25E-01−	6.44E-01±1.52E-02−	6.58E-01±1.96E-03
CF7	1.79E-01±2.38E-01−	2.66E-01±1.79E-01−	0.00E-00±1.65E-01−	0.00E-00±0.00E-00−	4.31E-01±7.78E-02−	4.69E-01±1.94E-01
CF8	0.00E-00±9.89E-02−	1.62E-01±5.95E-02−	2.17E-01±1.97E-01≈	2.00E-01±2.43E-01≈	2.24E-01±9.42E-02≈	2.04E-01±5.07E-02
CF9	2.03E-01±8.58E-02−	2.67E-01±2.05E-02−	2.85E-01±2.06E-02−	3.04E-01±1.96E-02−	3.31E-01±1.50E-02+	3.25E-01±2.91E-02
CF10	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	1.85E-01±3.78E-02−	2.07E-01±6.46E-02

+ , ≈ and − represent previous algorithm statistically significant better, similar and worse than the last algorithm, respectively

Table 4: Hypervolume Median and IQR of CMA-ES, SaDE, CoDE, MOEA/D-T3, MOEA/D-T', MOEA/D-T

MOPs	CMA-ES	SaDE	CoDE	MOEA/D-T3	MOEA/D-T'	MOEA/D-T
ZDT1	6.16E-01±1.16E-02−	6.62E-01±9.38E-06−	6.62E-01±1.22E-05−	6.62E-01±2.15E-06≈	6.62E-01±1.13E-08+	6.62E-01±4.51E-07
ZDT2	2.99E-01±2.26E-02−	3.28E-01±3.18E-06−	3.28E-01±3.28E-01−	3.28E-01±7.88E-09+	3.28E-01±3.05E-10+	3.28E-01±1.21E-08
ZDT3	3.90E-01±5.47E-02−	5.14E-01±1.23E-05−	5.14E-01±1.77E-05−	5.14E-01±2.73E-06≈	5.14E-01±4.86E-07+	5.14E-01±8.61E-06
ZDT4	0.00E-00±0.00E-00−	2.10E-01±2.08E-01−	0.00E-00±0.00E-00−	1.08E-01±2.10E-01−	2.75E-01±2.33E-01−	6.62E-01±1.05E-04
ZDT6	2.57E-01±3.83E-02−	4.01E-01±5.60E-07−	4.01E-01±8.23E-08−	4.01E-01±2.24E-08≈	4.01E-01±1.58E-09+	4.01E-01±6.21E-09
DTLZ1	0.00E-00±0.00E-00−	7.61E-01±1.63E-04+	1.06E-01±1.06E-01−	7.39E-01±6.55E-01−	7.61E-01±5.49E-05≈	7.61E-01±1.22E-04
DTLZ2	3.90E-01±4.01E-03−	3.93E-01±3.00E-04+	3.93E-01±2.77E-04+	3.93E-01±2.32E-04≈	3.93E-01±3.07E-04≈	3.93E-01±2.67E-04
DTLZ3	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	0.00E-00±0.00E-00−	3.93E-01±3.15E-04
DTLZ4	1.94E-01±3.06E-02−	3.89E-01±6.15E-04+	3.89E-01±5.20E-04+	3.88E-01±4.63E-04≈	3.88E-01±4.29E-04≈	3.88E-01±2.04E-04
DTLZ5	9.11E-02±5.47E-05−	9.15E-02±5.07E-07+	9.15E-02±6.82E-07≈	9.15E-02±1.50E-07≈	9.15E-02±2.50E-08≈	9.15E-02±2.49E-07
DTLZ6	0.00E-00±0.00E-00−	9.23E-02±6.59E-07≈	9.23E-02±8.99E-07≈	9.23E-02±5.58E-08≈	9.23E-02±1.18E-10≈	9.23E-02±1.23E-07
DTLZ7	2.28E-01±2.54E-02+	2.15E-01±2.34E-03≈	2.16E-01±2.55E-03≈	2.17E-01±2.65E-03≈	2.17E-01±2.61E-03≈	2.17E-01±2.75E-03
UF1	5.22E-01±2.23E-02−	6.40E-01±4.48E-03−	6.55E-01±2.12E-03−	6.55E-01±2.61E-03−	6.53E-01±6.08E-03−	6.57E-01±1.34E-03
UF2	6.24E-01±7.82E-03−	6.49E-01±5.45E-03−	6.53E-01±2.91E-03−	6.52E-01±4.31E-03−	6.49E-01±4.57E-03−	6.56E-01±1.15E-03
UF3	4.52E-01±2.42E-02−	5.16E-01±1.03E-01−	6.34E-01±2.19E-02−	6.46E-01±1.60E-02≈	6.19E-01±3.39E-02−	6.50E-01±8.98E-03
UF4	2.05E-01±3.66E-03−	2.81E-01±9.38E-04+	2.77E-01±6.80E-04−	2.79E-01±1.03E-03+	2.80E-01±1.99E-03+	2.78E-01±9.98E-04
UF5	0.00E-00±0.00E-00−	1.87E-01±1.07E-01≈	5.93E-02±1.27E-01−	1.72E-01±1.20E-01≈	2.34E-01±1.04E-01≈	2.00E-01±1.12E-01
UF6	7.49E-03±1.84E-02−	2.28E-01±5.13E-02−	2.14E-01±1.31E-01−	2.34E-01±4.18E-02≈	2.33E-01±6.99E-02−	2.51E-01±5.94E-02
UF7	1.84E-01±1.19E-01−	4.76E-01±5.49E-03−	4.86E-01±3.26E-03−	4.86E-01±2.64E-03−	4.84E-01±7.08E-03−	4.88E-01±2.45E-03
UF8	1.99E-01±1.63E-02−	2.91E-01±6.63E-03−	3.02E-01±3.91E-03≈	3.01E-01±5.89E-03≈	2.96E-01±8.34E-03−	3.02E-01±3.12E-03
UF9	4.69E-01±3.80E-02−	6.20E-01±1.13E-01≈	5.46E-01±1.10E-01≈	5.45E-01±1.07E-01≈	5.46E-01±1.06E-01≈	5.46E-01±1.07E-01
UF10	0.00E-00±0.00E-00−	1.86E-01±4.19E-02+	8.66E-02±7.76E-02−	1.14E-01±8.39E-02≈	6.25E-02±1.51E-01≈	1.32E-01±7.59E-02
UF11	3.18E-04±2.87E-04≈	1.80E-04±1.91E-04−	3.39E-04±2.15E-04≈	3.34E-04±2.88E-04≈	3.08E-04±2.86E-04≈	3.08E-04±2.86E-04
UF12	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00
UF13	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00≈	0.00E-00±0.00E-00
CF1	4.41E-01±2.00E-02−	4.39E-01±2.00E-02−	4.39E-01±2.00E-02≈	4.39E-01±2.00E-02≈	4.39E-01±2.00E-02−	4.59E-01±2.00E-02
CF2	5.72E-01±4.76E-02−	6.50E-01±3.79E-03−	6.50E-01±9.01E-04−	6.50E-01±1.52E-03−	6.50E-01±1.47E-03−	6.51E-01±1.19E-03
CF3	3.13E-02±4.62E-02−	1.44E-01±4.68E-02≈	1.32E-01±1.23E-01≈	1.45E-01±6.43E-02≈	1.33E-01±1.35E-01≈	1.18E-01±5.46E-02
CF4	3.67E-01±8.93E-02−	5.19E-01±1.07E-02−	5.44E-01±5.55E-03−	5.44E-01±4.30E-03−	5.40E-01±7.07E-03−	5.49E-01±4.00E-03
CF5	0.00E-00±0.00E-00−	2.91E-01±1.69E-01−	3.30E-01±1.68E-01≈	4.23E-01±2.65E-01≈	2.92E-01±2.04E-01≈	3.38E-01±2.00E-01
CF6	6.41E-01±1.18E-02−	6.52E-01±7.46E-03−	6.56E-01±3.13E-03−	6.56E-01±2.53E-03−	6.45E-01±3.12E-02−	6.58E-01±1.96E-03
CF7	0.00E-00±9.56E-03−	5.02E-01±4.71E-02≈	4.26E-01±2.16E-01−	5.39E-01±2.21E-01≈	4.07E-01±2.31E-01−	4.69E-01±1.94E-01
CF8	2.02E-01±4.72E-02≈	1.63E-01±8.09E-02−	1.71E-01±6.53E-02−	2.17E-01±5.64E-02≈	2.05E-01±6.89E-02≈	2.04E-01±5.07E-02
CF9	3.02E-01±2.67E-02−	2.78E-01±9.24E-03−	3.07E-01±4.08E-02−	3.15E-01±3.55E-02≈	2.89E-01±2.80E-02−	3.25E-01±2.91E-02
CF10	0.00E-00±0.00E-00−	2.02E-01±2.61E-03−	2.03E-01±2.37E-03−	2.07E-01±4.39E-04≈	2.07E-01±3.45E-04≈	2.07E-01±6.46E-02

+ , ≈ and − represent previous algorithm statistically significant better, similar and worse than the last algorithm, respectively