2019

A Simple and Effective Termination Condition for Both Single- and Multi-Objective Evolutionary Algorithms

Kukkonen, Saku

IEEE

Artikkelit ja abstraktit tieteellisissä konferenssijulkaisuissa

http://dx.doi.org/10.1109/CEC.2019.8790292

A Simple and Effective Termination Condition for Both Single- and Multi-Objective Evolutionary Algorithms

Saku Kukkonen
Machine Learning Group
School of Computing
University of Eastern Finland
Email: dr.saku.kukkonen@gmail.com

Carlos A. Coello Coello
Departamento de Sistemas
UAM-Azcapotzalco
Mexico City, MEXICO
Email: ccoello@cs.cinvestav.mx

***Abstract*—In this paper, a simple and effective termination condition for both single- and multi-objective evolutionary algorithms has been proposed. The termination condition is based on simply observing objective values of solution candidates during generations. Effectiveness of the termination condition is self-evident with single-objective problems but unclear with multi-objective problems. Therefore, experiments with some well known bi- and tri-objective test problems have been performed. The proposed termination condition is implemented in Generalized Differential Evolution (GDE) that is a general purpose optimization algorithm for both single- and multi-objective optimization with or without constraints. Our preliminary results indicate that the proposed termination condition is a suitable termination condition also with multi-objective problems. With the termination condition and a control parameter adaptation technique previously introduced, GDE has become a fully automated optimization algorithm that can be used by any optimization practitioner.**

I. Introduction

Evolutionary Algorithms (EAs) [1] are population based stochastic methods useful especially with hard optimization problems. However, usability of EAs is limited by many parameters that need to be fixed before optimization. These parameters include termination condition, population size, and usually some method specific control parameters, *e.g.*, for crossover and mutation of an EA.

This paper focuses on finding a termination condition for both single- and multi-objective EAs. The problem of deciding a termination condition is as a very old one. The most commonly used termination condition is the number of generations or function evaluations. This is suitable with easy problems and test problems but when dealing with real-world problems, it would be desirable to have a termination condition that is automatic and based on convergence of the search, since otherwise probably too few or too many fitness function evaluations are performed. Besides the number of generations, many different types of termination conditions have been used such as the maximal allowed CPU time elapsed, lack of improvement in the found fitness value, and/or the population diversity drops under some given threshold [1, pp. 23–24]

Determining the termination condition for multi-objective EAs (MOEAs) is harder than for single-objective EAs since the search does not converge to a single solution but multiple solutions. A taxonomy of online termination criteria for MOEAs has been proposed in [2]. That paper also contains a survey of sophisticated termination conditions used with MOEAs, *e.g.*, in [3], [4]. Those conditions often involve maintaining an archive for previous generations / best solutions candidates, different kinds of progress indicators, and the use of statistics. To the authors' best knowledge no simple effective termination condition for MOEAs has been proposed so far.

The remainder of this paper is organized as follows: Background of multi-objective optimization and EAs used in this paper are described in Section II. Section III describes the proposed termination condition. Section IV provides performance evaluation of the proposed termination condition. Finally, our conclusions and discussion are provided in Section V.

II. Background

A. Constrained Multi-Objective Optimization

A multi-objective optimization problem (MOOP) with constraints can be presented in the form [5, p. 37]:

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

Thus, there are M functions to be minimized and K inequality constraints. Maximization problems can be converted to minimization problems, and all the constraints can be converted into the form $g_k(\vec{x}) \leq 0$. Thereby, the formulation above is without loss of generality. Unconstrained single-objective optimization is a special case of the above definition with $M = 1$ and $K = 0$.

The goal of evolutionary based multi-objective optimization is to find an approximation of the Pareto-front, *i.e.*, to find a set of solutions that are not dominated by any other solution. A weak dominance relation $\preceq$ between two vectors is defined in such a way that $\vec{x}$ weakly dominates $\vec{y}$, *i.e.*, $\vec{x} \preceq \vec{y}$ iff $\forall i : f_i(\vec{x}) \leq f_i(\vec{y})$. The dominance relation $\prec$ between two vectors is defined in such a way that $\vec{x}$ dominates $\vec{y}$, *i.e.*, $\vec{x} \prec \vec{y}$ iff $\vec{x} \preceq \vec{y} \ \wedge \ \exists i : f_i(\vec{x}) < f_i(\vec{y})$. The dominance relationship can be extended to take into consideration constraint values and objective values at the same time. Constraint-domination $\prec_c$ is defined in this paper so that $\vec{x}$ constraint-dominates $\vec{y}$, *i.e.*, $\vec{x} \prec_c \vec{y}$ iff any of the following conditions is true [6]:

- $\vec{x}$ and $\vec{y}$ are feasible and $\vec{x}$ dominates $\vec{y}$ in objective function space.
- $\vec{x}$ is feasible and $\vec{y}$ is not.
- $\vec{x}$ and $\vec{y}$ are infeasible and $\vec{x}$ dominates $\vec{y}$ in constraint function violation space.

The definition for weak constraint-domination $\preceq_c$ is analogous by the dominance relation changed to weak dominance in the above definition.

B. 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. Usually a predefined upper limit (G_{max}) for the number of generations to be computed is used as the termination condition. Other control parameters for DE are the crossover control parameter (*CR*), the mutation factor (F), and the population size (*NP*).

At each generation G, DE goes through each D dimensional decision vector $\vec{x}_{i,G}$ of the population and creates the corresponding trial vector $\vec{u}_{i,G}$ as follows [9]:

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

This is the most common DE version, DE/rand/1/bin, also known as the classic DE. Functions $rand_i[0,1)$ and $rand_j[0,1)$ return a random number drawn from the uniform distribution between 0 and 1 for each i and j. 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 $(0, 1+]$ (*i.e.* larger than 0 and upper limit is around 1 although there is no hard upper limit). In practice, *CR* controls rotational invariance of the search, and a small value for it (*e.g.*, 0.1) is useful with separable problems while larger values (*e.g.*, 0.9) are useful 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 a similar effect on the convergence rate as F has. [10]

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.

C. Generalized Differential Evolution

Generalized Differential Evolution (GDE) [6] is an extension of DE for constrained multi-objective optimization. There exist several development versions of GDE that are shortly described below. All the GDE versions can handle different numbers of M objectives and a different 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 DE, and this is why they are referred to as *Generalized* DEs.

The first version of GDE extended DE for constrained multi-objective optimization, and it modified only the selection rule of the basic DE [11]. 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 the old vector. There was no explicit sorting of non-dominated solutions [12, 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.

The second version, GDE2, made the selection based on crowdedness when the trial and old vector were feasible and non-dominating with respect to each other in the objective space [13]. This improved the extent and distribution of the obtained set of solutions but slowed down the convergence of the overall population because it favored isolated solutions far from the Pareto-front until all the solutions had converged near the Pareto-front.

The third version is known as GDE3 [14], [15]. Besides the selection, another part of the basic DE has also been modified. Now, in the case of feasible and non-dominating solutions, both solutions 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. GDE3 can be conspired as an improved version of the elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) [16] that is the most used multi-objective optimization method in the MOEA literature.

The pruning technique used in the original GDE3 is based on crowding distance of NSGA-II, 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 [15] [1]. For problems having more than two objectives, a more general diversity maintenance technique proposed in [18] is used. The technique is based on a crowding estimation using the nearest neighbors of solutions in an Euclidean sense, and an efficient nearest neighbors search technique. GDE3 has performed well both in several academic studies [10], [19], [20] and in several practical applications [21]–[25], *e.g.*, NASA has applied GDE3 for solving some space science optimization problems [26]. Therefore, GDE3 has been selected as an optimization method of this study.

GDE has the same control parameters as DE. There already exist studies of automatic control parameter adaptation techniques both for single- and multi-objective DEs. In [27] the Exponential Weighting Moving Average (EWMA) control parameter adaptation technique was implemented in GDE3 and was found to perform well both in single- and multi-objective optimization. In the same paper, some rules for selecting the population size are provided. Thus, only an automatic termination condition is missing from a fully automated GDE method.

III. The Proposed Termination Condition

The basic idea is to calculate an indicator value S that is the sum of objective values [2] in the population of feasible solutions. Formally:

$$S_G = \sum_{n=1}^{NP} \sum_{m=1}^{M} f_{G,n,m} \quad ,$$

where $f_{G,n,m}$ is the mth objective value of nth population member in generation G. Reasoning behind the indicator is that its value will decrease until reaching convergence since the goal of optimization is to minimize objectives [3]. However, since the search might temporally stop or fluctuate, the value of S is not used directly as a termination condition but instead, the history of S values is used. For this purpose a table $\mathbf{H}$ is used. The length of the table is H_{max} and it determines desired certainty / reliability for the convergence. Last S values are stored in $\mathbf{H}$ and the sum of elements in $\mathbf{H}$ is used as a termination condition. At generation G, the sum is:

$$H_G = \sum \mathbf{H} = \sum \left[S_G, S_{G-1}, S_{G-2}, \ldots, S_{G-H_{max}+1}\right]$$

At the beginning of the search, $\mathbf{H}$ is initialized with larger values than S_0 (*i.e.*, S value of the initial population). Search is continued as long as $H_G < H_{G-1}$ (one can use the maximal number of generations as a second termination condition just in case).

IV. Performance Evaluation

All the implementations were made in Matlab and tests were performed in an ordinary laptop. In all the tests, $H_{max} = 50$ was used for reliability. All the problems have limits for decision variable values. If these limits were violated [4] then the violation was corrected using the following rule:

$$x_i = \begin{cases} 2x_i^{(lo)} - x_i & \text{if} \quad x_i < x_i^{(lo)} \\ 2x_i^{(up)} - x_i & \text{if} \quad x_i > x_i^{(up)} \end{cases} ,$$

where x_i^{lo} and x_i^{up} represent the lower and upper bounds for variable x_i, respectively. This rule reflects violated variable values inside violated boundary by the amount of the violation [5].

A. Single-Objective Optimization

Even without numerical experiments, one can figure out the behavior of a termination condition in single-objective optimization with DE or GDE. Since the optimization method is elitist, the sum of objective values in the population will not deteriorate. The search terminates when there has not been improvement in the population during the last H_{max} generations. Termination happens because of stagnation or convergence. For resolving which one has been the reason for termination, one can repeat search several times. If the solution is always the same, there is a good probability that the solution is optimal.

The termination condition was tested with two classical single-objective multi-modal test problems, Rastrigin's and Schwefel's functions with 20 variables. In termination, all the population members had the optimal objective value as assumed.

B. Multi-Objective Optimization

For evaluating the performance of the proposed termination condition in multi-objective optimization, it was implemented in GDE3 and tested with some well known bi- and tri-objective test problems. Most often problems appearing in studies have had only two or three objectives [28, p. 305] and therefore no higher number of objectives were tested. The EWMA control

[1] Although crowding distance works well only with two objectives, it is still used in studies (such as in [17]) with more than two objectives.

[2] In this paper objective values are used directly, but they could be also normalized using magnitudes of objectives determined from the initial population. Also, instead of calculating sum of objective values, calculating mean of objective values could be used.

[3] In multi-objective optimization individual objective values might increase while other objective values decrease but the sums of objective values are assumed to decrease during the search if the optimization method is able to minimize objective values.

[4] DE and GDE are able to create new solutions candidates outside the original initialization range and / or current population.

[5] With certain test problems, Pareto-optimal solutions lay on the boundary of the decision variable space and the constraint violation correction rule (especially such one that corrects the violation to the boundary) can have a great impact on convergence. The reflection rule is general and does to take advantage of the problem's features.

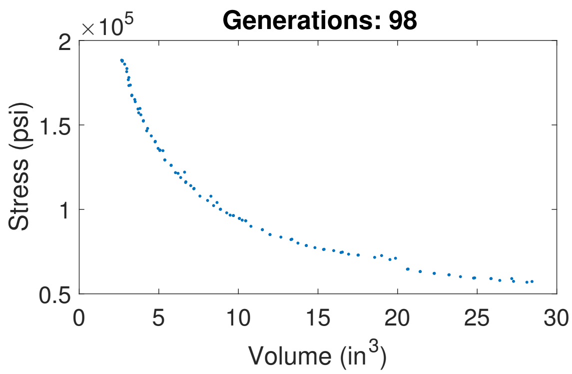

Fig. 3. The final population at termination for the Spring Design problem.

parameter adaption technique and diversity maintenance technique for many-objective (more than two) optimization were also implemented with GDE3. The population size and initial control parameter values are $NP = 100 \cdot (M - 1)$, $CR = 0.2$, and $F = 0.2$.

The first validation was performed with the ZDT test problems [12, pp. 356–360] that are bi-objective and were very popular at the beginning of this century. For assessing the quality of obtained solutions, Inverted Generational Distance (IGD) [29] was used. IGD measures the average distance of the Pareto-front to the solutions and it measures overall quality of the solution candidates. The optimal value is zero. Figure 1 shows the final populations and IGD curves for the ZDT problems. It can be observed that all the ZDT problems have a good convergence to the true Pareto-fronts.

The second validation was performed with the DTLZ test problems [30] that are well-known tri-objective test problems. Figure 2 shows the final populations and IGD curves for the DTLZ problems. Well-converged good approximations of the Pareto-fronts can be observed also with these problems [6]

Since the previous test problems are artificial with the same value range for the objectives and without constraints, the termination condition was tested also with the Spring Design problem [31], [32]. The problem has two objectives that have totally different value ranges (magnitude difference is about 10 000). Besides the objectives, the problem has eight constraints. Figure 3 illustrates the final population at termination. This problem does not have an analytical Pareto-front but based on earlier experiments with the problem (*e.g.*, in [6]) the obtained solution is optimal.

V. Conclusions and Discussion

A termination condition, which is suitable for both single- and multi-objective optimization has been proposed and evaluated. Based on our experiments, the condition is suitable for multi-objective optimization although the condition is very simple. The condition has one parameter value, but fixing it is much easier than fixing, *e.g.*, the number of generations.

One should note that this termination condition does not guarantee convergence to the global optimum. If the problem is difficult and / or the optimization method is not capable to find the global optimum, it is clear that the proposed termination condition cannot guarantee convergence either. Therefore, it is recommended to perform several independed optimization runs. If the solution is the same in several repetitions, one can assume with good confidence that the solution is optimal.

The termination condition was implemented in Generalized Differential Evolution (GDE). Other parameters were also automatically set / adapted. Therefore, GDE performed now fully automatically without any user-defined parameters. This would make the method usable for any practitioner.

There are probably several improvements that one can make to GDE, but the method is already now suitable for general optimization. This completes the work of the first author on GDE started in 2003. The next step would be to publish the program codes of the method.

[6] For DTLZ2, IGD value does not reach absolute zero because the population size is not large enough to cover the Pareto-front totally.

Acknowledgments

The first author would like to acknowledge support of the South Savo Regional Fund of the Finnish Cultural Foundation (Suomen Kulttuurirahaston Etelä-Savon rahasto), the Emil Aaltonen Foundation, and the Academy of Finland. The second author gratefully acknowledges support from CONACyT grant no. 2016-01-1920 (*Investigación en Fronteras de la Ciencia 2016*). The second author is on sabbatical leave from CINVESTAV-IPN (Departamento de Computación), in Mexico City, México.

References

[1] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Springer-Verlag Berlin Heidelberg, 2003.

[2] T. Wagner, H. Trautmann, and L. Martí, "A Taxonomy of Online Stopping Criteria for Multi-Objective Evolutionary Algorithms," in *Evolutionary Multi-Criterion Optimization, 6th International Conference, EMO 2011*, R. H. Takahashi, K. Deb, E. F. Wanner, and S. Grecco, Eds. Ouro Preto, Brazil: Springer. Lecture Notes in Computer Science Vol. 6576, April 2011, pp. 16–30.

[3] L. Marti, J. Garcia, A. Berlanga, and J. M. Molina, "A Stopping Criterion for Multi-Objective Optimization Evolutionary Algorithms," *Information Sciences*, vol. 367, pp. 700–718, November 2016.

[4] T. Goel and N. Stander, "A non-dominance-based online stopping criterion for multi-objective evolutionary algorithms," *International Journal For Numerical Methods in Engineering*, vol. 84, no. 6, pp. 661–684, November 5 2010.

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

[6] S. Kukkonen, "Generalized Differential Evolution for global multi-objective optimization with constraints," Ph.D. dissertation, Lappeenranta University of Technology, Acta Universitatis Lappeenrantaensis 475, May 2012.

[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, "An introduction to Differential Evolution," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. London: McGraw-Hill, 1999, pp. 79–108.

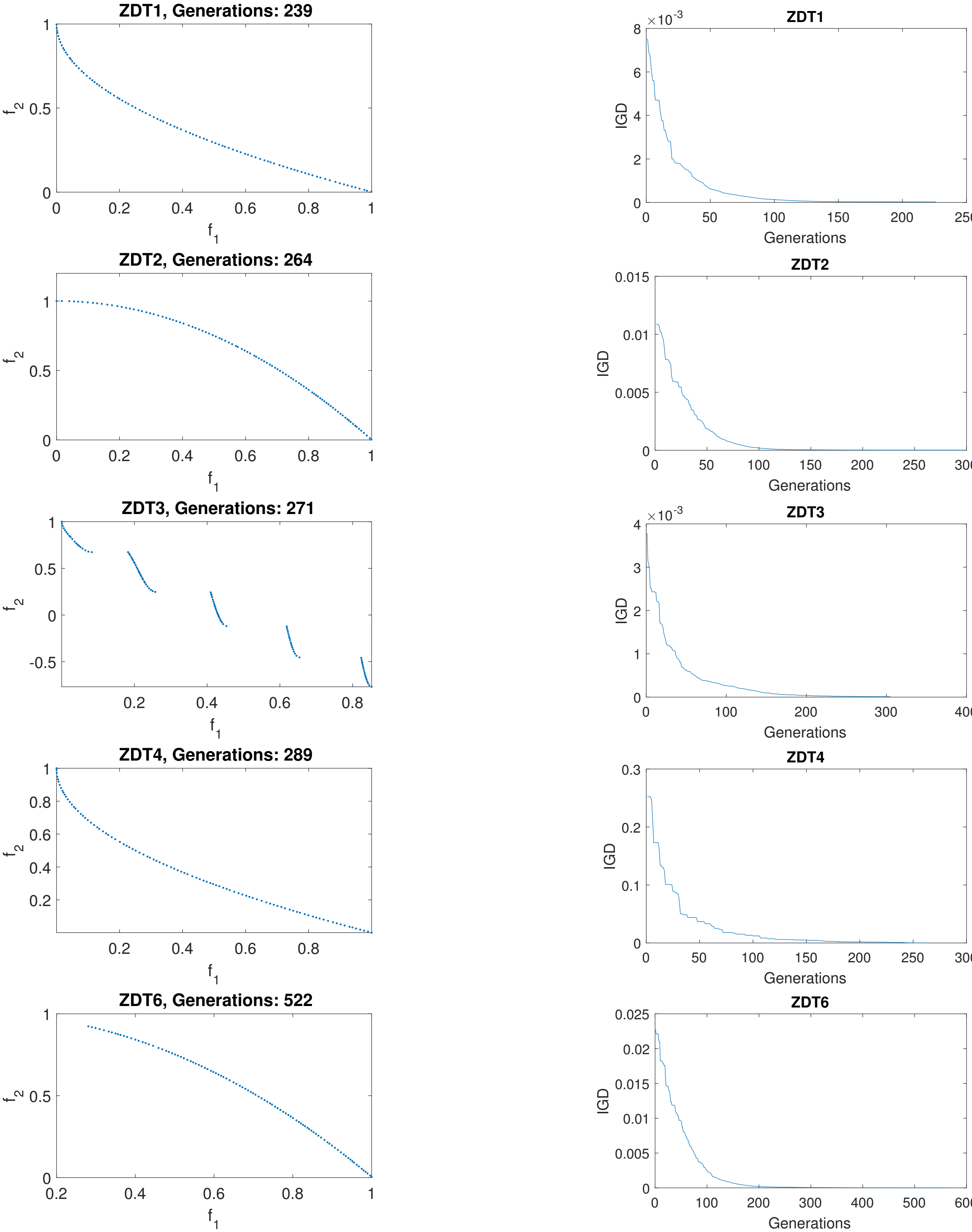

Fig. 1. Final populations and Inverted Generational Distance curves for the ZDT test problems.

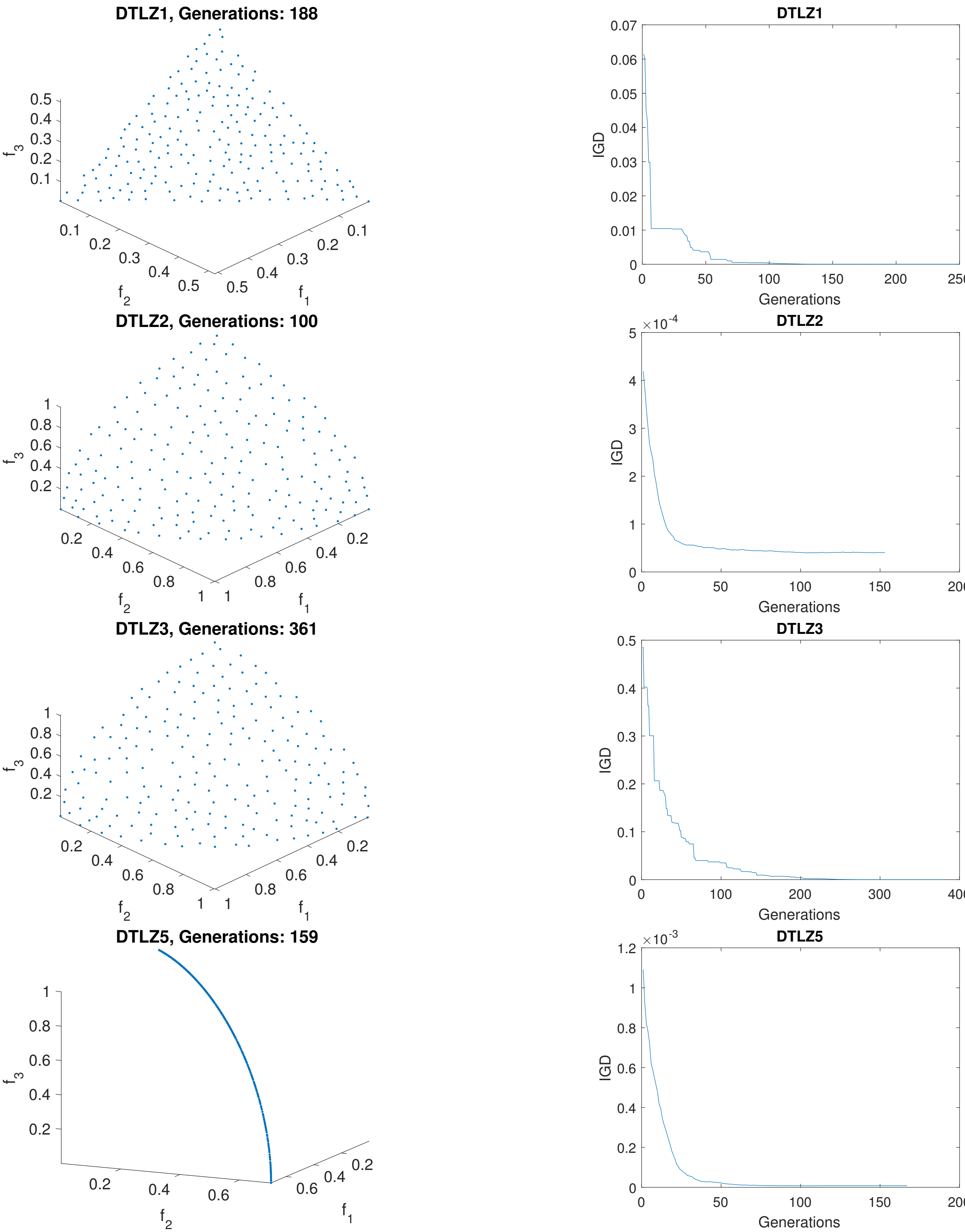

Fig. 2. Final population and Inverted Generational Distance curves for the DTLZ test problems.

[10] S. Kukkonen and J. Lampinen, "Constrained real-parameter optimization with Generalized Differential Evolution," in *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)*, G. G. Yen, S. M. Lucas, G. Fogel, G. Kendall, R. Salomon, B.-T. Zhang, C. A. Coello Coello, and T. P. Runarsson, Eds. Vancouver, BC, Canada: IEEE Press, July 2006, pp. 911–918.

[11] ——, "A Differential Evolution algorithm for constrained multi-objective optimization: Initial assessment," in *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2004)*, M. H. Hamza, Ed. Innsbruck, Austria: ACTA Press, Feb 2004, pp. 96–102.

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

[13] 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)*, ser. Lecture Notes in Computer Science (LNCS), vol. 3242, Birmingham, England, Sept 2004, pp. 752–761.

[14] ——, "GDE3: The third evolution step of Generalized Differential Evolution," in *Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005)*. Edinburgh, Scotland: IEEE Press, Sept 2005, pp. 443–450.

[15] 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)*, G. G. Yen, S. M. Lucas, G. Fogel, G. Kendall, R. Salomon, B.-T. Zhang, C. A. Coello Coello, and T. P. Runarsson, Eds. Vancouver, BC, Canada: IEEE Press, July 2006, pp. 3995–4002.

[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, April 2002.

[17] A. A. Bidgoli, S. Mahdavi, S. Rahnamayan, and H. Ebrahimpour-Komleh, "GDE4: The Generalized Differential Evolution with Ordered Mutation," in *Proceedings of the 10th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2019)*, ser. Lecture Notes in Computer Science (LNCS), K. Deb, E. Goodman, C. A. Coello Coello, K. Klamroth, K. Miettinen, S. Mostaghim, and P. Reed, Eds., vol. 11411, East Lansing, MI, USA, March 2019, pp. 101–113.

[18] 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)*, ser. Lecture Notes in Computer Science (LNCS), T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, and X. Yao, Eds., vol. 4193. Reykjavik, Iceland: Springer-Verlag, Berlin, Heidelberg, Sept 2006, pp. 553–562.

[19] 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)*, D. Srinivasan and L. Wang, Eds. Singapore: IEEE, Sept 2007, pp. 3593–3600.

[20] ——, "Performance assessment of Generalized Differential Evolution 3 with a given set of constrained multi-objective test problems," in *Proceedings of the 2009 IEEE Congress on Evolutionary Computation (CEC 2009)*, Trondheim, Norway, May 2009, pp. 1943–1950.

[21] 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: IEEE, April 2007, pp. 302–309.

[22] K. Tagawa, "Multi-objective optimum design of balanced SAW filters using generalized Differential Evolution," *WSEAS Transactions on Systems*, vol. 8, no. 8, pp. 923–932, 2009.

[23] S. K. Goudos and J. N. Sahalos, "Pareto optimal microwave filter design using multiobjective Differential Evolution," *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 1, pp. 132–144, 2010.

[24] S. K. Goudos, K. Siakavara, E. Vafiadis, and J. N. Sahalos, "Pareto optimal Yagi-Uda antenna design using multi-objective Differential Evolution," *Progress In Electromagnetics Research*, vol. 10, pp. 231–251, 2010.

[25] M. M. Bech, C. Noergaard, D. B. Roemer, and S. Kukkonen, "A global multi-objective tool for design of mechatronic components using Generalized Differential Evolution," in *Proceedings of the IECON 2016 - 42th Annual Conference of the IEEE Industrial Electronics Society*. Florence, Italy: IEEE, Oct 2016, pp. 475–481.

[26] M. D. Johnston and M. E. Giuliano, "Multi-objective scheduling for space science missions," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 15, no. 8, pp. 1140–1148, 2011.

[27] S. Kukkonen and C. A. Coello Coello, "Applying exponential weighting moving average control parameter adaptation technique with generalized differential evolution," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC 2016)*. Vancouver, Canada: IEEE, July 2016, pp. 4755–4762.

[28] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. Springer Science+Business Media, 2007.

[29] C. A. Coello Coello and N. Cruz Cortés, "Solving multiobjective optimization problems using an artificial immune system," *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, 2005, inverted generational distance is a proposal of an anonymous reviewer.

[30] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002)*, X. Yao, Ed. Honolulu, HI, USA: IEEE Service Center, Piscataway, NJ, USA, May 2002, pp. 825–830.

[31] B. K. Kannan and S. N. Kramer, "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design," *Journal of Mechanical Design*, vol. 116, no. 2, pp. 405–411, 1994.

[32] K. Deb, A. Pratap, and S. Moitra, "Mechanical component design for multiple objectives using elitist non-dominated sorting GA," in *Proceedings of the Parallel Problem Solving from Nature VI (PPSN VI)*, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, Eds. Paris, France: Springer, 2000, pp. 859–868.