Performance Assessment of Generalized Differential Evolution 3 with a Given Set of Constrained Multi-Objective Test Problems

Saku Kukkonen, *Student Member, IEEE* and Jouni Lampinen

***Abstract*— This paper presents results for the CEC 2009 Special Session on "Performance Assessment of Constrained / Bound Constrained Multi-Objective Optimization Algorithms" when Generalized Differential Evolution 3 has been used to solve a given set of test problems. The set consist of 23 problems having two, three, or five objectives. Problems have different properties in the sense of separability, modality, and geometry of the Pareto-front. The most of the problems are unconstrained, but 10 problems have one or two constraints.**

According to the numerical results with an inverted generational distance, Generalized Differential Evolution 3 performed well with all the problems except with one five objective problem. It was noticed that a low crossover control parameter value provides the best average results according to the metric.

I. Introduction

In this paper, a general purpose Evolutionary Algorithm (EA) called Generalized Differential Evolution 3 (GDE3) [1] with a diversity maintenance technique suited for many-objective problems [2] has been used to solve multi-objective problems defined for the CEC 2009 Special Session on "Performance Assessment of Constrained / Bound Constrained Multi-Objective Optimization Algorithms". The problems have been defined in [3], where also evaluation criteria are given. The problems have two, three, or five objectives, and the number of decision variables varies from 10 to 30. Difficulty of functions vary and they had different geometry of Pareto-optimal solutions in the objective and decision variable spaces. All the problems contain boundary constraints for decision variables and ten of problems contain also one or two inequality constraints.

This paper continues with the following parts: Multi-objective optimization with constraints is briefly defined in Section II. Section III describes the multi-objective optimization method used to solve the given set of problems. Section IV describes experiments and results. Finally, conclusions are given in Section V.

II. Multi-Objective Optimization with Constraints

A multi-objective optimization problem (MOOP) with constraints can be presented in the form [4, p. 37]:

$$\begin{array}{ll} \text{minimize} & \{f_1(\vec{x}), f_2(\vec{x}), \ldots, f_M(\vec{x})\} \\ \text{subject to} & (g_1(\vec{x}), g_2(\vec{x}), \ldots, g_K(\vec{x}))^T \leq \vec{0}. \end{array}$$

S. Kukkonen is with the Department of Mathematics and Physics, Lappeenranta University of Technology, P.O. Box 20, FIN-53851 Lappeenranta, Finland; email: saku.kukkonen@lut.fi. J. Lampinen is with the Department of Computer Science, University of Vaasa, P.O. Box 700, FI-65101 Vaasa, Finland; email: jouni.lampinen@uwasa.fi.

Thus, there are M functions to be optimized and K inequality constraints.

The goal of *a posteriori* multi-objective optimization is to find an approximation of a Pareto-front, *i.e.*, to find a set of solutions that are not dominated by any other solution. Weak dominance relation $\preceq$ between two vectors is defined such a way that $\vec{x}_1$ weakly dominates $\vec{x}_2$, *i.e.*, $\vec{x}_1 \preceq \vec{x}_2$ iff $\forall i : f_i(\vec{x}_1) \leq f_i(\vec{x}_2)$. Dominance relation $\prec$ between two vectors is defined such a way that $\vec{x}_1$ dominates $\vec{x}_2$, *i.e.*, $\vec{x}_1 \prec \vec{x}_2$ iff $\vec{x}_1 \preceq \vec{x}_2 \ \wedge \ \exists i : f_i(\vec{x}_1) < f_i(\vec{x}_2)$. The dominance relationship can be extended to take into consideration constraint values and objective values at the same time. A constraint-domination $\prec_c$ is defined here in such a way that $\vec{x}_1$ constraint-dominates $\vec{x}_2$, *i.e.*, $\vec{x}_1 \prec_c \vec{x}_2$ iff any of the following conditions is true [5]:

- $\vec{x}_1$ and $\vec{x}_2$ are infeasible and $\vec{x}_1$ dominates $\vec{x}_2$ in constraint function violation space[1].
- $\vec{x}_1$ is feasible and $\vec{x}_2$ is not.
- $\vec{x}_1$ and $\vec{x}_2$ are feasible and $\vec{x}_1$ dominates $\vec{x}_2$ in objective function space.

The definition for weak constraint-domination $\preceq_c$ is analogous dominance relation changed to weak dominance in the definition above. This constraint-domination is a special case of more general concept of having goals and priorities that is presented in [6].

III. Optimization Method

A. Differential Evolution

The Differential Evolution (DE) algorithm [7], [8] was introduced by Storn and Price in 1995. The design principles of DE are simplicity, efficiency, and the use of floating-point encoding instead of binary numbers. As a typical EA, DE has a random initial population that is then improved using selection, mutation, and crossover operations. Several ways exist to determine a stopping criterion for EAs but usually a predefined upper limit (G_{max}) for the number of generations to be computed provides an appropriate stopping condition. Other control parameters for DE are the crossover control parameter (CR), the mutation factor (F), and the population size (NP).

In each generation G, DE goes through each D dimensional decision vector $\vec{x}_{i,G}$ of the population and creates the

[1]The constraint function violations are calculated as $\max\left(g_k(\vec{x}), 0\right), k = 1, \ldots, K$.

corresponding trial vector $\vec{u}_{i,G}$ as follows [9]:

$$
\begin{array}{l}
r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}, \text{(randomly selected,} \\
\quad \text{except mutually different and different from } i) \\
j_{rand} = \text{floor}\left(rand_i[0,1) \cdot D\right) + 1 \\
\text{for}(j = 1; j \leq D; j = j + 1) \\
\{ \\
\quad \text{if}(rand_j[0,1) < CR \vee j = j_{rand}) \\
\quad\quad u_{j,i,G} = x_{j,r_3,G} + F \cdot (x_{j,r_1,G} - x_{j,r_2,G}) \\
\quad \text{else} \\
\quad\quad u_{j,i,G} = x_{j,i,G} \\
\}
\end{array}
$$

This is the most common DE version, DE/rand/1/bin. Both CR and F remain fixed during the entire execution of the algorithm. Parameter $CR \in [0, 1]$, which controls the crossover operation, represents the probability that an element for the trial vector is chosen from a linear combination of three randomly chosen vectors and not from the old vector $\vec{x}_{i,G}$. The condition "$j = j_{rand}$" ensures that at least one element of the trial vector is different compared to the elements of the old vector. Parameter F is a scaling factor for mutation and its value range is $(0, 1+]$[2]. In practice, CR controls rotational invariance of the search, and its small value (*e.g.*, 0.1) is practicable with separable problems while larger values (*e.g.*, 0.9) are for non-separable problems. Parameter F controls the speed and robustness of the search, *i.e.*, a lower value for F increases the convergence rate but it also increases the risk of getting stuck into a local optimum. Parameters CR and NP have the similar effect on the convergence rate as F has.

After the mutation and crossover operations, the trial vector $\vec{u}_{i,G}$ is compared to the old vector $\vec{x}_{i,G}$. If the trial vector has an equal or better objective value, then it replaces the old vector in the next generation. This can be presented as follows in the case of minimization of an objective [9]:

$$
\vec{x}_{i,G+1} = \begin{cases} \vec{u}_{i,G} & \text{if} \quad f(\vec{u}_{i,G}) \leq f(\vec{x}_{i,G}) \\ \vec{x}_{i,G} & \text{otherwise} \end{cases} .
$$

DE is an elitist method since the best population member is always preserved and the average objective value of the population will never deteriorate.

B. Generalized Differential Evolution

The first version of Generalized Differential Evolution (GDE) extended DE for constrained multi-objective optimization, and it modified only the selection rule of the basic DE [5]. The basic idea in the selection rule of GDE is that the trial vector is selected to replace the old vector in the next generation if it weakly constraint-dominates (*cf.* Section II) the old vector. There was no explicit sorting of non-dominated solutions [10, pp. 33–44] during the optimization process or any mechanism for maintaining the distribution and extent of solutions. Also, there was no extra repository for non-dominated solutions.

[2]Notation means that F is larger than 0 and upper limit is in practice around 1 although there is no hard upper limit.

The second version, GDE2, made the selection based on crowdedness when the trial and old vector were feasible and non-dominating each other in the objective function space [11]. This improves the extent and distribution of the obtained set of solutions but slows down the convergence of the overall population because it favors isolated solutions far from the Pareto-front until all the solutions are converged near to the Pareto-front.

The third and latest version is GDE3 [1], [12]. Besides the selection, another part of the basic DE has also been modified. Now, in the case of feasible and non-dominating solutions, both vectors are saved for the population of next generation. Before continuing to the next generation, the size of the population is reduced using non-dominated sorting and pruning based on diversity preservation [12]. The GDE3 algorithm is presented in Figure 1. Parts that are new compared to previous GDE versions are framed in Figure 1. Without these parts, the algorithm is identical to the first version of GDE.

The pruning technique used in the original GDE3 is based on crowding distance [10, pp. 248–249], which provides a good crowding estimation in the case of two objectives. However, crowding distance fails to approximate crowdedness of solutions when the number of objectives is more than two [12]. Since, the provided problem set in [3] consists of problems with more than two objectives, a more general diversity maintenance technique proposed in [2] is used. The technique is based on a crowding estimation using the nearest neighbors of solutions in Euclidean sense, and an efficient nearest neighbors search technique.

All the GDE versions handle any number of M objectives and any number of K constraints, including the cases where $M = 0$ (constraint satisfaction problem) and $K = 0$ (unconstrained problem). When $M = 1$ and $K = 0$, the versions are identical to the original (unconstrained single-objective) DE, and this is why they are referred as *Generalized* DEs.

GDE can be implemented in such a way that the number of function evaluations is reduced because not always all the constraints and objectives need to be evaluated, *e.g.*, inspecting a few (even one) constraint violations is often enough to determine, which vector to select for the next generation [8], [13]–[16]. This can give truly remarkable computational effort reductions compared to constraint handling approaches that evaluate all the constraints when a solution candidate is evaluated. This has importance in practice when functions are often computationally expensive to evaluate and reductions in the number of function evaluations directly affect into overall time needed for the search.

One should note that this optimization method used is totally evolutionary thus no hybridization with, *e.g.*, some local search method is used.

IV. Experiments

A. Configuration

GDE3 and the given problems were implemented in the ANSI-C programming language and compiled with the GCC

Input : $D, G_{max}, NP \geq 4, F \in (0, 1+], CR \in [0, 1]$, and initial bounds: $\vec{x}^{(lo)}, \vec{x}^{(hi)}$

$$\text{Initialize} : \begin{cases} \forall i \leq NP \wedge \forall j \leq D : x_{j,i,G=0} = x_j^{(lo)} + rand_j[0,1] \cdot \left(x_j^{(hi)} - x_j^{(lo)}\right) \\ i = \{1, 2, \ldots, NP\}, j = \{1, 2, \ldots, D\}, G = 0, n = 0, rand_j[0,1] \in [0,1], \end{cases}$$

While $G < G_{max}$

$\forall i \leq NP$:

Mutate and recombine:

$r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$, randomly selected, except mutually different and different from i

$j_{rand} \in \{1, 2, \ldots, D\}$, randomly selected for each i

$$\forall j \leq D, u_{j,i,G} = \begin{cases} x_{j,r_3,G} + F \cdot (x_{j,r_1,G} - x_{j,r_2,G}) \\ \qquad \text{if} \quad rand_j[0,1) < CR \vee j == j_{rand} \\ x_{j,i,G} \quad \text{otherwise} \end{cases}$$

Select :

$$\vec{x}_{i,G+1} = \begin{cases} \vec{u}_{i,G} & \text{if} \quad \vec{u}_{i,G} \preceq_c \vec{x}_{i,G} \\ \vec{x}_{i,G} & \text{otherwise} \end{cases}$$

Set :

$$\begin{matrix} n = n + 1 \\ \vec{x}_{NP+n,G+1} = \vec{u}_{i,G} \end{matrix} \quad \text{if} \begin{cases} \forall k : g_k(\vec{u}_{i,G}) \leq 0 \\ \wedge \\ \vec{x}_{i,G+1} == \vec{x}_{i,G} \\ \wedge \\ \neg(\vec{x}_{i,G} \prec \vec{u}_{i,G}) \end{cases}$$

Select $\vec{x} \in \mathcal{P} = \{\vec{x}_{1,G+1}, \vec{x}_{2,G+1}, \ldots, \vec{x}_{NP+n,G+1}\}$:

$$\begin{cases} \vec{x} \text{ belongs to the last non-dominated set of } \mathcal{P} \\ \wedge \\ \vec{x} \text{ is the most crowded in the last non-dominated set} \end{cases}$$

Remove $\vec{x}$

$n = n - 1$

$G = G + 1$

Fig. 1. Generalized Differential Evolution 3

compiler. The hardware was an ordinary PC with 2.8 GHz CPU & 1 GB RAM, and the operating system was Debian linux 4.0.

In the case of boundary constraint violations, violating variable values were "reflected" back from the violated boundary using following rule before the selection operation of GDE3:

$$u_{j,i,G} = \begin{cases} 2x_j^{(lo)} - u_{j,i,G} & \text{if} \quad u_{j,i,G} < x_j^{(lo)} \\ 2x_j^{(up)} - u_{j,i,G} & \text{if} \quad u_{j,i,G} > x_j^{(up)} \\ u_{j,i,G} & \text{otherwise} \end{cases},$$

where $x_j^{(lo)}$ and $x_j^{(up)}$ are lower and upper limits respectively for a decision variable x_j.

B. Parameter Setting

Along stopping criterion and size of the population (NP), GDE3 has two control parameters (CR and F) as described in Section III-A, where the effect and ranges of these are also given. As a rule of thumb in single-objective optimization, if nothing is known about the problem in hand, then suitable initial control parameter values are $CR = 0.9$ and $F = 0.9$, and $NP = 5 \cdot D \ldots 30 \cdot D$, where D is the number of decision variables of the problem [17]. For an easy problem (*e.g.*, unimodal and low dimensional), a small value of NP is sufficient but with difficult problems, a large value of NP is recommended in order to avoid stagnation to a local optimum. In general, increase of control parameter values, will also increase the number of function evaluations (FES) needed. Dependency between NP and FES needed is linear while FES needed increases more rapidly along CR and F [16]. If values of F and/or NP are too low, search is prone to stagnate to a local optimum (with very low control parameter values the search converges without the selection pressure) [18].

In the case of multi-objective optimization and conflicting objectives, often lower control parameter values (*e.g.*, 0.2) for CR and F can be used than in single-objective optimization because conflicting objectives already maintain diversity and restrain the search speed. This has been noted in [19], [20], where the effect of the control parameters has been studied empirically. Also, if the complexity of objectives differ (*e.g.*, as in the case of the ZDT problems [10, pp. 356–360]), then a high value for CR might lead to premature convergence with respect to one objective compared to another. However, if problems are more difficult containing linkage between decision variables and objectives, then higher value for CR might be better [21]. The value of NP can be selected in same way as in single-objective optimization or it can be selected according to a desired approximation set size of the Pareto-optimal front.

To keep the setup as simple as possible, the same set of control parameter values were used for all the problems. It would had been also possible to apply some kind of dependency on the number of objectives and/or the number of decision variables. As well, it would had been possible to use some control parameter adaptation strategy as in [22]–[26]. However, these were not applied, because then it would had been unclear, how parameter adjustment *vs.* the optimization algorithm itself contributes to the results.

The control parameter values used were $CR = 0.0$, $F = 0.5$, $NP = 200$, and $G_{max} = 1499$. The first two control parameter values were almost the same as used in [27]. These values were obtained experimentally by trying through a combination of different control parameter values. Outcome of different control parameter values has been reported in the following section.

The size of the population was set according to preliminary tests with the problems ($NP = 200$ provided better results with some problems than $NP = 100$ or $NP = 300$). This population size was also suitable in respect to given approximation set sizes. With chosen NP and G_{max} values, the number of solution candidate evaluations is exactly 300 000, which was the upper limit given in [3].

In [3], approximation sets of different sizes were demanded for different problems. In GDE3, the size of the approximation set is usually same as NP. Now, NP was kept fixed for all the problems, and solutions for the approximation set were collected during generations. Populations of 500 last generations were merged together and non-dominated solutions were selected from this merged set of solutions. If the size of non-dominated set was larger than the desired approximation set size, then the set was reduced to desired size using the pruning technique described in [2].

C. Results of Experiments

The problem set given in [3] contain problems with boundary constraints (problems UF1 – UF10, R2_DTLZ2_M5, R2_DTLZ3_M5, and WFG1_5M) and problems with one (CF1 – CF5 & CF8 – CF10) or two (CF6 & CF7) inequality constraints in addition to boundary constraints[3]. The most of the problems contain two objectives and the maximal number of solutions allowed in the approximation set is 100 [3]. Problems UF8 – UF10 and CF8 – CF10 contain three objectives and the maximal number of solutions allowed in the approximation set is 150. Three problems R2_DTLZ2_M5, R2_DTLZ3_M5, and WFG1_5M contain five objectives and the maximal number of solutions allowed in the approximation set is 800 for these.

The problems were solved 30 times and achieved results are presented in Table I, which shows the inverted generational distance (IGD)[4] values and CPU times for different problems. The final approximation set for the UF and CF problems with the smallest IGD value are shown in Figures 2 and 3.

According to the IGD indicator values in Table I, the R2_DTLZ3_M5 problem was the hardest to solve. Relatively high IGD values were also obtained for UF10, CF10, and WFG1_5M. For the rest of the problems IGD values were closer to 0. It appeared that more objectives cause more problems to converge close to the Pareto-front that is nowadays already self-evident observation.

CPU times for the optimization method were around five seconds for the two and three objective problems. For the five objective problems CPU times were several times larger. GDE3 is well scalable algorithm with the simple genetic operations of DE. Therefore, also problems with a large number of decision variables and/or a large population size as in [28] are solvable in reasonable time. The most complex operation in GDE3 is non-dominated sorting, which time complexity is $O\left(N \log^{M-1} N\right)$ [29][5].

The final approximation sets in Figures 2 and 3 are good for the most of the problems when they are compared with sets in [3]. Notable worse approximation sets are obtained for UF6, UF8, and CF3 while the reason for this is unknown. When control parameter values $CR = 0.99$, $F = 0.5$ were used instead, better looking approximation sets were obtained for these problems.

Approximation sets for CF8, CF9, and CF10 differ from the approximation sets given in [3]. When data files were examined, it was noticed that the inequality constraint is not

[3]Inequality constraints were considered exactly the same ways as they were defined without allowable -10^{-6} error tolerance presented in [3]

[4]If P^* is a set of uniformly distributed solutions on the Pareto-front and Q is an approximation set, then corresponding IGD value is defined as [3]:

$$IGD(Q, P^*) = \frac{\sum_{v \in P^*} d(v, Q)}{|P^*|}, \quad (1)$$

where $d(v, Q)$ is minimum Euclidean distance between v and points in Q. The value of IGD is always non-negative and a lower IGD value is better.

[5]Actual non-dominated sorting implementation in GDE3 is naive that increased the computation time.

TABLE I
IGD METRIC VALUES AND CPU TIMES FOR THE PROBLEMS (STD MEANS STANDARD DEVIATION)

	IGD				CPU times (s)	
	Mean	Std	Best	Worst	Mean	Std
UF1	0.005342	0.000342	0.004815	0.006242	3.948333	0.026272
UF2	0.011953	0.001541	0.009020	0.014972	6.440333	0.022358
UF3	0.106395	0.012900	0.085759	0.138381	8.136333	0.078630
UF4	0.026506	0.000372	0.025857	0.027550	6.222333	0.028969
UF5	0.039281	0.003947	0.031791	0.045880	1.989000	0.020736
UF6	0.250913	0.019573	0.204163	0.282468	4.993333	0.030210
UF7	0.025228	0.008891	0.014125	0.042002	3.971667	0.067112
UF8	0.248556	0.035521	0.194990	0.365385	7.136000	0.028599
UF9	0.082482	0.022485	0.045261	0.133812	6.674333	0.040995
UF10	0.433261	0.012323	0.393773	0.445574	8.553667	0.047596
R2_DTLZ2_M5	0.234256	0.010040	0.218349	0.255739	44.014000	0.989216
R2_DTLZ3_M5	202.125819	37.705584	138.164747	278.261876	43.293000	1.355246
WFG1_M5	3.205734	0.082125	2.996780	3.306210	28.735667	0.222659
CF1	0.029402	0.002292	0.024174	0.034840	4.661000	0.060876
CF2	0.015976	0.007558	0.007590	0.044158	4.127667	0.054437
CF3	0.127506	0.023861	0.074643	0.173374	3.043667	0.057804
CF4	0.007991	0.001227	0.006551	0.012555	3.431667	0.045568
CF5	0.067998	0.013526	0.056052	0.117258	2.473000	0.116209
CF6	0.061998	0.026863	0.029581	0.137857	6.154000	0.046727
CF7	0.041691	0.010768	0.028622	0.075913	2.825333	0.055195
CF8	0.138722	0.058564	0.072034	0.322660	6.783333	0.299670
CF9	0.114504	0.022108	0.081175	0.157444	6.744333	0.066472
CF10	0.492331	0.001676	0.487207	0.494315	7.912333	0.190076

active and there is an error in the given problem descriptions in [3].

With each problem and repetition, maximal number of solutions allowed in the approximation set was obtained in the search. It was observed that IGD values decreased almost monotonically with each problem. The convergence graph for the hardest problem, R2_DTLZ3_M5, is shown in Figure 4. The search is advancing also with this problem but IGD values are in totally different scale compared to the other problems, *e.g.*, WFG1_M5 in Figure 5.

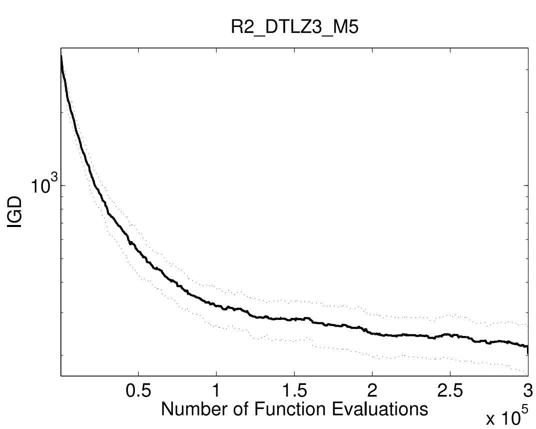

Fig. 4. Mean convergence curve with standard deviations for R2_DTLZ3_M5 (note logarithmic scale of the IGD values)

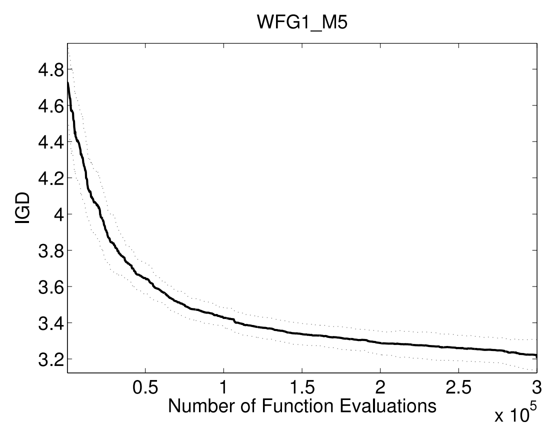

Fig. 5. Mean convergence curve with standard deviations for WFG1_M5

As it is noted in Section III-B, the constraint handling approach used in GDE3 reduces the number of function evaluations. Table III contains the actual number of function evaluations for CF1 – CF10, which have one or two inequality constraints. With a larger number of constraints the reduction in the number of function evaluations can be truly remarkable as note in [16].

Table II shows mean IGD values over all the problems and all the repetitions for different CR and F values that were tried. It can be noticed that according the mean IGD values, $CR = 0.0$ and $F = 0.5$ provide clearly the best results

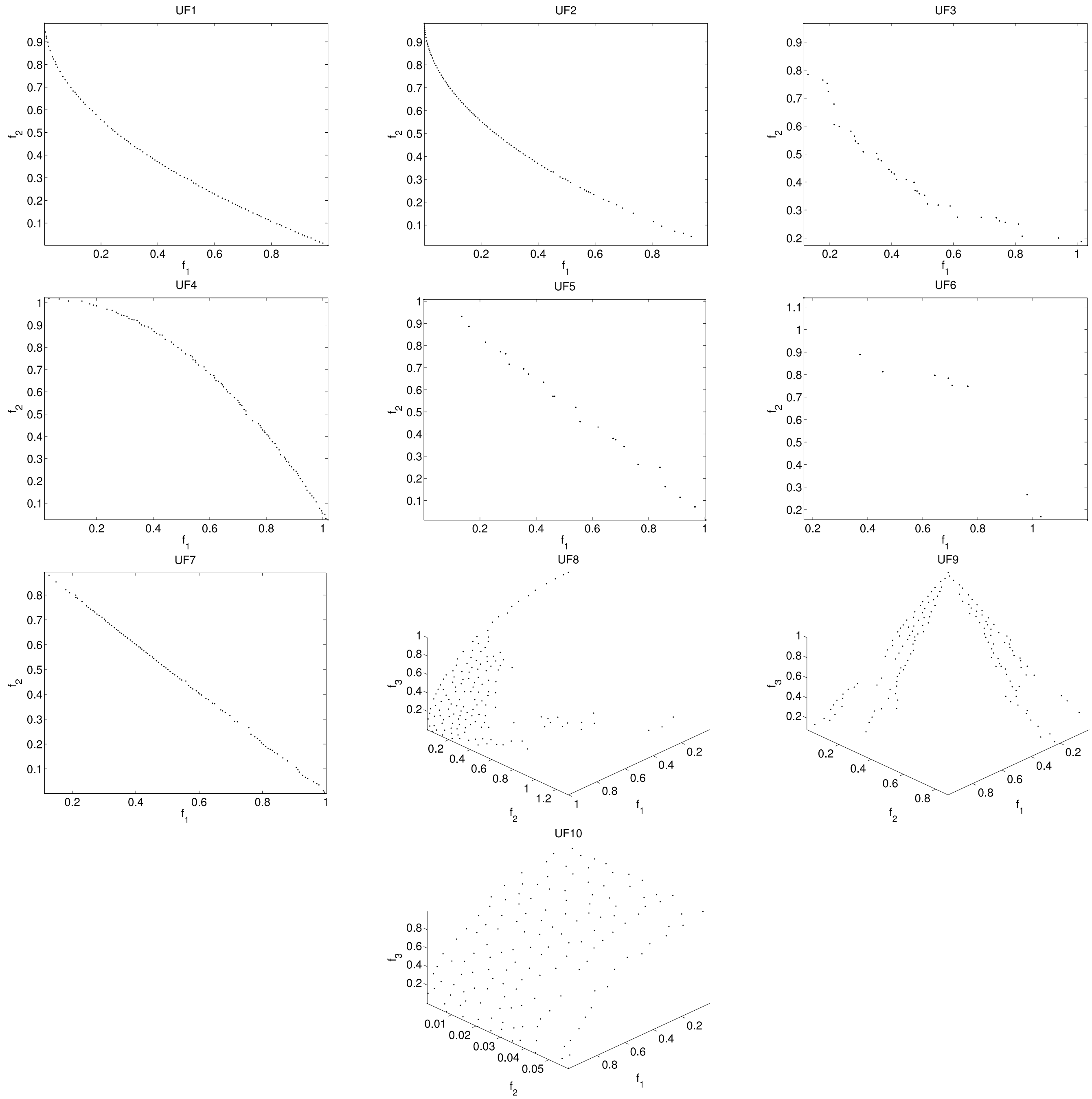

Fig. 2. The final approximation sets for the UF problems

for these problems. This is interesting observation since the problems are mainly non-separable and a higher *CR* value is usually recommended in such cases with the single-objective DE.

V. CONCLUSIONS

Results of Generalized Differential Evolution 3 for the CEC 2009 Special Session on "Performance Assessment of Constrained / Bound Constrained Multi-Objective Optimization Algorithms" have been reported. The problems given were solved with the same fixed control parameter settings, *i.e.*, there was no parameter adaption based on problem characteristic or other criteria. Also, the optimization method was a pure multi-objective evolutionary algorithm without hybridization with other techniques.

According to the numerical results, Generalized Differential Evolution 3 performed well with all the problems except with one five objective problem, R2_DTLZ3_M5. Natural deterioration of the performance was observed with the increased number of objectives.

Interestingly, it was noted that value 0 for the crossover control parameter *CR* gave the best numerical results al-

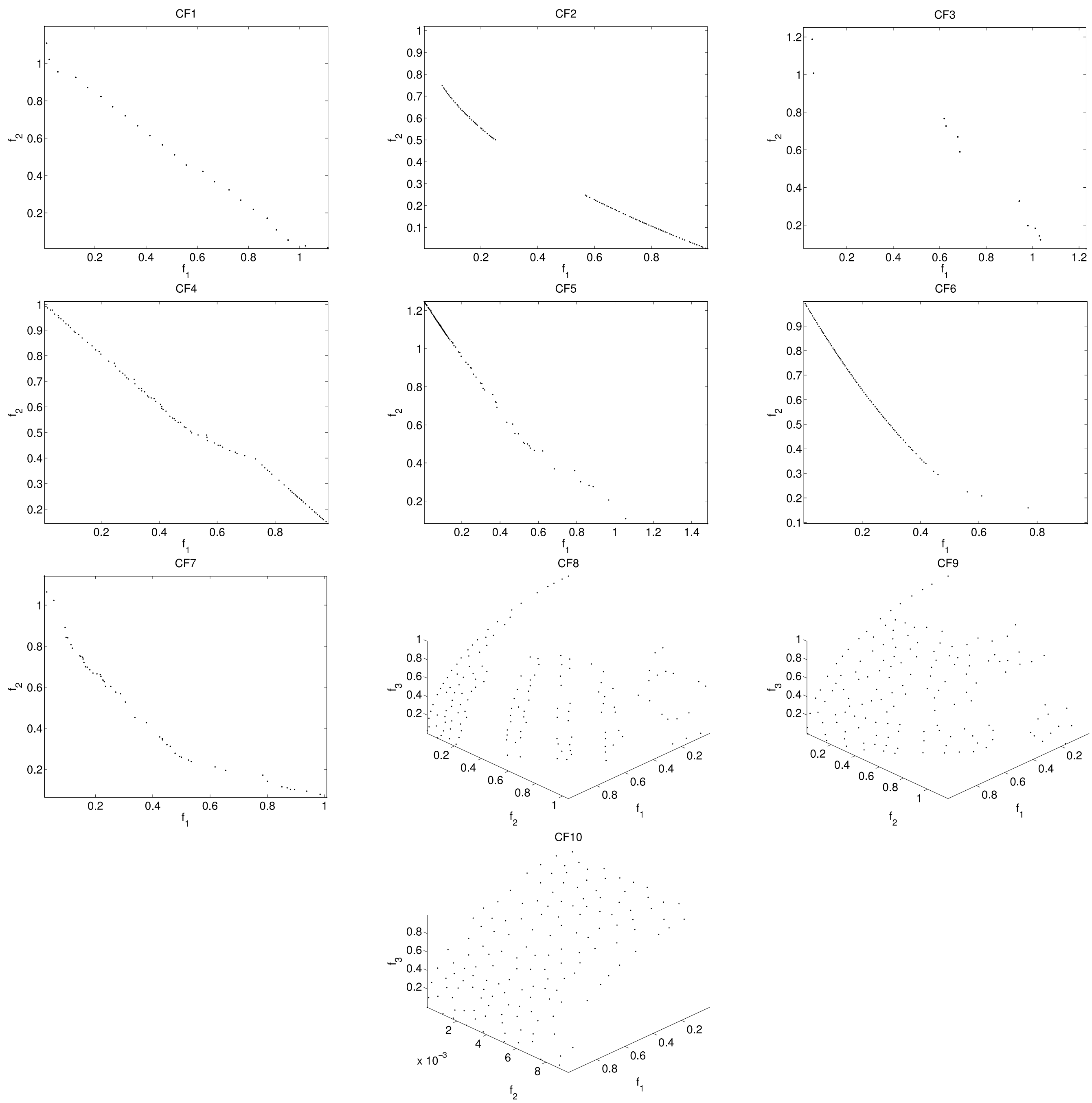

Fig. 3. The final approximation sets for the CF problems

TABLE II

MEAN IGD VALUES OVER ALL THE PROBLEMS AND ALL THE REPETITIONS FOR DIFFERENT CR AND F VALUES

CR	0.00	0.10	0.20	0.20	0.30	0.50	0.90	0.90	0.95	0.99	1.00
F	0.50	0.50	0.20	0.50	0.50	0.50	0.50	0.90	0.50	0.50	0.50
$\overline{IGD}$	9.04	26.83	37.23	42.22	48.25	56.80	68.40	88.80	70.96	72.22	70.16

TABLE III

THE AVERAGE NUMBER OF CONSTRAINT (g_1 & g_2) AND OBJECTIVE (f_1, f_2, & f_3) FUNCTION EVALUATIONS FOR CF1 – CF10. THE LAST COLUMN SHOWS PERCENTAGE OF ACTUAL NUMBER OF FUNCTION EVALUATIONS (AFES) COMPARED TO THE (USUAL) FULL AMOUNT OF FUNCTION EVALUATIONS

	g_1	g_2	f_1	f_2	f_3	AFES
CF1	300000	N/A	67984	67984	N/A	48.4%
CF2	300000	N/A	235819	235819	N/A	86.7%
CF3	300000	N/A	295999	295999	N/A	99.1%
CF4	300000	N/A	279055	279055	N/A	95.4%
CF5	300000	N/A	272688	272688	N/A	93.9%
CF6	300000	297502	296473	296473	N/A	99.2%
CF7	300000	272971	259986	259986	N/A	91.1%
CF8	300000	N/A	216924	216924	216924	79.2%
CF9	300000	N/A	255289	255289	255289	88.8%
CF10	300000	N/A	290785	290785	290785	97.7%

though the problems were complex and a higher CR is recommended in such cases with the single-objective DE.

REFERENCES

[1] S. Kukkonen and J. Lampinen, "GDE3: The third evolution step of Generalized Differential Evolution," in *Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005)*, Edinburgh, UK, Sept 2005, pp. 443–450, [online] http://www.iitk.ac.in/kangal/papers/k2005013.pdf, 6.11.2006.

[2] S. Kukkonen and K. Deb, "A fast and effective method for pruning of non-dominated solutions in many-objective problems," in *Proceedings of the 9th International Conference on Parallel Problem Solving from Nature (PPSN IX)*, Reykjavik, Iceland, Sept 2006, pp. 553–562.

[3] Q. Zhang, A. Zhou, S. Z. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," University of Essex and Nanyang Technological University, Essex, UK and Singapore, Technical Report CES-487, September 2008, [online] http://www3.ntu.edu.sg/home/EPNSugan/, 5.12.2008.

[4] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston: Kluwer Academic Publishers, 1998.

[5] J. Lampinen, "DE's selection rule for multiobjective optimization," Lappeenranta University of Technology, Department of Information Technology," Technical Report, 2001, [online] http://www.it.lut.fi/kurssit/03-04/010778000/MODE.pdf, 15.2.2006.

[6] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms–part I: A unified formulation," *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans*, vol. 28, no. 1, pp. 26–37, Jan 1998.

[7] R. Storn and K. V. Price, "Differential Evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec 1997.

[8] K. V. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin: Springer-Verlag, 2005.

[9] K. V. Price, *New Ideas in Optimization*. London: McGraw-Hill, 1999, ch. An Introduction to Differential Evolution, pp. 79–108.

[10] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, England: John Wiley & Sons, 2001.

[11] S. Kukkonen and J. Lampinen, "An extension of Generalized Differential Evolution for multi-objective optimization with constraints," in *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, Birmingham, England, Sept 2004, pp. 752–761.

[12] S. Kukkonen and K. Deb, "Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems," in *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)*, Vancouver, BC, Canada, July 2006, pp. 3995–4002.

[13] J. Lampinen, "Multi-constrained nonlinear optimization by the Differential Evolution algorithm," Lappeenranta University of Technology, Department of Information Technology," Technical Report, 2001, [online] http://www.it.lut.fi/kurssit/03-04/010778000/DECONSTR.PDF, 15.2.2006.

[14] ——, "A constraint handling approach for the Differential Evolution algorithm," in *Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002)*, Honolulu, HI, USA, May 2002, pp. 1468–1473.

[15] ——, *Soft Computing and Industry: Recent Advances*. Springer-Verlag, 2002, ch. Multi-Constrained Nonlinear Optimization by the Differential Evolution Algorithm, pp. 305–318.

[16] S. Kukkonen and J. Lampinen, "Constrained real-parameter optimization with Generalized Differential Evolution," in *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)*, Vancouver, BC, Canada, July 2006, pp. 911–918.

[17] J. Lampinen and R. Storn, *New Optimization Techniques in Engineering. Studies in Fuzziness and Soft Computing*. Springer-Verlag, 2004, vol. 141, ch. Differential Evolution, pp. 123–166.

[18] D. Zaharie, "Critical values for the control parameters of Differential Evolution algorithms," in *Proceedings of Mendel 2002, 8th International Conference on Soft Computing*, Brno, Czech Republic, June 2002, pp. 62–67.

[19] S. Kukkonen and J. Lampinen, "An empirical study of control parameters for Generalized Differential Evolution," in *Proceedings of the Sixth Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems (EUROGEN 2005)*, Munich, Germany, Sept 2005, 12 pages.

[20] ——, "An empirical study of control parameters for the third version of Generalized Differential Evolution (GDE3)," in *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)*, Vancouver, BC, Canada, July 2006, pp. 7355–7362.

[21] K. Deb, A. Sinha, and S. Kukkonen, "Multi-objective test problems, linkages, and evolutionary methodologies," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006)*, Seattle, WA, USA, July 2006, pp. 1141–1148.

[22] H. A. Abbass, "The self-adaptive Pareto Differential Evolution algorithm," in *Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002)*, Honolulu, HI, USA, May 2002, pp. 831–836.

[23] D. Zaharie, "Control of population diversity and adaptation in Differential Evolution algorithms," in *Proceedings of Mendel 2003, 9th International Conference on Soft Computing*, Brno, Czech Republic, June 2003, pp. 41–46.

[24] V. L. Huang, A. K. Qin, P. N. Suganthan, and M. F. Tasgetiren, "Multi-objective optimization based on self-adaptive differential evolution algorithm," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC 2007)*, Singapore, Sept 2007, pp. 3601–3608.

[25] K. Zielinski and R. Laur, "Differential evolution with adaptive parameter setting for multi-objective optimization," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC 2007)*, Singapore, Sept 2007, pp. 3585–3592.

[26] A. Zamuda, J. Brest, B. Bošković, and V. Žumer, "Differential evolution for multiobjective optimization with self adaptation," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC 2007)*, Singapore, Sept 2007, pp. 3617–3624.

[27] S. Kukkonen and J. Lampinen, "Performance assessment of Generalized Differential Evolution 3 (GDE3) with a given set of problems," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC 2007)*, Singapore, Sept 2007, pp. 3593–3600.

[28] S. Kukkonen, S. R. Jangam, and N. Chakraborti, "Solving the molecular sequence alignment problem with Generalized Differential Evolution 3 (GDE3)," in *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM 2007)*, Honolulu, HI, USA, April 2007, pp. 302–309.

[29] M. T. Jensen, "Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 5, pp. 503–515, Oct 2003.