 DOI: 10.1109/TEVC.2017.2673962

Improving evolutionary algorithms in a continuous domain by monitoring the population midpoint

Jarosław Arabas, Rafał Biedrzycki
Institute of Computer Science
Warsaw University of Technology
email: {jarabas, rbiedrzy}@elka.pw.edu.pl

***Abstract*—It is advocated that monitoring the population midpoint allows for improving the efficiency of population based evolutionary algorithms in $\mathbb{R}^d$. The theoretical motivation supporting this hypothesis is provided in this paper, and this phenomenon is empirically confirmed for selected typical evolutionary algorithms by a series of tests for fitness functions contained in the CEC2005 and CEC2013 benchmark sets.**

***Index Terms*—CEC benchmark set, convergence, population midpoint**

I. Introduction

EVOLUTIONARY algorithms (EAs) are iterative stochastic search techniques. The state of an EA is characterized by a population of individuals which are points in the search space. In each iteration, new individuals are randomly generated according to a procedure that includes the reproduction of individuals from the current population followed by their crossover and mutation. The whole process is based on the natural evolution model by Darwin and his followers.

This contribution explores a practical consequence of the theoretical achievements that have been made in the analysis of dynamics of populations processed by various types of EAs for a continuous domain. A common result of this analysis, for several important types of EAs, is that the expectation vector of populations converges in some stochastic sense to the optimum of the fitness function.

For classical EAs with Gaussian mutation, this result was achieved using the "infinite population size" model developed in [1], [2]. For this model it was proved, e.g. in [3], that if the fitness function is a Gauss function, then the expected population mean converges to the global optimum of the fitness function and the expected population variance converges to a characteristic value which can be predicted in advance when the EA parameters are known. The results hold for any even, unimodal, concave fitness function.

For Differential Evolution (DE), it has been reported that the probability distribution function of points in consecutive generations should converge to Dirac's delta, located in the local optimum [4]. This implies that the expected value of generated points should stochastically converge to the optimum. There is also some experimental evidence about the profits of computing the midpoint when the population is evaluated in parallel on a GPU [5].

According to the aforementioned theoretical findings, it can be expected that a midpoint of the current population should stochastically converge to the local optimum of the fitness function, provided that the population has settled down in its basin of attraction. Consequently, computing the population midpoint should improve effectiveness in the exploitation phase of the evolutionary search. The price to pay for this improvement is the computing cost of evaluating an additional point in the population.

In the analysis presented here it is assumed that the midpoint will not become a population member. We do not change the way in which individuals are processed or generated by the analyzed algorithms. We believe that addition of the midpoint to the population would result in processing more compact populations which would introduce a higher risk of premature convergence.

In this paper we provide a series of arguments for computing the fitness value of the population midpoint. We start in Section II, where we provide a statistical analysis of a set of points which are generated randomly with a multidimensional normal distribution around some expectation vector. Statistical analysis shows that the population mean will usually be closer to the expectation vector than any other point in the population.

More practical issues are studied in Section III, where various types of EAs are used for searching for the optimum value of the quadratic function. Two definitions of midpoint are examined: arithmetic mean and the median. The results of conducted experiments confirm a quality gain due to computing the fitness value of the population midpoint.

The most realistic arguments are raised in Section IV, where the results of benchmarking using the CEC2005 [6] and CEC2013 [7] benchmark sets are provided. We analyze various types of algorithms, with and without computing the population midpoint. In the set of considered algorithms we include not only plain classical EAs and DE, but also a few selected methods which achieved good results at CEC competitions: CMA-ES[8], JADE [9], SADE [10] and b6e6rl [11]. We assume the finite budget according to the benchmarking criteria and the results are compared to the statistical basis using the Wilcoxon test. This methodology is in agreement with recommendations published in [12]. The results confirm the thesis on the profitability of midpoint analysis. The paper is briefly concluded with Section V.

II. Standardized normal variates in $\mathbb{R}^d$

Prior to discussing any EA, let us study some phenomena that can be observed in a population of d-dimensional normal variates. These effects relate to a situation when an EA has converged to a local optimum of a fitness function.

Consider a Euclidean norm of a vector in $\mathbb{R}^d$:

$$||\vec{x}|| = \sqrt{\sum_{i=1}^{d}(x_i)^2} \tag{1}$$

which takes its minimum at $\vec{0}$.

Assume a population $X \subset \mathbb{R}^d$ containing N points which were independently generated with a standardized, d-dimensional normal distribution. In other words, the coordinates of each point $\vec{x} \in X$ are independent and identically distributed standardized normal variates. Recall that the root of the sum of squares of d independent normal variates has a chi distribution with d degrees of freedom [13]. In our example, this will be the distribution of the Euclidean norm of points contained in X. Since they are generated randomly and the expectation vector is zero, the norm value of point $\vec{x} \in X$ equals the distance between $\vec{x}$ and the expectation vector.

Note that for $d \geq 2$, the most probable values of $||\vec{x}||$ span a range which does not include values very close to zero, which follows from properties of the chi distribution. Moreover, the average value of $||\vec{x}||$ increases along with $\sqrt{d}$.

$F_b(y)$ — the cumulative distribution function (cdf) — of the smallest value of points from X can be derived directly from the definition of the cdf:

$$\begin{aligned} F_b(y) &= Pr(||b(X)|| \leq y) = 1 - Pr(||b(X)|| > y) = \\ &= 1 - Pr(\min_{\vec{x}\in X}(||\vec{x}||) > y) = \\ &= 1 - \prod_{\vec{x}\in X}(1 - Pr(||\vec{x}|| > y)) = 1 - (1 - F_\chi(y))^N \end{aligned} \tag{2}$$

where F_χ is the cdf of the chi distribution with d degrees of freedom, and $b(X)$ is the point from X with the smallest norm value:

$$b(X) = \arg\min_{\vec{x}\in X} ||\vec{x}|| \tag{3}$$

Note that the point $b(X)$ is the best approximation of the expectation vector among all points from X, since the expectation vector is $\vec{0}$.

The mean vector of the population X is defined as

$$m(X) = \frac{1}{N}\sum_{\vec{x}\in X}\vec{x} \tag{4}$$

Recall that $\vec{x} \in X$ are standardized normal independent variates. Therefore, the mean vector $m(X)$ is normally distributed with zero expectation and its covariance matrix equals I/N, where I is the identity matrix. This means that the value of $||m(X)||$ is chi distributed and its cdf equals:

$$F_m(y) = F_\chi(N \cdot y) \tag{5}$$

Plots of the cdf of the best and the mean point in the population of d-dimensional standardized normal random variates are depicted in Fig. 1 for different values of the population size N and the dimension d. It can be observed that for any considered combination of d and N, the probability of observing a point whose norm value (and in the same time the distance from the expectation vector) falls into the range

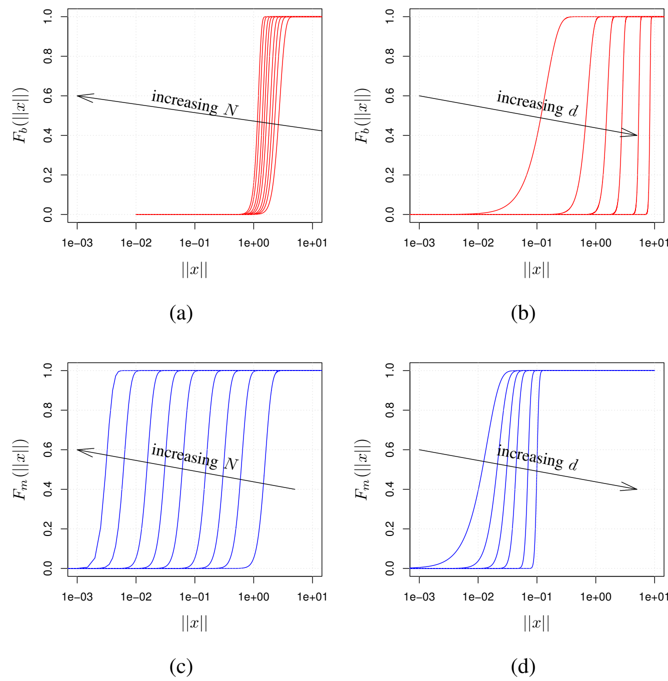

Fig. 1. Semilog plots of the cdf of observed norm values for the best point (a,b) and the mean point (c,d) of the population. Plots were generated for $d = 10$ and $N \in \{2, 5, 10, 20, 50, 100, 200, 500, 1000\}$ (a,c) and for $N = 100$ and $d \in \{2, 5, 10, 20, 50, 100\}$ (b,d)

$(0, a)$ is always greater for the midpoint $m(X)$ than for the best point $b(X)$ for all $a > 0$. Hence, the population midpoint will approximate the expectation vector much better than the best point from the population. The superiority of $m(X)$ over $b(X)$ grows along with the population size N and the search space dimensionality d.

Recall that for several types of EAs it can be proved that the expectation vector of the population converges stochastically to the fitness optimum. When this occurs, the expected value of the midpoint will converge as well. Moreover, if the fitness function is locally convex, the population midpoint may be located closer to the local optimum than any point from the population.

III. Dynamics of Populations for the Quadratic Fitness Function

This section illustrates relations between the midpoint fitness and the best point fitness assuming the quadratic fitness function. Quadratic function is a convex function with a unique optimum, therefore global optimization techniques are not methods of the first choice in this case. Nevertheless, the experimental analysis for the quadratic function allows for checking efficiency of evolutionary techniques in precise location of optimum.

In certain classes of optimization problems, quadratic function may be a good local approximation of the fitness function nearby local optima. Good examples are the problems of least-squared fitting nonlinear models to data.

Observations made in the previous section indicate that if the population were generated with the expectation vector which would perfectly match the optimum, then it could be expected that the population midpoint would usually be a better approximation of the optimum than the best point in the population. For realistic EAs, even when populations converge to the optimum, they will approach it in an irregular random fashion. Then the expectation vector of the population is not perfectly matched with the optimum and it cannot be assumed that the population is a set of independent normal variates either.

Then the arithmetic mean might be not the best estimator of the local optimum and robust statistics might work better. Therefore, in addition to the arithmetic mean, the population median point is considered. Its j-th coordinate is the median of the j-th coordinate of points from the population X:

$$m(X)_j = \text{median}(X_{1,j}, ..., X_{N,j}) \tag{6}$$

where

$$\text{median}(\{s_1, ..., s_N\}) = \begin{cases} s_{\frac{1}{2}(N+1)} & \text{when } N \text{ is odd} \\ \frac{1}{2}(s_{\frac{N}{2}} + s_{\frac{N}{2}+1}) & \text{when } N \text{ is even} \end{cases} \tag{7}$$

assuming that the set $\{s_1, ..., s_N\}$ is sorted.

Below we illustrate dynamics of populations' best fitness and the quality gain due to computing the midpoint fitness for the quadratic fitness function:

$$q(\vec{x}) = \sum_{i=1}^{d} x_i^2 \tag{8}$$

assuming the number of dimensions $d = 10$.

We considered seven EAs — three of them represented "classical" methods and four others were sophisticated methods whose versions were among winners of black box optimization competitions:

1) GEA: a generational EA with Gaussian mutation with an identity covariance matrix, arithmetic crossover with probability 0.7, and nonelitist binary tournament selection,
2) DE: a classical DE/rand/1/bin algorithm with the scaling factor $F = 0.9$ and the crossover rate $CR = 0.9$,
3) ES: the $(\mu + \lambda)$ Evolution Strategy with self adaptation of the vector of mutation variance values, ($\mu = 25, \lambda = 100$),
4) CMA-ES [8] which was implemented in [14],
5) b6e6rl [11] which was implemented in [15],

6–7) own implementation of SADE [10] and JADE [9], based on cited papers.

For CMA-ES and b6e6rl we accepted default parameter values and for SADE and JADE we followed the parameter values suggested in cited papers.

In Fig. 2 we provide convergence curves of the fitness function computed for the arithmetic mean, the median and for the best point in the population. The curve value for iteration t represents the best-so-far value of the best points or midpoints that have been observed from the first to the t-th generation. All algorithms generated $N = 100$ individuals in each generation except for CMA-ES, which evaluated only $N = 10$ points. The population was initialized with uniform distribution in a rectangle $(-100, 100)^d$.

The simulation results show that for the quadratic fitness function, the fitness value of the population midpoint usually converged faster than the value for the best point in the population. The convergence speed was significantly different for compared methods, which can be observed when looking at different span of values on the y-axis of plots in Fig. 2. Therefore a direct comparison of convergence curves for different methods is impossible. A better insight into the results can be gained by analyzing the ratio of fitness values of the best-so-far midpoint and the best-so-far individual for each generation:

$$r(t) = \frac{\min_{k=1,...,t}(q(b(X_k)))}{\min_{k=1,...,t}(q(m(X_k)))} \tag{9}$$

where X_k stands for the population in the k-th generation, q is the fitness function and $m(X_k), b(X_k)$ are the midpoint and best point of X_k. The evolution of the $r(t)$ value in consecutive generations is plotted in Fig. 3 for both the mean point and the median point.

The fitness of the population mean was on average 32 times smaller than the best individual for DE. For GEA, JADE, ES and b6e6rl, values of $r(t)$ fluctuated around 12, 13, 14 and 15, respectively. In the case of SADE, $r(t)$ took values around 7 and for CMA-ES — around 2.

The ratio $r(t)$ for the median point was generally smaller than for the mean point, which indicates that the mean point converged faster than the median for the quadratic fitness function.

IV. Experiments for fitness functions from CEC2005 and CEC2013 benchmark sets

Observations made for the quadratic function encourage to formulate a hypothesis that the quality of results yielded by EAs will be improved by monitoring the population midpoint. Such improvement can be expected in the exploitation phase when the population is contained by the attraction basin of a local optimum.

Possibility of such improvement was experimentally verified for optimization problems defined in the CEC2005 [6] and CEC2013 [7] benchmark sets.

We performed experimental analysis, making two extreme assumptions about how to consider the midpoint. In the first case, it was assumed that computing the midpoint is cost-free. In the second case, the midpoint costs as much as a single point in one generation, and the number of individuals generated in each iteration was decreased by one. The analysis covered results generated by GEA, DE, ES, CMA-ES, JADE, SADE and b6e6rl for 10, 30 and 50 dimensions, for all functions from the CEC2005 and CEC2013 benchmark sets.

Since the goal of the paper is to illustrate an improvement introduced by midpoint monitoring rather than to beat the best result for the CEC benchmark sets, the parameters of the compared EAs were set as in Section III without any tuning.

a) Conditions of experiments: Both benchmark sets contain problems with box constraints. In the experiments, these problems were handled by a repairing mechanism. Every time an infeasible mutant $\vec{x}$ was generated, it was substituted by a feasible point $\vec{y}$ which was created by reflecting all infeasible

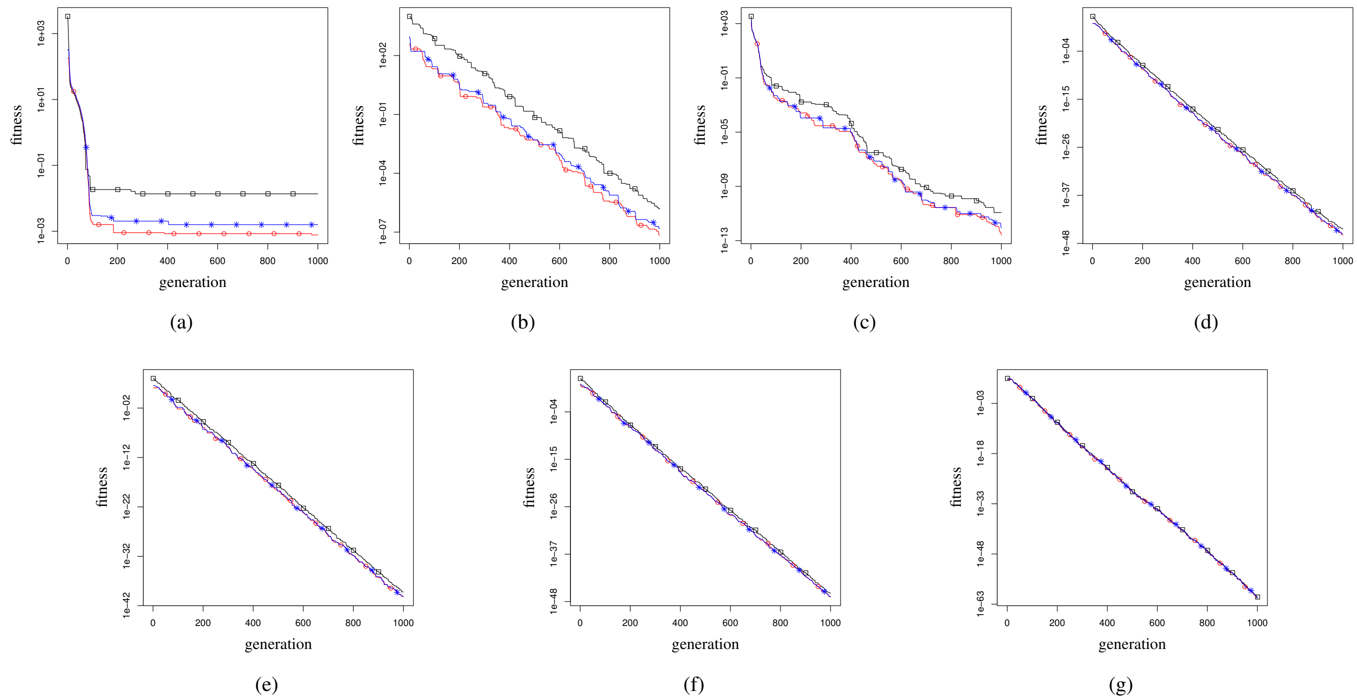

Fig. 2. Convergence curves of the best-so-far solution (squares) and the best-so-far midpoint value for two midpoint definitions: arithmetic mean (circles) and the median (stars), for GEA (a), DE (b), ES (c), JADE (d), b6e6rl (e), SADE (f) and CMA-ES (g); note different scales for the y-axis

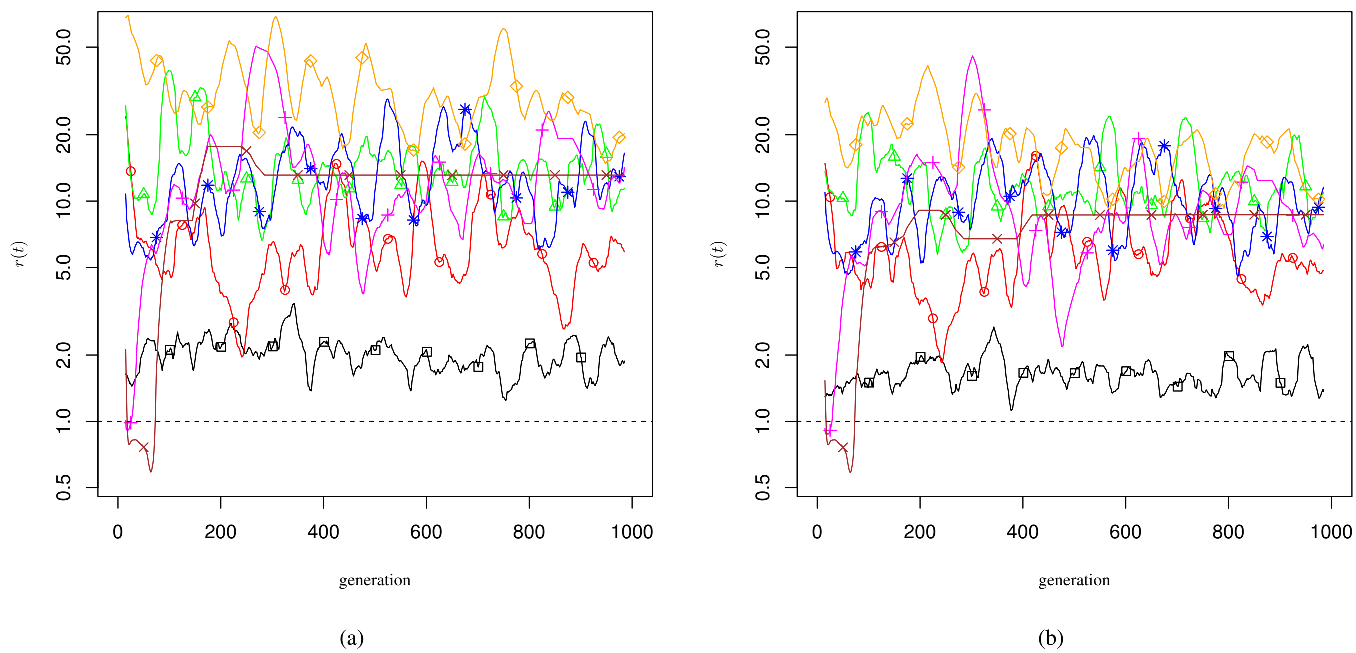

Fig. 3. Ratio of the fitness of the best-so-far individual to the best-so-far mean point (a) and median point (b) for CMA-ES (squares), SADE (circles), GEA (crosses), JADE (stars), ES (pluses), b6e6rl (triangles) and DE (diamonds)

coordinate values as appropriate limiting values, according to the formula:

$$y_i = \begin{cases} 2l_i - x_i & \text{when } x_i < l_i \\ x_i & \text{when } l_i \le x_i \le u_i \\ 2u_i - x_i & \text{when } x_i > u_i \end{cases} \quad (10)$$

where x_i and y_i denote the i-th coordinate values of the mutant and its feasible substitute, and l_i, u_i denote the lower and upper limiting values for the i-th dimension. The exception for that rule was SADE, where each infeasible point was replaced with a new point which was picked up randomly with uniform distribution in the feasible area. This inconsistency of constraint handling techniques was due to the definition of SADE, where handling constraints by replacement with the uniformly generated random individual was explicitly postulated.

The experiment was organized as follows. For each algorithm, each dimension, each benchmark set and each fitness function, 51 independent pairs of runs were performed. In each run, the number of the fitness function evaluations was limited to $K = 10,000 \cdot d$, according to the benchmarking rules. The first run in a pair was started with a population of N points initialized uniformly in the feasible area. Two statistics were recorded:

1) The best fitness value of all individuals generated in that run, denoted by α_j,
2) The best fitness value of all generated individuals and of all midpoints, denoted by β_j,

where $j = 1...51$ is the number of a pair of runs.

The second run in the pair was performed assuming $N-1$ points in each population. For CMA-ES we assumed that $N-1$ points are generated in each iteration, and for ES we set $\lambda = 99$. The initial population was equal to the initial population from the first run in the pair, except for the last individual. We again recorded the best fitness value of all individuals and all midpoints, denoted by γ_j.

For each benchmark set, each function, each dimension and each algorithm, the set of $A = \{\alpha_1, ..., \alpha_{51}\}$ was compared with the set $B = \{\beta_1, ..., \beta_{51}\}$, and the statistical significance of the difference was checked using the Wilcoxon test in pairs. We also made a similar comparison between set A and $G = \{\gamma_1, ..., \gamma_{51}\}$. For each algorithm and each dimension, the number of fitness functions was counted when set B was superior to A, which reflects the case when the midpoint fitness is computed with zero cost. Similarly, the number of fitness functions when set G was superior or inferior to set A shows the consequences of computing the midpoint in each generation at a unit cost. The results of the comparison are summarized in Table I.

b) Results of experiments: Comparison of quality gain obtained using the arithmetic mean and the median point does not show any clear difference. For this reason the arithmetic mean is more advisable, since it can be computed faster.

When the cost of evaluating the fitness of midpoints was zero, it was a rule that computing midpoints improved the quality of results for some fitness functions. The improvement in quality depended on the algorithm and on the problem dimension. For simple algorithms (GEA, DE, ES) and for CMA-ES the improvement rate was high, for b6e6rl it was moderate, and for SADE and JADE — almost negligible. There is no clear pattern of influence of the problem dimensionality on the improvement rate. Improvement was more often observed for the CEC2013 benchmark set.

When the midpoint was evaluated at the expense of one individual from the population, the improvement rate decreased in comparison to the zero-cost case. It even happened that for some fitness functions, results became worse in the statistically significant case. When looking at particular algorithms, for GEA, ES, CMA-ES and DE , improvement was observed more frequently than deterioration. JADE, SADE and b6e6rl were different again — in their case, the number of fitness functions with improvement was small and not very different from the number of fitness functions with deterioration. Moreover, it happened that the number of deteriorations exceeded the number of improvements.

TABLE I
NUMBER OF FITNESS FUNCTIONS WHEN INTRODUCING THE MIDPOINT IMPROVED ('+') OR DETERIORATED ('–') THE OVERALL RESULT WITH STATISTICAL SIGNIFICANCE ACCORDING TO THE WILCOXON TEST IN PAIRS. TWO CASES WERE CONSIDERED: POPULATION SIZE N AND REDUCED POPULATION SIZE $N-1$

Results for CEC2005

		10D			30D			50D		
alg.	midpoint	N	$N-1$		N	$N-1$		N	$N-1$	
type	def.	+	+	–	+	+	–	+	+	–
GEA	mean	13	4	0	14	8	0	15	7	0
	median	10	3	0	11	6	0	15	7	0
ES	mean	14	6	2	19	13	0	20	16	0
	median	15	5	2	18	14	0	20	16	0
CMAES	mean	18	4	0	6	3	1	23	8	0
	median	18	3	1	4	2	1	21	4	0
DE	mean	10	9	0	10	9	0	8	6	2
	median	13	8	0	14	11	0	10	9	2
JADE	mean	3	2	0	5	3	1	2	0	4
	median	1	1	0	0	1	1	0	0	4
SADE	mean	5	2	1	3	6	4	1	0	0
	median	1	2	1	0	3	4	1	0	0
b6e6rl	mean	9	6	1	8	3	1	6	1	2
	median	5	4	1	4	2	1	6	2	3

Results for CEC2013

		10D			30D			50D		
alg.	midpoint	N	$N-1$		N	$N-1$		N	$N-1$	
type	def.	+	+	–	+	+	–	+	+	–
GEA	mean	20	4	0	21	5	4	23	14	1
	median	20	4	0	21	5	4	23	14	1
ES	mean	26	7	1	27	12	0	27	16	0
	median	23	4	1	26	11	1	26	16	0
CMAES	mean	21	5	1	9	3	1	21	8	4
	median	21	4	4	7	3	1	20	7	5
DE	mean	19	13	1	17	13	1	17	10	0
	median	21	15	1	22	15	1	22	13	0
JADE	mean	4	3	0	4	1	1	2	1	1
	median	4	3	0	2	2	1	1	2	0
SADE	mean	3	4	0	2	0	1	2	2	0
	median	2	3	0	2	0	1	2	3	0
b6e6rl	mean	5	3	1	8	2	1	10	4	0
	median	5	3	2	7	3	2	10	5	0

c) Comments: The results show that for a variety of EAs, monitoring population midpoints is an effective way to improve the quality of results. Even if the midpoint is computed at every generation at the expense of one individual, an improvement of overall results can be expected more often than deterioration and the price of additional evaluations of midpoints is worth paying.

In the case of SADE, JADE and b6e6rl, computing midpoints did not achieve an impressive improvement, and in the scenario with $N-1$ points in populations it even yielded a deterioration of results for few benchmark problems. Interpretation of this phenomenon would need much deeper insight into the dynamics of populations processed by the analyzed methods and will go far beyond the scope of the paper. We can only conjecture that this effect can be explained in the following way.

Improvement of results due to the midpoint monitoring can be expected when the population is a group of individuals concentrated around a certain midpoint. In practice, individuals will form clusters, whose number and location will vary over time. Then the population midpoint will not match the center-point of any cluster. Since sustainable clusters will be rather located nearby local optima, the midpoint of the population with clusters will usually mismatch any local optimum.

The tendency to form clusters and the number of generations they may last in populations depends on the algorithm and its parameters. In CMA-ES, such clusters would not last more than one generation, since in every generation all individuals will be generated from the multivariate normal distribution. In GEA or ES, clusters are continuously formed since one individual may be multiply selected, but in the same time there is a tendency to reduce the number of clusters since every individual competes with all others to reproduce. In algorithms from the Differential Evolution family, each individual from the parent population competes with exactly one offspring individual. Therefore, it may be kept in its old position for many generations until it can be substituted by a better offspring. Consequently, clusters are expected to last relatively long.

The plain DE differs significantly from SADE, JADE and b6e6rl in the improvement due to the midpoint monitoring. We believe that this can be explained in the light of the convergence curves from Fig. 2. DE converges very slowly, whereas SADE, JADE and b6e6rl are much faster. Therefore, the quality of results for the plain DE is poor enough to be improved by the midpoint.

V. Conclusions

This paper provided analytical reasons for evaluating the fitness function for the population midpoint. It empirically showed that the effectiveness of population based EAs in $\mathbb{R}^d$ can be improved with this addition. In the experiment using fitness functions from the CEC2005 and CEC2013 benchmark sets, a significant improvement was observed for several types of EAs. Two exceptions to this rule were observed, possibly due to a formation of clusters introduced by specific schemes of differential mutation. Deeper insight into this effect is planned for further research.

In this paper, we studied extreme cases — when the fitness of the midpoint was evaluated cost-free and when it cost one individual. In practical application, there would be no need to evaluate the fitness of the midpoint in every generation. Instead, one can imagine less costly strategies, e.g. computing the midpoint in every k-th generation, computing the midpoint after stagnation of the best-so-far fitness value, or performing clustering and then computing the midpoints of clusters. We believe that these issues define the next steps of research to be taken.

References

[1] X. Qi and F. Palmieri, “Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space part I: Basic properties of selection and mutation,” *IEEE Trans. Neural Netw.*, vol. 5, no. 1, pp. 102–119, Jan. 1994.

[2] I. Karcz-Dulęba, “Dynamics of infinite populations evolving in a landscape of uni and bimodal fitness functions,” *IEEE Trans. Evol. Comput.*, vol. 5, no. 4, pp. 398–409, Aug. 2001.

[3] J. Arabas, “Approximating the genetic diversity of populations in the quasi-equilibrium state,” *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 632–644, Oct. 2012.

[4] S. Ghosh, S. Das, A. V. Vasilakos, and K. Suresh, “On convergence of Differential Evolution over a class of continuous functions with unique global optimum,” *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 42, no. 1, pp. 107–124, Feb. 2012.

[5] J. Arabas, O. Maitre, and P. Collet, “PARADE: a massively parallel differential evolution template for EASEA,” in *Swarm and Evolutionary Computation*. Springer, 2012, pp. 12–20.

[6] P. N. Suganthan *et al.*, “Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,” Nanyang Technological University, Singapore, and KanGAL, IIT Kanpur, India, Tech. Rep., May 2005.

[7] J. Liang, B. Qu, P. Suganthan, and A. G. Hernández-Díaz, “Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization,” Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Tech. Rep., 2013.

[8] N. Hansen and A. Ostermeier, “Completely derandomized self-adaptation in evolution strategies,” *Evol. Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[9] J. Zhang and A. C. Sanderson, “JADE: Adaptive differential evolution with optional external archive,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.

[10] A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

[11] J. Tvrdík, “Adaptation in differential evolution: A numerical comparison,” *Appl. Soft Comput.*, vol. 9, no. 3, pp. 1149–1155, Jun. 2009.

[12] S. García, D. Molina, M. Lozano, and H. Francisco, “A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization,” *J. of Heuristics*, vol. 15, no. 6, p. 617, 2008.

[13] C. Walck, “Hand-book on statistical distributions for experimentalists,” Particle Physics Group, Fysikum, University of Stockholm, Sweden, Tech. Rep., Sep. 2007.

[14] H. Trautmann, O. Mersmann, and D. Arnu, *cmaes: Covariance Matrix Adapting Evolutionary Strategy*, 2011, R package version 1.0-11. [Online]. Available: https://cran.r-project.org/package=cmaes

[15] M. Spruzina, *b6e6rl: Adaptive differential evolution, b6e6rl variant*, 2013, R package version 1.1. [Online]. Available: https://cran.r-project.org/package=b6e6rl