A Version of IPOP-CMA-ES Algorithm with Midpoint for CEC 2017 Single Objective Bound Constrained Problems

Rafał Biedrzycki

Institute of Computer Science
Warsaw University of Technology
email: rbiedrzy@elka.pw.edu.pl

***Abstract*—This paper presents the RB-IPOP-CMA-ES algorithm which is an enhanced version of IPOP-CMA-ES. The algorithm uses midpoint of the population as an approximation of the optimum. The midpoint fitness is also used to introduce a new restart trigger for IPOP. Other IPOP restart triggers and parameters are also corrected.**

The performance of the proposed approach is evaluated on 30 problems from the CEC 2017 benchmark for 10, 30, 50 and 100 dimensions. The results confirm that RB-IPOP-CMA-ES achieves better results than its version that does not utilize midpoint and is a considerable improvement over a plain IPOP-CMA-ES.

I. Introduction

The CMA-ES [1] algorithm has become a standard for real parameter optimization. Since it is a very robust but local optimizer [1], it could stop its operation in any of local optima. To improve CMA-ES results on multimodal problems, several restart strategies have been proposed. One of them is IPOP-CMA-ES [2] which involves restarts with increased population size when stagnation is detected. This simple approach won the CEC 2005 competition [3]. An implementation of IPOP-CMA-ES is provided by the *cmaesr* package [4] in the R language [5]. Its code was used as a base to develop the algorithm presented in this paper and it was also used as a reference method.

Every evolutionary optimization process consists of two phases: exploration, when the algorithm explores new regions of the search space, and exploitation, when the algorithm is trying to locate a (possibly local) optimum as accurately as possible. In evolutionary algorithms the exploitation phase can be sped-up by computing the population midpoint [6]. The midpoint better estimates an optimum than the best-so-far solution when the population has settled down in an attraction basin of a local optimum. It can be easily seen and theoretically proven for quadratic functions but it was also shown in [7] that computing and evaluating the midpoint is a cost worth paying for a wider class of functions. The published experiments showed that even for difficult problems taken from CEC 2005 and CEC 2013 [8] the improvement achieved by computing the midpoint is easily noticeable for a generational evolutionary algorithm, an evolutionary strategy, classical differential evolution and CMA-ES.

The remainder of this paper is organized as follows. Section II describes the IPOP-CMA-ES algorithm. Section III presents all modifications that were introduced to the IPOP-CMA-ES R implementation. The experimental comparison of the IPOP-CMA-ES version introduced in [2] to its R implementation is given in Section IV-A. Section IV-B investigates the impact of changes introduced in RB-IPOP-CMA-ES on optimization results. The full CEC tables are given in Section IV-C. Section V concludes the paper.

II. IPOP-CMA-ES

Since RB-IPOP-CMA-ES is an extension of IPOP-CMA-ES a description of IPOP will be provided in this section. The pseudo-code of IPOP is presented in Fig. 1. It can be observed the algorithm require an initial solution to be optimized. The method consists in iterating CMA-ES and recording the best-so-far solution. The CMA-ES is stopped not only when convergence is observed but also when its parameter values indicate that further proceeding will become ineffective. Before each restart of CMA-ES the population size is increased, and a new initial solution is selected. The best-so-far solution, that has ever been observed in all turns of the loop, is returned as the algorithm result.

Paradoxically in the R implementation of the IPOP-CMA-ES restarts are disabled by default. They could be enabled by setting the *max.restarts* variable to a value greater than zero and setting the *restart.triggers* variable to a list of stopping criteria names which instead of terminating optimization should result in a restart. The default stopping criteria are as follows:

- tolX – the standard deviation of the normal distribution is below tolerance in all coordinates, with default tolerance set to 1e-12,
- noEffectAxis – addition of $0.1 * \sigma$ in a principal axis direction of the covariance matrix (C) does not change the mutation distribution mean value (m),

978-1-5090-4601-0/17/$31.00 ©2017 IEEE

```
Require: initial solution
  while not stop do
    run CMA-ES with modified stop criterion
    increase population size and sample an initial solution
  end while
  return best-so-far
```

Fig. 1. Pseudo-code of IPOP-CMA-ES.

- noEffectCoord – addition of $0.2 * \sigma$ in any coordinate does not change m,
- conditionCov – condition number of the covariance matrix exceeds a tolerance value, with default tolerance set to 1e14,
- indefCovMat – the covariance matrix is not numerically positive definite.

Other stopping criteria are also available:

- stopOnOptValue – stop after reaching the optimal fitness value with a given tolerance,
- stopOnMaxEvals – stop after a specified number of objective function evaluations,
- stopOnMaxIters – stop after a specified number of iterations,
- stopOnOptParam – stop after finding the optimal parameters with a given tolerance,
- stopOnTimeBudget – stop if maximal running-time budget was reached.

As it was described in [2], in IPOP-CMA-ES a stagnation of the algorithm is detected by examining the best fitness value of the *LastIts* search iterations, where *LastIts* is defined as follows:

$$LastIts = 10 + \lceil 30n/\lambda \rceil \tag{1}$$

If the range of the best objective function values of the last *LastIts* iterations is zero, or the range of these function values and all function values of the recent generation is below 10^{-12} then a restart is triggered. This restart strategy is not implemented in the R version of the algorithm.

In the IPOP-CMA-ES R implementation when a restart is triggered, a new starting point is set to the mean (m) of the search distribution that was calculated just before the restart. In the original version of the algorithm [2] a new starting point is randomly drawn from a uniform distribution in the admissible area.

In the IPOP-CMA-ES R implementation all CMA-ES-related parameters are set as in plain CMA-ES, i.e., the offspring population size λ is set to $4 + \lfloor 3 * \log(n) \rfloor$, where n is problem dimensionality, the population size μ is set to $\lfloor \lambda/2 \rfloor$, and the initial step size σ is set to 0.5.

III. Implementation and tuning of RB-IPOP-CMA-ES

The RB-IPOP-CMA-ES is based on the freely available R implementation of the IPOP-CMA-ES [4]. The pseudo-code of the presented approach is identical to the pseudo-code of IPOP that was presented in Fig.1. All aforementioned default stopping criteria of the R implementation of the IPOP-CMA-ES were used as restart triggers. The maximum number of restarts was set to 100.

According to the CEC2017 rules [11] the following stopping criteria were used:

- stopOnOptValue – stop after reaching the optimal value with tolerance 1e-8,
- stopOnMaxEvals – stop after a specified number of objective function evaluations.

A. Parameter tuning

The most straightforward way to improve an algorithm's performance is to tune its parameters. Two main CMA-ES parameters were changed. One is the default offspring population size λ, which was multiplied by 4, and the other is the initial value of σ (initial step-size), which was set to 7. The intuition behind increasing λ is that a very small population size is not able to correctly sample landscapes of sophisticated benchmark functions. The change of σ was triggered by inconsistency of its default value and values used in IPOP-related publications, e.g., it was set to 100 in [9], or even to 120 in [10]. Each introduced change underwent limited verification in 10 dimensions on functions 5 and 6 from the CEC 2017 [11] benchmark.

B. Starting point

CMA-ES requires providing a starting point. Selecting randomly such a point may result in a start in a very poor area of the search space. Therefore in this work the best from 100 randomly generated solutions is used as a starting point.

C. IPOP modifications

To achieve further improvements, some modifications of the algorithm were introduced. They can be divided into two groups: 1) to force algorithm to restart as early as possible to avoid useless evaluations of the fitness function; 2) to repair parameters which went out of reasonable bounds after adaptation performed during evolution.

In the considered R implementation of IPOP-CMA-ES it is assumed that if any eigen vector value of the covariance matrix (C) is less than the first eigen vector value multiplied by 2.22e-16 then a restart is performed. In the proposed approach all values below 1e-15 are set to 1e-15. A restart is performed when the maximum eigen vector value exceeds the minimal eigen value multiplied by 1e14.

In the default restart criterion *stopOnNoEffectAxis* a restart is performed if addition of 0.1 multiplied by σ does not change the mean value (m). In the proposed modified version of the algorithm 0.1 was changed to 0.3.

In the IPOP-CMA-ES R implementation when a restart is triggered, a new starting point is set to the mean (m) of the search distribution that was calculated just before the restart. The algorithm was changed to restart with a new solution randomly drawn from a uniform distribution in the admissible area as it was described in [2].

Another modification of the algorithm was introduced to reduce mutation strength, i.e., when any value from the $\sigma * \sqrt{diag(C)}$ vector exceeds $Max = (u - l)/5$ then σ is set to $\min(Max/\sqrt{diag(C)})/4$, where u and l are the box constraints of the search space. If in the current run of the algorithm the number of σ corrections is greater than 5, a restart is performed.

Whenever a flat fitness landscape is detected in the IPOP-CMA-ES algorithm, the value of σ is increased. In the presented approach the corrected σ is additionally multiplied by 2 to speed up the escape from the current region of the search space.

D. *Utilizing population midpoint*

In the case of CMA-ES the midpoint does not have to be calculated from the population because the population is generated from one point (m), which is the mean of the search distribution. Therefore in the presented algorithm m is directly used as a midpoint.

The midpoint is used for two purposes: 1) to detect stagnation of the algorithm; 2) to approximate the optimum.

In the presented approach the stagnation detection is not based on examining the best fitness value of the *LastIts* search iterations as in IPOP. Here it is based on the fitness of the midpoint. Every *LastIts* iterations of the algorithm the quality of the midpoint is evaluated. If the difference between two consecutive evaluations is less than 1e-8 then a restart is triggered.

In the CEC 2017 competition the task of participants is to report the error value achieved after a specified fraction of the maximum available budget. The list of 14 fractions is as follows: 0.01, 0.02, 0.03, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. In the proposed approach before generating every report the midpoint is evaluated and compared with the best-so-far solution. The better of the two is considered as the result of the algorithm. Of course the cost of 14 evaluations is subtracted from the budget available for the algorithm.

IV. Experimental validation

The experimental validation of the described method was performed according to the rules of the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, bound constrained case [11]. There is no official version of the benchmark in the R language but a wrapper to the official C code has been made available [12].

All 30 functions defined in the benchmark were minimized. For each considered dimension (D), 51 independent runs of the algorithm were executed. Each run was terminated after $10000 * D$ fitness evaluations. The cost of all evaluations used to select the starting point and to asses the fitness of the midpoints is subtracted from the budget available for CMA-ES.

A. *Comparison of IPOP-CMA-ES to its R implementation*

As described in section III, the R implementation of IPOP-CMA-ES does not exactly follow the description provided in [2]. Therefore the results of the R implementation of IPOP-CMA-ES will be called IPOP-CMAESR within this paper.

TABLE I
Average fitness, standard deviations and paired Wilcoxon test outcomes in 10D (reference=IPOP-CMAESR)

	IPOP-CMAESR		IPOP-CMA-ES		
F.	Mean	Std	Mean	Std	T
1	1.0847e+02	7.7463e+02	2.5035e+08	5.6755e+08	+
2	0.0000e+00	0.0000e+00	8.1556e+09	4.3847e+10	+
3	1.4271e−08	5.3663e−09	9.0450e+03	2.1127e+04	=
4	1.5183e−08	4.7717e−09	1.4423e−08	5.5710e−09	=
5	6.5153e+01	2.4114e+01	6.4636e+01	2.3804e+01	=
6	5.0469e+01	1.1180e+01	5.0838e+01	1.0730e+01	+
7	3.1167e+02	7.1574e+01	3.1372e+02	7.2287e+01	=
8	5.9689e+01	1.9943e+01	5.9701e+01	1.9987e+01	+
9	1.2940e+03	5.7372e+02	1.3017e+03	5.7659e+02	+
10	1.3745e+03	3.7569e+02	1.3708e+03	3.6843e+02	+
11	7.1603e+01	3.6178e+01	7.6368e+01	5.8374e+01	=
12	6.8545e+02	5.1423e+02	1.1316e+07	2.1261e+07	+
13	9.2730e+02	4.3981e+03	2.3467e+04	2.6452e+04	+
14	9.0421e+01	6.4653e+01	4.6364e+03	5.5081e+03	+
15	1.0066e+02	6.0873e+01	4.1247e+04	7.3777e+04	+
16	4.9515e+02	1.8285e+02	5.0312e+02	1.8166e+02	+
17	2.4886e+02	1.4636e+02	2.4865e+02	1.4734e+02	=
18	3.7562e+03	9.5736e+03	1.2639e+06	4.1164e+06	+
19	5.5076e+01	3.9701e+01	1.3307e+05	7.6558e+05	+
20	2.3372e+02	1.0322e+02	2.3500e+02	1.0048e+02	+
21	2.3321e+02	6.0428e+01	2.3276e+02	6.0241e+01	=
22	1.2006e+02	1.4194e+02	1.2103e+02	1.4791e+02	=
23	4.1745e+02	5.3291e+01	4.2041e+02	5.4951e+01	=
24	4.1356e+02	1.1122e+02	4.2163e+02	1.1402e+02	=
25	4.2098e+02	5.1808e+01	4.2057e+02	5.1009e+01	=
26	6.6121e+02	5.9627e+02	6.1373e+02	5.7441e+02	=
27	4.6323e+02	5.5548e+01	4.4858e+02	4.9957e+01	−
28	4.8007e+02	1.5957e+02	4.7723e+02	1.5708e+02	+
29	5.3868e+02	1.8035e+02	5.4491e+02	1.8069e+02	=
30	9.4061e+06	1.6644e+07	1.6131e+07	2.2653e+07	+

The two important nonconformities of IPOP-CMAESR have been detected: 1) IPOP-CMA-ES has an additional restart strategy in which stagnation of the algorithm is detected by examining the best solution fitness dynamics; 2) when a restart is triggered in IPOP-CMA-ES, a new starting point is randomly drawn from uniform distribution unlike in IPOP-CMAESR, where it is set to mean of the search distribution.

It is interesting to find out whether changes introduced in the R implementation improve or deteriorate the performance of the algorithm. To investigate that, an experiment was performed with 51 independent runs of the original and R version of the algorithm with the same set of starting points. The obtained mean and standard deviation values are reported in Table I. The results were also compared by the paired Wilcoxon test with a confidence level of 0.95.

It can be observed that IPOP-CMAESR is generally better than IPOP-CMA-ES. For that reason the IPOP-CMAESR will be used as a reference method in the remainder of the paper.

The inferiority of IPOP-CMA-ES may result from its stagnation detection strategy that is based on the analysis of the best solution dynamics. That strategy invokes restarts too early

TABLE II
AVERAGE FITNESS, STANDARD DEVIATIONS AND PAIRED WILCOXON TEST OUTCOMES IN 10D (REFERENCE=RB-IPOP)

	RB-IPOP		No midpoint			IPOP-CMAESR		
F.	Mean	Std	Mean	Std	T	Mean	Std	T
1	0.0e+00	0.0e+00	0.0e+00	0.0e+00	=	1.1e+02	7.7e+02	+
2	0.0e+00	0.0e+00	0.0e+00	0.0e+00	=	0.0e+00	0.0e+00	=
3	0.0e+00	0.0e+00	0.0e+00	0.0e+00	=	1.4e−08	5.4e−09	+
4	0.0e+00	0.0e+00	0.0e+00	0.0e+00	=	1.5e−08	4.8e−09	+
5	1.6e+00	2.0e+00	2.2e+00	2.9e+00	=	6.5e+01	2.4e+01	+
6	2.0e−07	6.2e−07	3.3e−07	1.6e−06	+	5.0e+01	1.1e+01	+
7	1.0e+01	2.7e+00	1.0e+01	1.9e+00	=	3.1e+02	7.2e+01	+
8	2.0e+00	2.3e+00	1.7e+00	2.0e+00	=	6.0e+01	2.0e+01	+
9	0.0e+00	0.0e+00	0.0e+00	0.0e+00	=	1.3e+03	5.7e+02	+
10	4.4e+02	1.9e+02	4.2e+02	1.9e+02	=	1.4e+03	3.8e+02	+
11	1.7e−01	4.1e−01	9.1e−02	2.8e−01	=	7.2e+01	3.6e+01	+
12	1.1e+02	9.4e+01	1.1e+02	8.6e+01	=	6.9e+02	5.1e+02	+
13	4.2e+00	3.6e+00	3.4e+00	3.1e+00	=	9.3e+02	4.4e+03	+
14	1.6e+01	1.2e+01	1.9e+01	1.1e+01	=	9.0e+01	6.5e+01	+
15	4.9e−01	4.8e−01	5.1e−01	4.5e−01	=	1.0e+02	6.1e+01	+
16	9.7e+01	1.0e+02	8.9e+01	1.1e+02	=	5.0e+02	1.8e+02	+
17	5.2e+01	3.4e+01	5.3e+01	3.4e+01	=	2.5e+02	1.5e+02	+
18	2.0e+01	2.3e+01	2.0e+01	1.7e+01	=	3.8e+03	9.6e+03	+
19	1.8e+00	3.3e+00	1.4e+00	1.2e+00	=	5.5e+01	4.0e+01	+
20	1.1e+02	7.0e+01	1.1e+02	7.4e+01	=	2.3e+02	1.0e+02	+
21	1.4e+02	4.9e+01	2.1e+02	5.4e+01	+	2.3e+02	6.0e+01	+
22	9.9e+01	5.6e+00	1.0e+02	1.2e−01	+	1.2e+02	1.4e+02	+
23	2.8e+02	7.1e+01	3.1e+02	3.1e+01	+	4.2e+02	5.3e+01	+
24	2.0e+02	1.0e+02	3.1e+02	8.5e+01	+	4.1e+02	1.1e+02	+
25	4.0e+02	6.5e+01	3.9e+02	7.9e+01	=	4.2e+02	5.2e+01	+
26	2.7e+02	1.5e+02	2.8e+02	1.5e+02	=	6.6e+02	6.0e+02	+
27	3.9e+02	1.1e+00	4.0e+02	6.5e+00	+	4.6e+02	5.6e+01	+
28	4.0e+02	1.6e+02	4.3e+02	1.7e+02	+	4.8e+02	1.6e+02	+
29	2.7e+02	4.5e+01	2.7e+02	4.6e+01	=	5.4e+02	1.8e+02	+
30	2.0e+03	1.0e+04	2.2e+03	9.2e+03	=	9.4e+06	1.7e+07	+

TABLE III
AVERAGE FITNESS, STANDARD DEVIATIONS AND PAIRED WILCOXON TEST OUTCOMES IN 30D (REFERENCE=RB-IPOP)

	RB-IPOP		No midpoint			IPOP-CMAESR		
F.	Mean	Std	Mean	Std	T	Mean	Std	T
1	3.2e−08	2.8e−08	5.3e−08	1.9e−08	+	6.0e−07	2.3e−06	+
2	0.0e+00	0.0e+00	7.4e+16	5.3e+17	=	1.1e+41	5.0e+41	+
3	0.0e+00	0.0e+00	0.0e+00	0.0e+00	=	1.8e−08	1.8e−09	+
4	5.5e+01	1.6e+01	5.4e+01	1.8e+01	=	4.6e+01	2.7e+01	=
5	1.6e+00	1.4e+00	1.6e+00	1.4e+00	=	3.7e+02	7.2e+01	+
6	1.2e−07	4.0e−08	1.5e−07	2.0e−08	+	7.1e+01	9.0e+00	+
7	3.4e+01	1.3e+00	3.4e+01	1.1e+00	=	2.2e+03	4.9e+02	+
8	1.8e+00	1.6e+00	2.0e+00	1.7e+00	=	3.1e+02	4.9e+01	+
9	0.0e+00	0.0e+00	0.0e+00	0.0e+00	=	8.2e+03	1.8e+03	+
10	1.4e+03	5.8e+02	1.4e+03	5.6e+02	=	4.6e+03	8.0e+02	+
11	4.1e+01	4.8e+01	5.8e+01	5.6e+01	=	1.8e+02	6.1e+01	+
12	1.1e+03	2.8e+02	1.1e+03	3.3e+02	=	5.1e+07	3.6e+08	+
13	1.2e+02	4.0e+02	5.2e+01	1.2e+02	=	3.0e+07	2.2e+08	+
14	9.1e+01	5.6e+01	9.1e+01	4.9e+01	=	1.0e+05	7.2e+05	+
15	2.2e+02	1.8e+02	2.3e+02	1.8e+02	=	1.0e+04	1.7e+04	+
16	5.0e+02	2.5e+02	5.1e+02	2.6e+02	=	1.4e+03	4.7e+02	+
17	1.3e+02	9.5e+01	1.4e+02	9.4e+01	=	1.0e+03	3.0e+02	+
18	1.6e+02	1.1e+02	1.7e+02	8.2e+01	=	1.8e+06	9.4e+06	+
19	1.1e+02	6.6e+01	1.2e+02	6.9e+01	=	4.5e+06	2.2e+07	+
20	3.0e+02	1.2e+02	2.7e+02	1.4e+02	=	8.9e+02	2.3e+02	+
21	2.1e+02	1.7e+01	2.1e+02	1.9e+01	+	5.3e+02	6.1e+01	+
22	6.7e+02	7.6e+02	5.8e+02	7.4e+02	=	5.3e+03	1.2e+03	+
23	3.4e+02	4.9e+01	3.6e+02	1.2e+01	+	1.1e+03	1.4e+02	+
24	4.2e+02	3.1e+00	4.3e+02	1.3e+01	+	7.6e+02	2.9e+02	+
25	3.9e+02	1.5e−02	3.9e+02	1.2e−02	=	3.9e+02	2.4e+00	+
26	3.9e+02	2.2e+02	4.3e+02	2.1e+02	=	5.8e+03	3.3e+03	+
27	5.1e+02	1.1e+01	5.1e+02	9.6e+00	=	6.4e+02	2.7e+02	+
28	3.1e+02	3.0e+01	3.3e+02	4.5e+01	+	3.6e+02	6.8e+01	+
29	4.9e+02	1.0e+02	5.2e+02	8.9e+01	=	1.3e+03	3.5e+02	+
30	2.8e+03	1.9e+03	4.1e+03	5.7e+03	=	2.1e+08	2.6e+08	+

for the most of the CEC 2017 functions, especially for highly conditioned functions.

B. Comparison of RB-IPOP-CMA-ES to its simplified methods

A comparison of the method proposed by this work (RB-IPOP) to its version without the utilization of midpoints (No midpoint) and to the IPOP-CMA-ES R implementation (IPOP-CMAESR) was performed. All algorithms were run on the same set of starting points. The mean, standard deviation, and the outcome of the paired Wilcoxon test (with a confidence level of 0.95) are presented in Table II (10D), III (30D), and IV (50D).

According to the Wilcoxon test (columns 'T' of Tables II, III, IV), the RB-IPOP-CMA-ES was never worse than the other algorithms under comparison. It was better than its version without utilizing midpoint on 6 to 9 functions. Its worth noticing that the number of functions with improvement due to midpoint utilization increased with dimensionality of the problem (from 7 in 10D to 9 in 50D). The RB-IPOP-CMA-ES was better than IPOP-CMAESR on 29 functions in all dimensions.

C. CEC 2017 tables

Following the rules of the CEC 2017 benchmark for single objective bound constrained problems, the results of 51 independent runs of RB-IPOP-CMAES were gathered for problem dimensions 10, 30, 50 and 100. For each test function and each dimension the best, worst, median and average error value and its standard deviation were calculated. These values are reported for 10, 30, 50 and 100 dimensions in Tables V, VI, VII and VIII, respectively.

It can be observed that from the set of unimodal functions (1-3) the first one is the hardest. When looking at the median value it can be seen that functions 2 and 3 were solved in all dimensions. Function 1 was solved in 10 dimensions, and nearly solved in other dimensions.

Considering simple multimodal functions (4-10), function 9 is the easiest. It was solved in all dimensions. Function 4 was solved in 10 dimensions.

From the set of hybrid functions (11-20) only function 11 was solved in 10 dimensions.

None of the composition functions was solved. Function 30 was the hardest to solve.

D. Algorithm complexity

The experiments were performed on a machine equipped with an Intel Xeon CPU E7-4830 v3 @ 2.1 GHz that was working under the control of Linux Debian Jessie. The program was written in the R language and running in a single thread. The complexity of the algorithm was calculated according to [11]. More specifically, $T0$ is the execution time of one million evaluations of some basic mathematical

TABLE IV
AVERAGE FITNESS, STANDARD DEVIATIONS AND PAIRED WILCOXON TEST OUTCOMES IN 50D (REFERENCE=RB-IPOP)

	RB-IPOP		No midpoint			IPOP-CMAESR		
F.	Mean	Std	Mean	Std	T	Mean	Std	T
1	1.1e−07	4.3e−08	1.1e−07	3.6e−08	=	3.9e+02	1.9e+03	+
2	2.8e+05	2.0e+06	1.8e+08	1.3e+09	=	6.4e+82	3.9e+83	+
3	0.0e+00	0.0e+00	0.0e+00	0.0e+00	=	1.9e−08	1.1e−09	+
4	3.0e+01	4.1e+01	4.3e+01	4.4e+01	=	4.0e+01	4.6e+01	=
5	2.8e+00	1.4e+00	2.7e+00	1.7e+00	=	6.9e+02	8.5e+01	+
6	1.6e−07	1.4e−07	1.6e−07	2.1e−08	+	7.7e+01	6.5e+00	+
7	5.7e+01	1.4e+00	5.7e+01	1.7e+00	+	4.5e+03	4.2e+02	+
8	2.6e+00	1.8e+00	2.8e+00	1.6e+00	=	6.9e+02	7.1e+01	+
9	0.0e+00	0.0e+00	0.0e+00	0.0e+00	=	2.1e+04	3.5e+03	+
10	1.7e+03	9.5e+02	1.9e+03	1.0e+03	=	8.2e+03	1.3e+03	+
11	1.8e+02	5.2e+01	2.1e+02	6.3e+01	+	3.0e+02	7.6e+01	+
12	2.4e+06	1.7e+07	2.5e+03	5.2e+02	=	2.0e+10	2.8e+10	+
13	1.7e+03	1.1e+03	1.4e+03	1.2e+03	=	3.6e+07	2.6e+08	+
14	2.4e+02	7.1e+01	2.4e+02	7.4e+01	=	1.1e+06	6.0e+06	+
15	5.3e+02	1.2e+02	5.4e+02	1.6e+02	=	9.5e+03	1.1e+04	+
16	8.9e+02	3.7e+02	9.2e+02	4.2e+02	=	2.2e+03	5.6e+02	+
17	4.0e+02	1.6e+02	4.2e+02	2.1e+02	=	1.4e+03	4.4e+02	+
18	3.6e+02	1.6e+02	3.6e+02	1.6e+02	=	9.7e+06	3.5e+07	+
19	1.4e+02	4.8e+01	1.6e+02	5.8e+01	+	8.9e+06	5.2e+07	+
20	5.5e+02	2.3e+02	6.1e+02	2.8e+02	=	1.9e+03	3.8e+02	+
21	2.1e+02	3.2e+00	2.3e+02	1.9e+01	+	9.3e+02	8.0e+01	+
22	2.1e+03	1.8e+03	1.8e+03	1.2e+03	=	8.8e+03	1.2e+03	+
23	4.2e+02	1.4e+01	4.3e+02	1.2e+01	+	1.8e+03	1.7e+02	+
24	4.9e+02	5.7e+00	4.9e+02	8.6e+00	=	1.0e+03	4.1e+02	+
25	4.8e+02	5.2e+00	4.9e+02	2.3e+01	+	5.1e+02	3.6e+01	+
26	6.5e+02	3.0e+02	7.7e+02	2.5e+02	+	6.0e+03	4.9e+03	+
27	6.1e+02	5.9e+01	6.3e+02	5.4e+01	+	1.4e+03	9.4e+02	+
28	4.7e+02	1.9e+01	4.8e+02	2.4e+01	=	4.9e+02	2.3e+01	+
29	6.7e+02	2.0e+02	6.5e+02	2.0e+02	=	1.7e+03	7.1e+02	+
30	6.5e+06	5.1e+06	6.3e+06	3.5e+06	=	1.6e+09	2.2e+09	+

TABLE V
BEST, WORST, MEDIAN, AVERAGE, AND STANDARD DEVIATION FOR 51 INDEPENDENT RUNS ON CEC 2017 PROBLEMS FOR DIMENSION 10

F.	Best	Worst	Median	Mean	Std
1	0.0000e+00	1.4935e−08	0.0000e+00	0.0000e+00	2.0913e−09
2	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
3	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
4	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
5	0.0000e+00	9.9496e+00	9.9496e−01	1.5802e+00	1.9616e+00
6	1.8682e−08	3.9800e−06	7.9057e−08	2.0103e−07	6.2085e−07
7	6.4345e−07	1.2216e+01	1.0815e+01	1.0110e+01	2.6905e+00
8	0.0000e+00	1.1939e+01	9.9496e−01	1.9704e+00	2.3248e+00
9	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
10	3.7473e−01	1.0308e+03	4.0726e+02	4.3543e+02	1.9035e+02
11	0.0000e+00	1.9899e+00	0.0000e+00	1.7118e−01	4.1048e−01
12	2.0814e−01	4.6092e+02	1.1995e+02	1.1046e+02	9.3745e+01
13	0.0000e+00	1.6291e+01	4.8371e+00	4.1702e+00	3.6141e+00
14	1.1805e−07	5.8803e+01	2.0008e+01	1.5933e+01	1.2492e+01
15	6.5063e−04	2.3778e+00	4.9999e−01	4.9090e−01	4.7594e−01
16	6.5509e−01	3.5886e+02	1.1914e+02	9.7098e+01	1.0294e+02
17	6.6927e+00	1.6403e+02	4.1752e+01	5.2461e+01	3.3580e+01
18	2.6422e−02	1.4434e+02	2.0511e+01	1.9722e+01	2.3494e+01
19	1.9432e−02	2.3615e+01	1.2091e+00	1.8158e+00	3.2994e+00
20	1.6193e+00	2.6380e+02	1.3847e+02	1.0605e+02	6.9525e+01
21	1.0000e+02	2.0906e+02	1.0000e+02	1.3735e+02	4.9239e+01
22	6.0274e+01	1.0040e+02	1.0000e+02	9.9259e+01	5.5693e+00
23	1.0000e+02	3.0916e+02	3.0287e+02	2.7502e+02	7.0550e+01
24	1.6466e−06	3.4517e+02	2.0000e+02	1.9758e+02	1.0164e+02
25	1.0000e+02	4.4338e+02	3.9804e+02	4.0227e+02	6.5374e+01
26	1.1399e−06	1.1802e+03	3.0000e+02	2.7269e+02	1.5110e+02
27	3.9382e+02	3.9863e+02	3.9501e+02	3.9486e+02	1.0888e+00
28	3.0446e−07	6.1182e+02	3.0000e+02	4.0226e+02	1.6374e+02
29	2.2625e+02	4.8150e+02	2.5444e+02	2.6583e+02	4.5310e+01
30	3.9461e+02	7.4708e+04	4.4266e+02	2.0460e+03	1.0390e+04

TABLE VI
BEST, WORST, MEDIAN, AVERAGE, AND STANDARD DEVIATION FOR 51 INDEPENDENT RUNS ON CEC 2017 PROBLEMS FOR DIMENSION 30

F.	Best	Worst	Median	Mean	Std
1	0.0000e+00	1.0572e−07	3.1423e−08	3.1535e−08	2.7534e−08
2	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
3	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
4	0.0000e+00	6.4117e+01	5.8562e+01	5.5276e+01	1.6455e+01
5	2.1074e−08	4.9748e+00	9.9501e−01	1.6486e+00	1.3719e+00
6	3.8387e−08	1.8404e−07	1.3284e−07	1.2079e−07	3.9666e−08
7	3.2707e+01	3.7733e+01	3.4052e+01	3.4334e+01	1.2764e+00
8	0.0000e+00	6.9647e+00	9.9608e−01	1.7564e+00	1.6477e+00
9	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
10	3.8330e+02	2.7665e+03	1.3644e+03	1.4426e+03	5.8271e+02
11	0.0000e+00	1.6644e+02	9.9564e+00	4.1069e+01	4.7633e+01
12	4.4403e+02	1.7377e+03	1.0632e+03	1.0933e+03	2.8097e+02
13	3.9798e+00	2.5382e+03	2.6454e+01	1.1873e+02	3.9972e+02
14	2.0229e+01	2.0039e+02	1.0955e+02	9.0827e+01	5.6177e+01
15	5.0000e−01	6.2613e+02	2.2398e+02	2.1772e+02	1.8406e+02
16	1.7076e+01	1.2366e+03	5.0113e+02	5.0162e+02	2.5411e+02
17	2.3326e+01	4.1301e+02	1.0280e+02	1.3226e+02	9.5401e+01
18	2.2484e+01	5.9360e+02	1.2600e+02	1.6005e+02	1.1400e+02
19	7.8457e+00	2.9443e+02	1.0392e+02	1.1455e+02	6.5932e+01
20	4.1928e+01	5.1036e+02	2.8532e+02	2.9653e+02	1.1908e+02
21	1.0000e+02	2.2341e+02	2.1045e+02	2.0858e+02	1.6676e+01
22	1.0000e+02	2.5115e+03	1.0000e+02	6.7163e+02	7.6258e+02
23	1.0000e+02	3.6627e+02	3.4658e+02	3.3935e+02	4.9294e+01
24	4.1146e+02	4.2481e+02	4.1989e+02	4.1938e+02	3.0639e+00
25	3.8669e+02	3.8676e+02	3.8671e+02	3.8671e+02	1.4745e−02
26	2.0000e+02	8.4792e+02	3.0000e+02	3.9386e+02	2.1681e+02
27	4.8971e+02	5.3947e+02	5.1164e+02	5.1178e+02	1.1318e+01
28	3.0000e+02	4.1398e+02	3.0000e+02	3.0893e+02	2.9560e+01
29	3.7031e+02	9.6428e+02	4.6622e+02	4.9332e+02	1.0360e+02
30	1.9503e+03	1.4306e+04	2.2860e+03	2.8438e+03	1.9413e+03

TABLE VII
BEST, WORST, MEDIAN, AVERAGE, AND STANDARD DEVIATION FOR 51 INDEPENDENT RUNS ON CEC 2017 PROBLEMS FOR DIMENSION 50

F.	Best	Worst	Median	Mean	Std
1	3.2919e−08	2.3301e−07	1.0505e−07	1.1306e−07	4.2646e−08
2	0.0000e+00	1.4114e+07	0.0000e+00	2.7675e+05	1.9764e+06
3	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
4	0.0000e+00	1.2573e+02	1.2036e+01	2.9587e+01	4.0680e+01
5	0.0000e+00	6.9647e+00	2.9849e+00	2.7898e+00	1.4352e+00
6	5.2292e−08	1.0909e−06	1.5047e−07	1.6319e−07	1.3794e−07
7	5.4304e+01	6.0908e+01	5.6511e+01	5.6631e+01	1.3900e+00
8	0.0000e+00	6.9647e+00	1.9899e+00	2.5752e+00	1.7929e+00
9	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
10	1.2639e+02	4.1803e+03	1.7272e+03	1.7262e+03	9.5307e+02
11	6.6141e+01	2.9978e+02	1.8549e+02	1.8324e+02	5.2028e+01
12	1.6142e+03	1.2422e+08	2.3739e+03	2.4381e+06	1.7394e+07
13	6.3405e+01	3.7377e+03	1.9913e+03	1.6518e+03	1.1454e+03
14	2.1840e+01	4.3624e+02	2.4486e+02	2.4162e+02	7.0739e+01
15	3.4950e+02	7.8741e+02	5.2693e+02	5.2901e+02	1.1525e+02
16	1.4859e+02	1.8037e+03	8.6909e+02	8.8992e+02	3.6631e+02
17	8.6526e+01	8.7743e+02	3.7960e+02	3.9780e+02	1.5827e+02
18	6.3134e+01	7.1648e+02	3.4772e+02	3.5735e+02	1.5580e+02
19	7.4424e+01	3.2139e+02	1.2475e+02	1.3935e+02	4.7737e+01
20	7.7765e+01	1.2599e+03	5.3862e+02	5.4683e+02	2.3306e+02
21	2.0000e+02	2.1766e+02	2.0559e+02	2.0619e+02	3.2302e+00
22	1.0000e+02	7.1608e+03	1.5584e+03	2.0512e+03	1.7615e+03
23	3.9645e+02	4.6619e+02	4.2054e+02	4.2269e+02	1.3915e+01
24	4.8292e+02	5.1022e+02	4.8972e+02	4.9116e+02	5.7300e+00
25	4.5828e+02	4.9198e+02	4.8024e+02	4.8070e+02	5.1849e+00
26	3.0000e+02	1.1885e+03	5.8541e+02	6.5482e+02	3.0146e+02
27	5.2678e+02	8.1430e+02	5.8721e+02	6.0807e+02	5.8628e+01
28	4.5885e+02	5.0769e+02	4.5885e+02	4.7000e+02	1.9362e+01
29	3.8642e+02	1.1983e+03	6.4562e+02	6.6896e+02	1.9917e+02
30	1.2222e+06	2.6593e+07	4.5093e+06	6.4600e+06	5.0714e+06

TABLE VIII

BEST, WORST, MEDIAN, AVERAGE, AND STANDARD DEVIATION FOR 51 INDEPENDENT RUNS ON CEC 2017 PROBLEMS FOR DIMENSION 100

F.	Best	Worst	Median	Mean	Std
1	$1.8302e-07$	$4.0782e-07$	$2.6482e-07$	$2.7587e-07$	$5.8809e-08$
2	$0.0000e+00$	$0.0000e+00$	$0.0000e+00$	$0.0000e+00$	$0.0000e+00$
3	$0.0000e+00$	$0.0000e+00$	$0.0000e+00$	$0.0000e+00$	$0.0000e+00$
4	$0.0000e+00$	$2.1583e+02$	$2.0074e+02$	$1.5599e+02$	$8.8599e+01$
5	$9.9498e-01$	$2.9849e+01$	$7.9597e+00$	$1.1001e+01$	$7.6741e+00$
6	$3.8019e-08$	$2.0079e-07$	$1.6033e-07$	$1.3307e-07$	$5.3676e-08$
7	$1.0863e+02$	$1.4612e+02$	$1.2183e+02$	$1.2650e+02$	$9.6212e+00$
8	$1.3573e-04$	$3.0844e+01$	$8.9546e+00$	$1.2198e+01$	$9.0777e+00$
9	$0.0000e+00$	$0.0000e+00$	$0.0000e+00$	$0.0000e+00$	$0.0000e+00$
10	$1.3136e+02$	$1.1603e+04$	$5.4197e+03$	$5.5858e+03$	$3.4942e+03$
11	$2.1233e+02$	$1.6378e+03$	$1.0606e+03$	$1.0471e+03$	$2.3321e+02$
12	$3.6924e+03$	$2.6871e+06$	$5.3889e+03$	$5.7900e+04$	$3.7552e+05$
13	$2.2993e+03$	$6.2723e+03$	$4.4422e+03$	$4.5107e+03$	$9.4601e+02$
14	$4.0544e+02$	$7.8970e+02$	$5.8716e+02$	$5.7849e+02$	$8.9984e+01$
15	$2.3410e+02$	$1.0296e+03$	$7.6475e+02$	$7.5638e+02$	$1.4895e+02$
16	$7.9312e+02$	$2.7056e+03$	$1.7456e+03$	$1.6950e+03$	$4.6783e+02$
17	$7.4783e+02$	$2.0307e+03$	$1.3245e+03$	$1.3805e+03$	$2.6901e+02$
18	$1.8577e+02$	$5.6367e+02$	$2.9723e+02$	$3.1596e+02$	$8.3147e+01$
19	$2.7071e+02$	$1.1361e+03$	$5.8672e+02$	$5.9372e+02$	$2.2800e+02$
20	$1.0572e+03$	$2.9430e+03$	$1.8766e+03$	$1.9045e+03$	$3.7526e+02$
21	$2.2728e+02$	$2.5659e+02$	$2.4449e+02$	$2.4502e+02$	$5.8477e+00$
22	$3.4468e+02$	$1.4018e+04$	$6.4820e+03$	$6.3236e+03$	$4.0126e+03$
23	$5.4454e+02$	$7.4577e+02$	$5.6272e+02$	$5.8402e+02$	$5.2869e+01$
24	$8.6734e+02$	$9.2594e+02$	$8.9315e+02$	$8.9323e+02$	$1.4066e+01$
25	$5.7705e+02$	$7.8242e+02$	$6.9797e+02$	$6.7749e+02$	$4.3374e+01$
26	$3.0000e+02$	$4.1014e+03$	$2.9769e+03$	$2.8935e+03$	$5.6801e+02$
27	$6.2435e+02$	$7.2711e+02$	$6.7280e+02$	$6.7285e+02$	$2.2579e+01$
28	$3.0000e+02$	$6.2814e+02$	$5.1875e+02$	$4.9760e+02$	$8.0150e+01$
29	$1.3248e+03$	$3.0219e+03$	$2.0719e+03$	$2.0284e+03$	$4.0100e+02$
30	$4.8924e+03$	$2.3265e+04$	$1.1775e+04$	$1.1909e+04$	$3.7262e+03$

TABLE IX

COMPUTATIONAL COMPLEXITY OF THE ALGORITHM ON INTEL XEON CPU E7-4830 V3 @ 2.1 GHZ. THE PROGRAM WAS WRITTEN IN R LANGUAGE AS A SINGLE THREADED APPLICATION

Dim.	$T0$	$T1$	$\widehat{T2}$	$(\widehat{T2} - T1)/T0$
10	2.1s	1.9s	2.1s	0.1
30	1.8s	1.5s	6.0s	2.5
50	1.8s	1.9s	10.1s	4.6

expressions, $T1$ is the time of 200000 evaluations of function 18, and $\widehat{T2}$ is the mean taken from 5 independent runs of the presented algorithm. Each algorithm run was performed on a budget of 200000 evaluations of function 18. According to the rules of the benchmark, evaluations of function 18 that were used to calculate $T1$ were performed in an identical way as it is in the actually used implementation.

The results of the experiments are presented in Table IX.

It can be observed that the time complexity of the algorithm grows linearly with problem dimensionality.

V. CONCLUSIONS

The enhanced version of IPOP-CMA-ES, called RB-IPOP-CMA-ES, was proposed in this paper. The algorithm was used to solve problems from the CEC 2017 competition on single objective real-parameter numerical optimization. It was empirically shown that the introduced algorithm modifications substantially improved its results. It was also shown that utilizing the midpoint of the population to detect stagnation of the algorithm and as an approximation of the optimum improves the algorithm's performance on several functions in all considered dimensions.

The proposed approach is capable to obtain good solutions in a fast and reliable way.

REFERENCES

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

[2] A. Auger and N. Hansen, "A restart CMA evolution strategy with increasing population size," in *2005 IEEE Congress on Evolutionary Computation*, vol. 2, Sept 2005, pp. 1769–1776 Vol. 2.

[3] N. Hansen, "Compilation of results on the 2005 CEC benchmark function set," Computational Laboratory (CoLab), Institute of Computational Science, ETH Zurich, Tech. Rep., May 2006.

[4] J. Bossek, *cmaesr: Covariance Matrix Adaptation Evolution Strategy*, 2016, R package version 1.0.2.

[5] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2016. [Online]. Available: https://www.R-project.org/

[6] 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.

[7] J. Arabas and R. Biedrzycki, "Improving evolutionary algorithms in a continuous domain by monitoring the population midpoint," *IEEE Trans. Evol. Comput.*, 2017, accepted, DOI:10.1109/TEVC.2017.2673962.

[8] 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.

[9] T. Liao and T. Stützle, "Benchmark results for a simple hybrid algorithm on the CEC 2013 benchmark set for real-parameter optimization," in *2013 IEEE Congress on Evolutionary Computation*, June 2013, pp. 1938–1944.

[10] I. Loshchilov, "CMA-ES with restarts for solving CEC 2013 benchmark problems," in *2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 369–376.

[11] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization," Nanyang Technological University, Singapore and Jordan University of Science and Technology, Jordan and Zhengzhou University, Zhengzhou China, Tech. Rep., Oct 2016.

[12] D. Jagodziński, *Common interface to the 30 CEC-2017 benchmark functions - Special Session and Competition on Single Objective Real-Parameter Numerical Optimization - bound constrained case*, 2016. [Online]. Available: http://staff.elka.pw.edu.pl/~djagodzi/cec2017/cec2017_0.1.5.tar.gz