Noname manuscript No.
(will be inserted by the editor)

On the Adaptation of the Mutation Scale Factor in Differential Evolution

Carlos Segura · Carlos A. Coello Coello · Eduardo Segredo · Coromoto León

Received: date / Accepted: date

Abstract Differential Evolution (DE) is a simple yet effective metaheuristic specially suited for real-parameter optimization. The most advanced DE variants take into account the feedback obtained in the self-optimization process to modify their internal parameters and components dynamically. In recent years, some controversies have arisen regarding the adaptive schemes that incorporate feedback from the search process to guide the adaptation of the mutation scale factor. Some researchers have claimed that no significant benefits are obtained with these kinds of schemes. However, other studies have shown that they are highly effective. In this paper, we show that there is a relationship between the effectiveness of these adaptive schemes and the balance between exploration and exploitation induced by the trial vector generation strategy considered. State-of-the-art adaptive schemes are not useful for the trial vector generation strategies with the highest levels of exploration, which in fact seems to be the reason behind the controversies of recent years.

Keywords Differential evolution · Mutation scale factor · Adaptation · Parameter control

C. Segura
UMI LAFMIA 3175 CNRS, Departamento de Computación, CINVESTAV-IPN, México, D.F. 07300, MEXICO.
E-mail: csegura@ull.es

C. A. Coello Coello
Departamento de Computación (Evolutionary Computation Group), CINVESTAV-IPN, México, D.F. 07300, MEXICO.
E-mail: ccoello@cs.cinvestav.mx

E. Segredo · C. León
Dpto. Estadística, I. O. y Computación, Universidad de La Laguna. Avda. Astrofísico Fco. Sánchez, Edif. Matemáticas, 38271, Santa Cruz de Tenerife, Spain.
E-mail: esegredo@ull.es
E-mail: cleon@ull.es

1 Introduction

Differential Evolution (DE) [12] is a very popular metaheuristic specially suited for real-parameter optimization. Its good performance and reliability has been consistently demonstrated in several optimization contests and special issues [2, 3]. In addition, it has been successfully used to address a vast amount of optimization problems that arise in practical applications [2].

Since its inception, a large amount of research has been conducted around DE. The correct parameterization of DE has been one of the more active research topics [2]. Initially, DE was presented as a robust scheme with a low number of easily tunable parameters [12]. However, subsequent studies showed that DE is very sensitive to the setting of the control parameters [17]. Furthermore, several DE trial vector generation strategies have been proposed since then [9], hampering its parameterization. Some DE variants consider the use of several DE components and parameter values simultaneously to increase robustness. For instance, in [14] three trial vector generation strategies were simultaneously considered. Moreover, some adaptive schemes have been developed [5,15] that take into account the feedback obtained in the optimization process to guide a dynamic DE parameterization.

In the original DE proposal, three parameters must be set by the user: the mutation scale factor (F), the crossover rate (CR) and the population size (NP). In addition, the mutant vector generation strategy and the crossover operator must also be specified. Most up-to-date adaptive DE variants modify several parameters and components simultaneously. These adaptive schemes are usually compared against DE variants that do not consider any adaptation [1,15]. Thus, in most cases the benefits obtained by the adaptation of each parameter or component is not studied individually.

In recent years, some controversies have appeared regarding the suitability of considering feedback to set the value of F. The study in [16] reveals that the adaptation of F does not provide significant benefits, at least with the adaptation mechanisms devised until the appearance of such paper. Since its appearance, however, several researchers have devised other adaptive schemes that use feedback to adapt the value of F [15,5]. These novel schemes have yielded significant improvements for several problems. Among them, JADE [15] has been shown to be a very effective scheme [13]. However, JADE does not only modify the way in which F is set; other important components of DE are also changed. Among the novel components, the trial vector generation strategy has an important impact on performance. The strategy considers the best individuals in the population with the aim of improving the overall convergence speed. As a result, the diversity of the potential trials is reduced. Our hypothesis is that since the behavior of this scheme is less explorative and a low amount of randomness is considered, the adaptation process might be facilitated, which might explain the controversies that have appeared in some studies.

The aim of this research is to shed some light on this subject. Some of the most popular mutation scale factor adaptation schemes are compared considering several trial vector generation strategies that balance exploration and exploitation to various degrees. Our study shows the existing relationship between the vector generation strategy selected and the success or failure of mutation scale adaptation. Specifically, the adaptation fails in the most explorative schemes, which consequently generate trial vectors with a larger diversity. Since schemes that consider a large amount of diversity have been shown to be very effective for several complex problems, an important conclusion of this paper is that further research is required to successfully adapt the mutation scale factor in these cases.

The rest of the paper is organized as follows. The fundamentals of DE are presented in Section 2. Section 3 is devoted to a discussion of the balance between exploration and exploitation induced by differential trial vector generation strategies. A summary of the main mutation scale factor adaptation schemes is given in Section 4. A computational study is presented in Section 5. Finally, the main conclusions are given in Section 6.

2 Fundamentals of Differential Evolution

DE is a direct search method specially suited for single-objective continuous optimization problems [12] in which the variables governing the system to be optimized are given by a vector $\mathbf{X} = [x_1, x_2, x_3, ..., x_D]$, where D is the number of variables and each variable x_i is a real number. The quality level of each set of variables is given by the objective function $f(\mathbf{x})(f : \Omega \subseteq \mathbb{R}^D \rightarrow \mathbb{R})$. The aim of the optimization—consider a minimization problem—is to find a vector $\mathbf{x}* \in \Omega$ in which $f(\mathbf{x}*) \leq f(\mathbf{x})$ holds for all $\mathbf{x} \in \Omega$. In box-constrained optimization problems, the region Ω is specified with lower (a_j) and upper (b_j) bounds for each variable, i.e. $\Omega = \prod_{j=1}^{D} [a_j, b_j]$.

DE operates as follows. Initially, it creates a random population (P) with NP individuals ($P = [\mathbf{X}_1, ..., \mathbf{X}_{NP}]$). Each individual—also termed vector in DE—comprises D variables. The value of the variable j of the individual X_i is denoted by $X_{i,j}$. Then, successive iterations are evolved. In each DE iteration, the following steps are executed. First, for each vector in the population—called *target vector* (X_i)—a new *mutant vector* (V_i) is created using a mutant vector generation strategy. Several mutant vector generation strategies have been devised (see Section 3). Then, the mutant vector is combined with the target vector to generate the *trial vector* (U_i) through a crossover operator. The combination of the mutant vector generation strategy and the crossover operator is usually referred to as the trial vector generation strategy. After generating NP trial vectors, each one is compared against its corresponding target vector. The one that minimizes the objective function is selected to survive. In case of a tie, the offspring survives in our implementation.

The most commonly applied operator for combining the target and mutant vector—and the one considered herein—is the binomial (*bin*) crossover. The crossover operation is controlled by means of the crossover rate (CR). In the bin strategy, the trial vector is generated as shown in Eq. 1. A uniformly distributed random number in the range $[0, 1]$ is given by $rand_{i,j}$, and $j_{rand} \in [1, 2, ..., D]$ is a randomly chosen index which ensures that at least one variable is taken from the mutant vector. For the remaining cases, we see that the probability of the variable being inherited from the mutant is CR. Otherwise, the variable of the target vector is considered.

$$U_{i,j} = \begin{cases} V_{i,j} \ \ if \ (rand_{i,j} \ \leq \ CR \ or \ j \ = \ j_{rand}) \\ X_{i,j} \ otherwise \end{cases} \tag{1}$$

Finally, it is also important to note that the trial vector generation strategy, as described above, might generate vectors outside the feasible region. Several strategies for dealing with this scenario have been proposed [9]. A widely used scheme is based on randomly reinitializing the offending values in the feasible ranges. As this last approach is the most unbiased and has yielded promising results, it is the one applied in this paper.

3 Exploration and Exploitation in Differential Evolution

Several trial vector generation strategies have been proposed [7,2]. In this section, we present the strategies that are considered in our work and discuss some of their main features in terms of how they balance exploration and exploitation. Most of the trial vector generation strategies comprise a mutant vector generation strategy and a combination scheme. Regardless of the trial vector generation strategy, the term *base vector* is used to refer to an initial vector that is subsequently perturbed to generate the mutant vector. The perturbation is done by considering one or several differences among other vectors in the population. In order to classify the trial vector generation variants, the notation DE/x/y/z was introduced in [12]. The term x specifies how to select the base vector. The term y is the number of difference vectors used. Finally, z denotes the crossover or combination scheme. Thus, x and y set up the mutation strategy, and z the crossover scheme.

The most popular mutation strategy is probably the rand/1 scheme. In this strategy, any vector in the population different from the target vector is randomly selected as the base vector. Hence, the mutant vector V_i for target vector X_i is created as per Eq. 2, where r_1, r_2, and r_3 are mutually exclusive integers chosen at random from the range [1, NP]. In addition, they are all different from the index i. Since the objective value is not considered when selecting the individuals undergoing mutation, this scheme is highly explorative.

$$\mathbf{V}_i = \mathbf{X}_{r3} + F \times (\mathbf{X}_{r1} - \mathbf{X}_{r2}) \tag{2}$$

In recent years several strategies have been devised to promote intensification. The current-to-rand/1 strategy creates mutants using Eq. 3. As in the previous case, r_1, r_2 and r_3 are mutually exclusive integers different from the index i, chosen at random from the range [1, NP]. A new parameter (K) is taken into account. However, in order to facilitate the parameterization,

$K = F$ is usually considered. In this scheme, the mutant vector is created by considering the information of its own target vector. Since the target and mutant vectors are then combined, this scheme is not as explorative as the rand scheme.

$$\mathbf{V}_i = \mathbf{X}_i + K \times (\mathbf{X}_{r3} - \mathbf{X}_i) + F \times (\mathbf{X}_{r1} - \mathbf{X}_{r2}) \tag{3}$$

More intensification can be promoted by considering the current-to-best/1 strategy. The equation governing this strategy can be generated by simply substituting $r3$ in Eq. 3 with the index of the best vector in the population. The mutant vector considers the contents of the target vector and the contents of the best individual. Hence, the diversity is highly reduced in comparison with other schemes. In fact, the likelihood of stagnation and/or premature convergence is higher in this last scheme [6].

Finally, the current-to-pbest/1 strategy is a compromise between the last two schemes. In this case, a parameter p must be set. Then, the individual X_{r3} in Eq. 3 is replaced with a random individual selected from the best $p \times 100\%$ individuals. Thus, the balance between exploration and exploitation can be tuned with the value of p. In fact, the last scheme configured with $p = 1$ is similar to current-to-rand/1. In contrast, when it is configured with $p = \frac{1}{NP}$, it is similar to current-to-best/1.

4 Mutation Scale Factor Adaptation

Several schemes that consider a non-static F value have been proposed. In this section, the schemes considered in this work are briefly described. The reader is referred to [2, 13] for an extensive review of the literature. These kinds of methods can be separated into those schemes that take into account feedback to set the value of F and those that do not.

Among the schemes that do not consider any feedback, the most popular strategies set F by using a random distribution. Empirical studies have revealed that the results are highly dependent on the distribution used, though no single distribution has been shown to be superior to any other [11]. In any case, it seems that distributions with long tails—like Cauchy—have yielded promising results [8] when used with complex multimodal problems. In this work, two different random distributions are considered. The first one is a Gaussian distribution with mean 0.5 and standard deviation 0.3, which is the one applied in SaDE [10]. In the rest of the paper it is denoted as N(0.5, 0.3). The second one is the Cauchy distribution with location factor 0.5 and scale parameter 0.1, which is the initial distribution considered in JADE [15]. In the rest of the paper it is denoted as C(0.5, 0.1).

Regarding the schemes that consider feedback to adapt the scale factor, jDE [1], JADE [15] and a competitive DE [13] (cDE) are used. They are very representative of the different adaptation mechanisms presented in the literature. In jDE, each individual has its own value for F. Each time a new individual is created, a new random F value in the range $[F_{min}, F_{max}]$ is used with ratio

τ. Otherwise, the F value of the target vector is used. In JADE, the F value is generated based on a Cauchy distribution with location factor μ_F and scale parameter 0.1. If the generated value is lower than 0, it is regenerated. If it is higher than 1, it is truncated to 1. The location factor is initialized to 0.5 and updated after each generation by considering the Lehmer mean of the successful F values, the previous location factor and a parameter c, which represents the location factor's adaptation speed. Since a Lehmer mean is used, there is a bias towards larger F values. Finally, in cDE a set of values is considered to set up a pool of candidates. The probability of using each value is proportional to the number of times that it has been used successfully in previous stages, modified with a parameter n_0 that prevents drastic changes in the probabilities. To avoid degeneration of the search process, if any probability decreases below some given limit δ, the memory of the scheme is erased, i.e. the probabilities are reset to their initial values.

5 Experimental Study

Different trial vector generation strategies induce dissimilar balances between exploration and exploitation. In this section, we present a set of experiments whose aim is to demonstrate the relationship between this balance and the effectiveness of the mechanisms for adapting the mutation scale factor. The analyses were performed using the benchmark problems described in [4], which are a set of 19 scalable continuous optimization problems to be minimized. The parameter D allows setting the number of variables in the problems. In our study, it was set to 50. In order to analyze the schemes, two sets of experiments were carried out. In every case, each execution was repeated 1,000 times.

Since stochastic algorithms were considered, comparisons were carried out by applying the following statistical analysis, assuming a significance level of 5%. First, a *Shapiro-Wilk test* was performed to check whether or not the values of the results followed a Gaussian distribution. If so, the *Levene test* was used to check the homogeneity of the variances. If the samples had equal variance, an ANOVA *test* was done; if not, a *Welch test* was performed. For non-Gaussian distributions, the non-parametric *Kruskal-Wallis* test was used. In this work, the sentence "algorithm A is better than algorithm B" means that the differences between them are statistically significant, and that the mean and median obtained by A are lower—one of the metrics might be equal—than the mean and median achieved by B.

5.1 First Experiment: Exploitative Schemes

Since JADE is one of the most promising adaptive schemes, the objective of the first study was to analyze whether the mutation scale factor mechanism included in JADE is helpful—note that in the original paper the benefits of each single modification are not analyzed. Thus, its trial vector generation

Table 1 Statistical comparison of different F adaptive variants with the Current-to-pbest/1/bin strategy in 250,000 function evaluations

Strategy	↑	↓	Strategy	↑	↓
JADE	60	3	cDE	14	51
Cauchy(0.5, 0.1)	45	16	jDE	6	69
Normal(0.5, 0.3)	36	28	-	-	-

strategy (current-to-pbest/1/bin) was considered. Following the recommendations given in [15], intensification was promoted by considering a low p value ($p = 0.05$). In addition to the adaptation proposed in JADE, the ones proposed in jDE, cDE, as well as the random distributions N(0.5, 0.3) and C(0.5, 0.1), were also tested. The adaptive schemes were parameterized considering the recommendations given by their corresponding authors. In JADE, the c parameter was set to 0.1. In jDE, F_{min}, F_{max} and τ were set to 0.1, 0.9 and 0.1, respectively. Finally, in cDE, the pool comprised ten values equidistributed between 0.1 and 1, n_0 was set to 2 and δ was set to 0.02. In every case, the CR value was adapted with the mechanism proposed by JADE, and NP was set to 100. Some preliminary experiments were carried out to select the proper value for NP. Finally, the stopping criterion was set to 250,000 function evaluations.

In order to obtain an overall ranking of the different approaches tested, pairwise statistical comparisons between the five schemes tested were carried out. Since the benchmark set comprised 19 problems, 76 statistical tests were done for each scheme. Table 1 shows the number of tests where each model was better (↑) or worse (↓) than the other schemes considering the statistical approach discussed above. The JADE scheme was clearly superior to the rest of the schemes. In fact, it was only outperformed in 3 cases. An inspection of the trend of μ_F reveals that, at the end of the executions, its mean value was larger than 0.5 in every problem. The bias towards large F values introduced by the scheme is very beneficial because large F values might allow individuals to escape from non-optimal attraction basins. Thus, the combination of bias and feedback is quite convenient. It is also interesting to note that the two random distributions tested yielded better results than cDE and jDE.

5.2 Second Experiment: Increasing the Balance Towards Exploration

Since our hypothesis is that the F adaptation mechanisms fail with the explorative schemes, in this second experiment more explorative DE variants are considered. In DE/current-to-pbest/bin, exploration can be promoted by increasing the p value. Thus, experiments with different p values were carried out. Specifically, the JADE and C(0.5, 0.1) schemes were executed with ten p values equidistributed between 0.1 and 1. In addition, the values 0.01 and 0.05 were also considered. Figure 1 shows, for each case, the number of problems where JADE was better than C(0.5, 0.1), as well as the number of problems where the opposite was true. As exploration was promoted, the advantages of JADE diminished. In fact, considering feedback for large p values was not

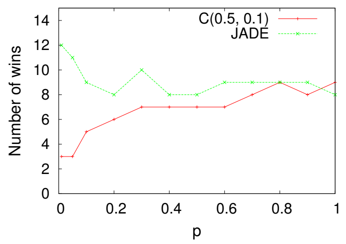

Fig. 1 Statistical comparison between JADE and C(0.5, 0.1) with current-to-pbest

Table 2 Statistical comparison of different F adaptive variants with the rand/1/bin strategy in 250,000 function evaluations

Strategy	↑	↓	Strategy	↑	↓
Cauchy(0.5, 0.1)	31	10	cDE	14	28
Normal(0.5, 0.3)	28	11	JADE	11	35
jDE	20	20	-	-	-

helpful. First, since the scheme is more explorative, the bias towards large F values introduced by JADE is not required. Additionally, the higher degree of randomness hampers the use of adaptive schemes.

In order to confirm the trend discovered in the previous analysis, a more explorative scheme (rand/1/bin) was considered. It was tested with the same adaptive schemes as in the first set of experiments. In this case, NP was set to 50 to avoid an excessive bias towards exploration. Table 2 shows the results of the pairwise statistical comparisons. In this case, the C(0.5, 0.1) distribution obtained the best overall results, demonstrating the poor performance of schemes that consider feedback to set the mutation scale factor.

Finally, it is also interesting to compare the results obtained with current-to-pbest and rand. Table 3 shows the mean and median of the errors obtained by the best schemes for each case. In the problems where differences were statistically significant, **boldface** is used. The overall results obtained with the rand scheme—even with a random distribution to set the F values—were superior. The current-to-pbest scheme yielded better results in only three problems. In the case of the F8 function, it can be easily solved using only local search [3], so the intensification promoted by current-to-pbest is adequate. F3 and F13 are the "banana" function and one of its variants. In these cases, there is a large flat, curved valley that can keep us from finding the optimum. In the current-to-pbest scheme with a large population size, this region is approached from many directions, increasing the probability of entering the region through a zone close to the optimum. Finally, in F4, the model considering the rand strategy obtained the optimum in every execution except one. Since this execution achieved a poor quality solution, it had a better median but worse mean than current-to-pbest. In any case, the overall superiority of rand is clear.

Table 3 Comparison of errors obtained with different trial vector generation strategies and their corresponding best F adaptive schemes in 250,000 evaluations

	current-to-pbest/1/bin — JADE		rand/1/bin — Cauchy(0.5, 0.1)	
	Median	Mean	Median	Mean
F1	0	0	0	0
F2	9.75	$1.00 \cdot 10^{1}$	$\mathbf{3.07 \cdot 10^{-2}}$	$\mathbf{5.97 \cdot 10^{-1}}$
F3	**3.73**	**5.32**	$8.34 \cdot 10^{1}$	$7.85 \cdot 10^{1}$
F4	$1.60 \cdot 10^{-11}$	$\mathbf{2.03 \cdot 10^{-11}}$	**0**	$1.98 \cdot 10^{-2}$
F5	0	$9.48 \cdot 10^{-4}$	**0**	**0**
F6	$5.68 \cdot 10^{-14}$	$3.51 \cdot 10^{-3}$	$\mathbf{5.68 \cdot 10^{-14}}$	$\mathbf{5.68 \cdot 10^{-14}}$
F7	0	0	0	0
F8	$\mathbf{1.57 \cdot 10^{-9}}$	$\mathbf{3.69 \cdot 10^{-9}}$	$1.05 \cdot 10^{4}$	$9.74 \cdot 10^{3}$
F9	$4.89 \cdot 10^{-2}$	$7.84 \cdot 10^{-2}$	**0**	**0**
F10	1.04	1.12	**0**	**0**
F11	$6.24 \cdot 10^{-2}$	$9.09 \cdot 10^{-2}$	**0**	**0**
F12	$1.11 \cdot 10^{-10}$	$6.30 \cdot 10^{-10}$	$\mathbf{2.82 \cdot 10^{-17}}$	$\mathbf{2.53 \cdot 10^{-17}}$
F13	$\mathbf{3.21 \cdot 10^{1}}$	$\mathbf{2.58 \cdot 10^{1}}$	$3.24 \cdot 10^{1}$	$3.39 \cdot 10^{1}$
F14	$1.18 \cdot 10^{-3}$	$1.24 \cdot 10^{-3}$	$\mathbf{2.05 \cdot 10^{-17}}$	$\mathbf{9.94 \cdot 10^{-4}}$
F15	0	$3.83 \cdot 10^{-3}$	**0**	**0**
F16	$1.37 \cdot 10^{-5}$	$3.01 \cdot 10^{-5}$	$\mathbf{1.85 \cdot 10^{-12}}$	$\mathbf{1.97 \cdot 10^{-12}}$
F17	8.69	9.33	**1.57**	**2.78**
F18	$2.50 \cdot 10^{-1}$	$2.52 \cdot 10^{-1}$	$\mathbf{9.11 \cdot 10^{-10}}$	$\mathbf{9.38 \cdot 10^{-10}}$
F19	0	$2.05 \cdot 10^{-1}$	**0**	**0**

6 Conclusions

Adaptive schemes have gained considerable popularity in recent years in the field of Evolutionary Computation. In the case of DE, several schemes have been proposed that consider the feedback obtained during the optimization process to guide the setting of the mutation scale factor. In most cases, not only is the mutation scale factor adapted, but other components are also changed. However, comparisons do not usually measure the benefits obtained by every single adaptation. This has led to some controversy in the field. While many schemes for adapting the mutation scale factor have been proposed, other researchers have not obtained benefits when considering adaptive schemes.

In this paper we demonstrate that there is a relationship between the diversity introduced by the trial vector generation strategies and the success or failure of adaptive schemes. Specifically, only when the current-to-pbest strategy with a low p value is considered does the use of feedback provide significant benefits. In this case, JADE intelligently combines feedback with the well-known recommendation to use large F values occasionally in this strategy. However, as the balance is moved towards exploration, the advantages given by the use of feedback gradually disappear. In fact, with the most explorative trial vector generation strategies, setting the mutation scale factor by using random distributions with long tails provides the best results. Since the most explorative schemes are more robust, an important open challenge is to develop adaptive schemes that can profit from the feedback gained in the optimization process and can be applied with these strategies.

Acknowledgements The second author is also affiliated with the UMI LAFMIA 3175 CNRS at CINVESTAV-IPN. He also acknowledges the financial support from CONACyT project no. 103570. This work was also funded in part by the EC (FEDER) and the Spanish Ministry of Science and Innovation as part of the 'Plan Nacional de I+D+i', with contract number TIN2011-25448. The work of Eduardo Segredo was funded by grant FPU-AP2009-0457.

References

1. Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems **10**(6), 646–657 (2006)
2. Das, S., Suganthan, P.: Differential evolution: A survey of the state-of-the-art **15**(1), 4–31 (2011)
3. LaTorre, A., Muelas, S., Pea, J.M.: A mos-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test. Soft Computing **15**(11), 2187–2199 (2011)
4. Lozano, M., Molina, D., Herrera, F.: Editorial Scalability of Evolutionary Algorithms and Other Metaheuristics for Large-scale Continuous Optimization Problems. Soft Computing - A Fusion of Foundations, Methodologies and Applications pp. 1–3 (2010)
5. Mallipeddi, R., Suganthan, P., Pan, Q., Tasgetiren, M.: Differential evolution algorithm with ensemble of parameters and mutation strategies. Applied Soft Computing **11**(2), 1679 – 1696 (2011)
6. Mezura-Montes, E.n., Velázquez-Reyes, J., Coello Coello, C.A.: A comparative study of differential evolution variants for global optimization. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation, GECCO '06, pp. 485–492. ACM, New York, NY, USA (2006)
7. Neri, F., Tirronen, V.: Recent advances in differential evolution: a survey and experimental analysis. Artificial Intelligence Review **33**(1-2), 61–106 (2010)
8. Price, K.: Differential evolution. In: I. Zelinka, V. Snel, A. Abraham (eds.) Handbook of Optimization, *Intelligent Systems Reference Library*, vol. 38, pp. 187–214. Springer Berlin Heidelberg (2013)
9. Price, K., Storn, R., Lampinen, J.: Differential Evolution: A Practical Approach to Global Optimization. Natural Computing Series. U.S. Government Printing Office (2005)
10. Qin, A.K., Huang, V.L., Suganthan, P.: Differential evolution algorithm with strategy adaptation for global numerical optimization **13**(2), 398–417 (2009)
11. Soliman, O.S., Bui, L.T., Abbass, H.A.: The effect of a stochastic step length on the performance of the differential evolution algorithm. In: 2007 IEEE Congress on Evolutionary Computation (CEC'07), pp. 2850–2857 (2007)
12. Storn, R., Price, K.: Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. J. of Global Optimization **11**(4), 341–359 (1997)
13. Tvrdk, J., Polkov, R., Veselsk, J., Bujok, P.: Adaptive variants of differential evolution: Towards control-parameter-free optimizers. In: I. Zelinka, V. Snel, A. Abraham (eds.) Handbook of Optimization, *Intelligent Systems Reference Library*, vol. 38, pp. 423–449. Springer Berlin Heidelberg (2013)
14. Wang, Y., Cai, Z., Zhang, Q.: Differential evolution with composite trial vector generation strategies and control parameters **15**(1), 55–66 (2011)
15. Zhang, J., Sanderson, A.: Jade: Adaptive differential evolution with optional external archive **13**(5), 945–958 (2009)
16. Zielinski, K., Wang, X., Laur, R.: Comparison of adaptive approaches for differential evolution. In: G. Rudolph, T. Jansen, S. Lucas, C. Poloni, N. Beume (eds.) Parallel Problem Solving from Nature PPSN X, *Lecture Notes in Computer Science*, vol. 5199, pp. 641–650. Springer Berlin Heidelberg (2008)
17. Zielinski, K., Weitkemper, P., Laur, R., Kammeyer, K.D.: Parameter study for differential evolution using a power allocation problem including interference cancellation. In: IEEE Congress on Evolutionary Computation 2006 (CEC 2006), pp. 1857–1864 (2006)