Swarm and Evolutionary Computation 44 (2019) 546–558

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Differential Evolution: A survey of theoretical analyses

Karol R. Opara [a,*], Jarosław Arabas [b]

[a] Systems Research Institute, Polish Academy of Sciences, Poland
[b] Institute of Computer Science, Warsaw University of Technology, Poland

ARTICLE INFO

Keywords:
DE population dynamics
Diversity
Convergence
Global optimization
Evolutionary algorithm

ABSTRACT

Differential Evolution (DE) is a state-of-the art global optimization technique. Considerable research effort has been made to improve this algorithm and apply it to a variety of practical problems. Nevertheless, analytical studies concerning DE are rather rare. This paper surveys the theoretical results obtained so far for DE. A discussion of genetic operators characteristic of DE is coupled with an overview of the population diversity and dynamics models. A comprehensive view on the current-day understanding of the underlying mechanisms of DE is complemented by a list of promising research directions.

1. Introduction

Differential Evolution (DE) is a simple and effective evolutionary algorithm used to solve global optimization problems in a continuous domain [1,2]. It was proposed by Price and Storn in 1995 in a series of papers [3–5] and since then, it has attracted the interest of researchers and practitioners. Comprehensive survey papers [6,7] provide an up-to-date view on this algorithm and discuss its various modifications, improvements and uses.

A growing interest in DE is reflected in the number of publications citing the most distinguished original paper [5]. The top plot in Fig. 1 illustrates the citation count of this paper retrieved from the Scopus database. For comparison, the bottom plot shows the relative increase in the number of articles about evolutionary algorithms (EA) indexed in the database. As the reference level 100%, we chose the number of publications about EA from 1999. The above-average increase in interest in DE is noticeable from 2004. In the last few years, the number of citations stabilized at a level of over 1000 papers a year, which shows the importance of this metaheuristic.

Roughly half of the papers citing the original DE article [5] describe its application in various domains. The rest offer modifications and improvements of the algorithm. Despite this massive amount of research, very few papers concern the theoretical foundations of DE. These studies lead to a better understanding of the algorithm and also provide tips for tuning parameters and further modifications. Only single sections in Refs. [6–8] are devoted to discussing the most prominent analytical results about the algorithm. This paper addresses this gap by comprehensively surveying the theoretical studies on DE.

Section 2 introduces the DE algorithm and establishes notation. Results concerning convergence analysis of DE and the role of invariances in this algorithm are discussed in section 3. Several papers have been devoted to modelling the differential mutation and crossover operators and their influence on population diversity. These results are summarized in section 4. To close the evolutionary loop, selection must be applied to the offspring population. Modelling this step is challenging because of dependence on the generally unknown objective function. The resulting population dynamics models are comprehensive but rather complex. They are critically reviewed in section 5. Several promising directions of further study in the theory of DE are unfolded in section 6, which is followed by a brief conclusion of the paper.

2. Differential Evolution

The DE algorithm processes the population $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_p}\}$ consisting of N_p individuals encoded as n-dimensional vectors of real numbers. After random initialization, the population iteratively undergoes mutation, crossover and becomes subject to a selection mechanism. The pseudocode of DE is given as Algorithm 1 We denote the

* Corresponding author.
E-mail address: karol.opara@ibspan.waw.pl (K.R. Opara).

https://doi.org/10.1016/j.swevo.2018.06.010
Received 3 June 2017; Received in revised form 26 March 2018; Accepted 21 June 2018
Available online 25 June 2018

Notation

Scalar values are written in italics. Lowercase bold letters are column vectors and uppercase bold letters denote matrices. Textual subscripts are written in regular font and indexing ones in italics.

Abbreviations

DE	Differential Evolution
CDF	cumulative distribution function
PDF	probability density function

Greek symbols

λ	weight of the target vector, see equations (4) and (5)
Φ	CDF of standard normal random variable

Latin symbols

C_r	crossover rate (parameter of DE)
$\mathbf{C}$	covariance matrix
$\mathrm{Cov}, \widehat{\mathrm{Cov}}$	covariance matrix operator and its estimator
$\mathrm{E}, \widehat{\mathrm{E}}$	expectation vector operator and its estimator
f	objective function
F	scaling factor (parameter of DE)
$g(F)$	generalized scaling factor, see (25)
h	PDF of infinite population
h_m	PDF of mutants in infinite population model, see (32)
k	number of difference vectors
n	search space dimension
$\mathcal{N}(m, v)$	normal distribution with mean m and variance v
N_p	population size (parameter of DE)
$\mathbf{O}$	offspring population
$\mathcal{O}$	big O notation
$\mathbf{o}_i$	vector encoding the ith offspring
p_c	probability of the cth realization of crossover
p_f	probability of strategy selection in DE/either-or, see (6)
p_m	probability of mutation (a function of C_r), see (15)
$\mathbb{R}$	set of real numbers
^{T}	transposition of a vector or matrix
t	iteration number (typically written as a superscript)
$\mathbf{U}$	mutant population
$\mathbf{u}_i$	vector encoding the ith mutant, see (1–6)
$\mathrm{Var}, \widehat{\mathrm{Var}}$	variance operator (in one-dimension) and its estimator
v	variance of a one-dimensional random variable
$\mathbf{x}_i$	vector encoding the ith individual in the population
$\mathbf{x}_\mathrm{best}$	best (fittest) individual in the population
$\mathbf{x}_\mathrm{mean}$	population midpoint, see (12)
$\mathbf{x}_r$	randomly selected individual
$\mathbf{X}$	population of individuals in DE

iteration number by superscript ^{t}, but this is omitted whenever it does not lead to ambiguity.

Algorithm 1 Differential Evolution (DE).
Initialize parameters C_r, F, and N_p and iteration counter $t \leftarrow 0$
Initialize population $\mathbf{X} = \left\{\mathbf{x}_1^t, \mathbf{x}_2^t, \ldots, \mathbf{x}_{N_\mathrm{p}}^t\right\}$
while stop condition not met **do**
  for all $i \in \{1, 2, \ldots, N_\mathrm{p}\}$ **do**
    $\mathbf{u}_i \leftarrow$ differential mutation $(F; i, \mathbf{X})$
    $\mathbf{o}_i \leftarrow$ crossover $\left(C_\mathrm{r}; \mathbf{x}_i^t, \mathbf{u}_i\right)$
    if $f(\mathbf{o}_i) \le f\left(\mathbf{x}_i^t\right)$ **then**
      $\mathbf{x}_i^{t+1} \leftarrow \mathbf{o}_i$
    else
      $\mathbf{x}_i^{t+1} \leftarrow \mathbf{x}_i^t$
    end if
  end for
  $t \leftarrow t + 1$
end while
return $\mathrm{argmax}_{\mathbf{x}_i^t} f\left(\mathbf{x}_i^t\right)$

Price and Storn [5] proposed a few variants of the DE algorithm, introducing notations DE/X/Y/Z, where X denotes the reproduction method, $Y$the number of difference vectors, and Z indicates the crossover operator.

2.1. Differential mutation

The differential mutation operator has a few basic variants which are described in references [5,9]. The most common one, denoted as DE/rand/1, consists of randomly choosing three individuals from the population and adding to the first of them $\mathbf{x}_{r_1}$ (known as the base vector) a scaled difference between two others $\mathbf{x}_{r_2}$ and $\mathbf{x}_{r_3}$

$$\mathbf{u}_i \leftarrow \mathbf{x}_{r_1} + F \cdot \left(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\right). \tag{1}$$

Parameter F is called a scaling factor, as it shrinks the length of the difference vector $(\mathbf{x}_{r_2} - \mathbf{x}_{r_3})$ appearing in equation (1). The indices that are the realizations of a random variable are denoted by r, whereas the indices that are deterministic are represented by other symbols such as i or j. The indices of individuals used by the mutation operator are often required to be pairwise distinct, however this assumption is usually not taken into account in theoretical analyses; cf. [10–13]. Skipping this condition causes independence of the location of the mutant $\mathbf{u}_i$ from the location of its parent $\mathbf{x}_i$. Keeping it would introduce only a weak dependence, especially for large populations, and would not hinder the optimization performance [14].

Price and Storn also defined the DE/rand/k variant which uses a larger number of difference vectors:

$$\mathbf{u}_i \leftarrow \mathbf{x}_{r_1} + F_1 \cdot \left(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\right) + \cdots + F_k \cdot \left(\mathbf{x}_{r_{2k}} - \mathbf{x}_{r_{2k+1}}\right), \tag{2}$$

where the scaling factors are usually assumed to be equal $F_1 = F_2 = \ldots = F_k = F$. In practice, the most frequently encountered mutation operators have one or two difference vectors.

Replacing a random base vector $\mathbf{x}_{r_1}$ with the best vector $\mathbf{x}_\mathrm{best}$ from the population gives another important variant of DE, denoted as DE/best/1:

$$\mathbf{u}_i \leftarrow \mathbf{x}_\mathrm{best} + F \cdot \left(\mathbf{x}_{r_1} - \mathbf{x}_{r_2}\right). \tag{3}$$

The majority of differential mutation operators can be represented as special cases of a generalized formula which uses a sum of k scaled difference vectors and a weighted mean between the best individual and a randomly selected one:

$$\mathbf{u}_i \leftarrow \lambda \mathbf{x}_\mathrm{best} + (1-\lambda)\,\mathbf{x}_{r_1} + \sum_{j=1}^{k} F_j \cdot \left(\mathbf{x}_{r_{2j}} - \mathbf{x}_{r_{2j+1}}\right), \tag{4}$$

where λ is a scalar taking values in the range from 0 to 1. In particular, formula (4) covers several common variants of DE mutation: DE/rand/1 ($\lambda = 0, k = 1$), DE/rand/2 ($\lambda = 0, k = 2$), DE/best/1 ($\lambda = 1, k = 1$).

A weighted mean between the best individual and current individual (rather than a random one) is used in the DE/current-to-best/k operator:

$$\mathbf{u}_i \leftarrow \lambda \mathbf{x}_\mathrm{best} + (1-\lambda)\,\mathbf{x}_i + \sum_{j=1}^{k} F_j \cdot \left(\mathbf{x}_{r_{2j-1}} - \mathbf{x}_{r_{2j}}\right). \tag{5}$$

The most complex of the proposed mutation variants is DE/either-or [9,15], in which with probability p_f, mutants are generated according to scheme DE/rand/1, and with probability $1 - p_\mathrm{f}$, according to a modified relation:

$$\mathbf{u}_i \leftarrow \begin{cases} \mathbf{x}_{r_1} + F \cdot \left(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\right) & \text{if rand}\,(0,1) < p_\mathrm{f}, \\ x_{r_1} + K \cdot \left(\mathbf{x}_{r_2} + \mathbf{x}_{r_3} - 2\mathbf{x}_{r_1}\right) & \text{otherwise}, \end{cases} \tag{6}$$

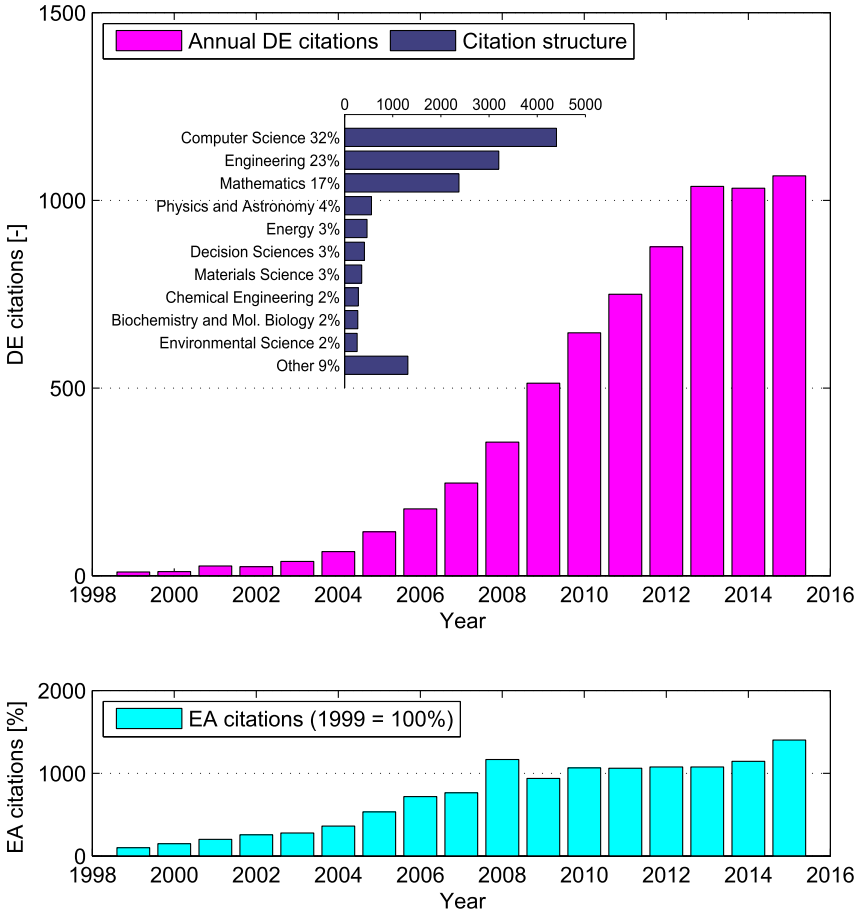

Fig. 1. Top Figure: Number of citations of paper [5] introducing DE and the division of citations into disciplines based on the Scopus bibliography database; Bottom Figure: Number of citations within evolutionary algorithms in percent.

where $K = \frac{F+1}{2}$. This operator replaces both differential mutation and crossover operators, ensuring rotational invariance of the resulting algorithm.

2.2. Crossover

In DE, the crossover operator is based on exchanging elements between vectors encoding the parent and the mutant. It is introduced to promote the population diversity [6]. In n-dimensional space, an offspring $\mathbf{o}_i = [o_1, \ldots, o_n]^T$ is created through exchanging elements of vectors that represent its parent $\mathbf{x}_i = [x_1, \ldots, x_n]^T$ and the mutant $\mathbf{u}_i = [u_1, \ldots, u_n]^T$. The most common variant is binomial crossover, denoted as DE/X/Y/bin. The offspring is created by randomly choosing elements either from the parent or from the mutant:

$$o_j = \begin{cases} u_j & \text{if } \mathrm{rand}(0,1) \le C_\mathrm{r} \\ x_j & \text{otherwise} \end{cases} \quad \text{for } j = 1, \ldots, n. \tag{7}$$

The number of elements inherited from the mutant is described using binomial distribution with n independent repeats and success probability C_r. The offspring is also required to contain at least one element from the mutant, however this assumption is often neglected in theoretical studies; cf. [10,11]. Under unit crossover rate $C_\mathrm{r} = 1$, the offspring is equal to the mutant $\mathbf{o}_i = \mathbf{u}_i$.

In exponential crossover DE/X/Y/exp, two numbers are randomly drawn. The first one is the index k which is uniformly distributed in the range from 1 to n, and the second is the length l, which is sampled from a censored geometric distribution with success probability C_r. The offspring is a clone of its parent, in which elements from k to $k + l$ (modulo search space dimension n) come from the mutant:

$$o_j = \begin{cases} u_j & \text{if } j-1 \in \{k \pmod n, \ldots, k+l \pmod n\} \\ x_j & \text{otherwise} \end{cases} \quad \text{for } j = 1, \ldots, n. \tag{8}$$

The continuous equivalent of the geometric distribution is the exponential distribution, which explains the symbol DE/X/Y/exp.

2.3. Reproduction and selection

The selective pressure in DE is introduced primarily by a competition between a parent and an offspring. However, the effect of guiding the population towards regions with better values of the objective function is amplified by some of the differential mutation operators. This additional selective pressure is imposed by the reproduction mechanism, i.e. the choice of base vector, which is typically described as an element of the differential mutation operator. In DE/rand/Y/Z, an individual $\mathbf{x}_{r_1}$ drawn randomly from the population reproduces, while in DE/best/Y/Z only the best one $\mathbf{x}_\mathrm{best}$ reproduces.

Selection consists of comparing an offspring with its parent. Only the better of them is chosen for the next generation. The greediness of this mechanism induces elitism, but it operates locally, which promotes the maintenance of the population spread. This approach is distinct from population-based methods of proportional and rank reproduction [16].

Selection in DE can be implemented either after the completion of the whole population of mutants or in a steady-state manner through

immediately replacing the losing vector. Most of the theoretical studies address the former approach.

2.4. Generalizations of DE

A large number of modifications to the original DE algorithm have been proposed so far. These include:

1. Parameter and strategy adaptation in DE [17], e.g. randomization and self-adaptation of parameters [18,19], adaptation of the population size [20,21], strategy adaptation [22].
2. Modification of the differential mutation operator, e.g. by choosing the individuals based on their mutual distances [23] or fitness values [24].
3. Enhancing DE with new mechanisms such as population archive [25], genotypic topology [26], opposition-based initialization [27] etc.
4. Hybridization. For instance, algorithm AM-DEGL [28] uses global and local neighbourhoods from paper [26], self-adaptation of parameters inspired by article [22] and local optimization based on the approach proposed in study [29] and conducted with a classical optimizer [30].

These and many more algorithms are comprehensively discussed in surveys [6,7]. The theoretical analyses are typically done for classical DE. Sometimes, the obtained results can be easily generalized to describe more cases. However, the introduction of new evolutionary mechanisms makes this task more challenging and creates a need for tailored theoretical studies.

2.5. Minimalistic variants of DE

Obtaining analytical results is difficult for complex DE frameworks. On the other hand, simple methods lead to clear, interpretable models with low computational overhead. Such a bottom-up approach towards algorithmic design does not necessarily hinder the performance of the algorithm [31,32].

A minimalistic variant of DE was recently introduced by Price [33]. The Black-Box Differential Evolution (BBDE) algorithm consists only of differential mutation and local selection. The current individual $\mathbf{x}_i$ is used as the base vector and generates an offspring through the addition of a randomly scaled difference vector:

$$\mathbf{o}_i \leftarrow \mathbf{x}_i + F_r \cdot \left(\mathbf{x}_{r_1} - \mathbf{x}_{r_2}\right), \tag{9}$$

where F_r is sampled from a log-normal distribution. Next, the offspring competes with its parent for survival to the following generation.

Scrapping all non-essential elements of DE was done in an attempt to introduce possibly many invariances to the search algorithm in order to improve its robustness. This approach seems to be inspired by Hansen's work on another state-of-the-art evolutionary algorithm, CMA-ES [34].

3. Convergence and invariances

There is a large body of papers on the theoretical foundations of classical evolutionary algorithms. Numerous studies concern individual aspects of these algorithms, such as the encoding of individuals [35], adaptation mechanisms [36] or constraint handling techniques [37]. Many papers address individual algorithms – their description can be found e.g. in articles [38,39]. The usefulness of these studies is limited by strong assumptions and significant differences between particular algorithms.

In the following sections, we indicate the main areas of theoretical studies of evolutionary algorithms to present the analyses conducted for DE in a wider context.

3.1. Computational complexity

Evolutionary algorithms generally lack theoretically justified stopping criteria. Usually, the algorithm is stopped after exceeding a maximum number of iterations $G_{\max}$. Stopping conditions could also be based on monitoring the population diversity, the required value of the objective function or the frequency of improvements of the best individual [40]. Parallel and distributed processing poses new conceptual challenges towards measuring complexity [41].

For serially processed DE, computational complexity is driven by the number of calls to genetic operators [42]. Their execution time is proportional to the search space dimension n. Consequently, the number of elementary operations is proportional to the maximal iteration number $G_{\max}$ (the outer loop in Pseudocode 1) and the population size N_{p} (the inner loop):

$$\mathcal{O}(n \cdot N_{\mathrm{p}} \cdot G_{\max}). \tag{10}$$

Formula (10) implicitly assumes that the computational complexity of the objective function evaluation $c(n)$ is at most linearly dependent on the search space dimension n. Otherwise, the complexity of DE is not driven by the execution of genetic operators but rather by the number of objective function evaluations:

$$\mathcal{O}(c(n) \cdot N_{\mathrm{p}} \cdot G_{\max}). \tag{11}$$

The termination of computations does not necessarily coincide with finding the global minimum.

3.2. Convergence proofs

Rudolph [43] provided a sufficient condition for a weak convergence of a continuous optimization method to the global optimum. An algorithm, after initialization at any feasible point, must with non-zero probability generate a finite series of points finishing in a non-zero measure neighbourhood of any other feasible point. Consequently, weak convergence happens if the whole feasible set is sufficiently densely sampled. This allows one to check which optimization methods cannot guarantee global convergence. Classical DE is one of such methods. When the whole population is initialized within a sufficiently large attraction basin of a single local optimum, the population cannot leave this basin because of elitist selection. Formal proof of this statement is given by Hu et al. [44] using drift analysis for a specially created "deceptive" function.

Hu et al. [45] show that convergence with probability 1 can be obtained after softening the selection in DE and adding a mutation strategy that samples from the whole feasible set. Another way to introduce the global optimization property to DE is to re-initialize the population, or its part, every k_{tol} iterations [46]. Although the global convergence property is desirable for evolutionary algorithms, its practical usefulness is typically limited by the prohibitively long time necessary to ensure that the optimum is reached.

3.3. Invariances

Invariances imply homogeneous performance for classes of objective functions. This allows empirical and theoretical results to be generalized. The incorporation of invariances is argued to be a fundamental design criterion for the development of search algorithms [47]. Classical DE is characterized by scale and translation invariance. DE behaves in the same way for an objective function $f : \mathbb{R}^n \to \mathbb{R}$ and its transform $f_1(\mathbf{x}) = f(a\mathbf{x} + \mathbf{b})$ if the initial population is scaled accordingly, where $a \neq 0$ is a scalar and $\mathbf{b} \in \mathbb{R}^n$ a translation vector.

DE is also invariant to order-preserving transformations. The selection mechanism of DE requires only an indication of which of the two solutions is preferable. Therefore, DE behaves in the same way for an objective function f and its order-preserving transformations, such as

shifting $f_2(\mathbf{x}) = a + f(\mathbf{x})$, scaling by a positive number $a > 0$, $f_3(\mathbf{x}) = a \cdot f(\mathbf{x})$, or composition with a strictly increasing function $s : \mathbb{R} \to \mathbb{R}$, $f_4(\mathbf{x}) = s(f(\mathbf{x}))$. Consequently, results obtained for simple models generalize to the whole classes of problems. For instance, DE behaves in exactly the same way for the quadratic function $f(\mathbf{x}) = \mathbf{x}^T\mathbf{x}$ as for all radial functions, whose values depend solely on the distance from the origin and are strictly increasing. DE uses only the order between individuals and operates in an ordinal rather than interval scale. This property might be useful in the rare cases when it is possible to compare two solutions, but it is not feasible to quantify the objective function, e.g. when comparing preference towards certain market baskets.

Classical DE is not rotationally invariant. An angle-preserving transformation of the search space changes the offspring distribution. This is a result of crossover operators which exchange elements along the axes of the coordinate system. This problem is addressed by rotationally invariant crossover operators. In DE/either-or [9], arithmetic recombination is built into the differential mutation. As an alternative solution, Guo and Yang [48] proposed crossover along the eigenvectors of the population rather than the axes of the coordinate system. This approach makes use of the experimentally observed adaptation of the population to the local shape of the objective function, which is known as "contour fitting".

Price [15] analysed both classical DE and DE/either-or. He showed that these algorithms suffer from mutation bias, i.e. a situation in which the mutants are not centre-symmetrically distributed around the current vector. The requirement of central symmetry is stronger than the coincidence of the expectation of the mutant distribution with the current vector.

Price [15] also underlined the presence of selection bias, i.e. a situation in which trial vectors are not symmetric around the current individual but shifted towards the population midpoint:

$$\mathbf{x}_{\text{mean}} = \frac{1}{N_\text{p}} \sum_{i=1}^{N_\text{p}} \mathbf{x}_i. \tag{12}$$

Selection bias disappears if the current individual $\mathbf{x}_i$ is used as the base vector. These ideas have led to a drift-free DE, in which "each operation has a unique function: mutation explores, recombination homogenizes and selection improves" [15]. Price elaborated this approach further in the BBDE algorithm [33] presented in section 2.4.

3.4. Universal optimization algorithm

The *no free lunch* theorem for optimization was proved by Wolpert and Macready in article [49] and further discussed by Köppen et al. [50]. It shows that in a finite search space for any non-repeating optimization algorithm and any performance measure, obtaining above-average performance for one class of problems correlates with below-average performance for another class. There are some discussions about the assumptions necessary to generalize the *no free lunch* theorem to continuous domains [51]. No single "best" universal optimizer exists, but for particular problem classes, some algorithms give much better results than others [52].

Optimization methods are effective if they exploit the regularities of the problem [53,54] such as symmetry, convexity or continuity. From the theoretical point of view, a "typical" optimization task is extremely noisy and has no structure [55]. On the other hand, practical problems are most often characterized by a large number of regularities, such as continuity. Consequently, the *no free lunch* theorem does not undermine the sense of developing new optimizers [53]. Instead, it shows the necessity to appropriately match the problems with the algorithms [52].

Properties of the optimization problem may themselves become the subject of analyses. Characterizing the influence of the problem properties on the algorithm's performance is a way to overcome the limitations imposed by the *no free lunch* theorem by appropriately assigning the algorithm to a given task [56].

3.5. Other approaches

Among the first theoretical results about genetic algorithms were the schema theorem [57] and building block hypothesis [58]. These approaches assume that the optimization problem is separable to some degree. This echoes in the exponential crossover operator in DE, which exchanges only consecutive elements of the vectors that encode the parent and the mutant. However, the building block hypothesis does not explain the performance-related issues of the search process [39,59].

Population drift in evolutionary algorithms can be also modelled using supermartingales [60,61]. This kind of analysis can be used to derive the lower bound of the expected runtime of the analysed algorithm [62].

4. Population diversity in DE

Before discussing the results of studies concerning population diversity and dynamics in DE, it is worth summarizing the assumptions commonly made by researchers:

1. Individuals for differential mutation are drawn with replacement, i.e. the requirement of pairwise distinctness is abandoned, to make these vectors independent [10–13,63–66].
2. It is not required that at least one element of the mutant must be crossed over with the parent vector [10–12,64].
3. DE is run in a one-dimensional search space, which makes formulas and proofs easier [10–12,64,67].
4. An infinite population is used, or the expected value of the population distribution is calculated [11–13,68].

Assumptions 1 and 2 have a rather technical character, and 3 and 4 significantly restrict the applicability of results.

4.1. Diversity analyses

Zaharie [11] investigated the population diversity of DE/rand/1/bin in one dimension. For a given population $\mathbf{X} = \{x_1, \ldots, x_{N_\text{p}}\}$, mutants $\mathbf{U}$ are realizations of a random variable. Consequently, their empirical variance $\widehat{\text{Var}}(\mathbf{U})$ is also a one-dimensional random variable. Zaharie used probability theory to prove that the expected variance of individuals after mutation is given by

$$\text{E}\left(\widehat{\text{Var}}(\mathbf{U})\right) = \left(2F^2 + \frac{N_\text{p} - 1}{N_\text{p}}\right)\widehat{\text{Var}}(\mathbf{X}), \tag{13}$$

where $\widehat{\text{Var}}$ is an estimate of variance. The expectation of the empirical variance is taken with respect to all possible mutant populations generated by differential mutation. This expectation can be estimated with repetitive runs of the algorithm. After crossover between mutants and the parents, the expected variance of the offspring population is given by

$$\text{E}\left(\widehat{\text{Var}}(\mathbf{O})\right) = \left(1 + 2p_\text{m}F^2 - \frac{2p_\text{m}}{N_\text{p}} + \frac{p_\text{m}^2}{N_\text{p}}\right)\widehat{\text{Var}}(\mathbf{X}), \tag{14}$$

where p_m denotes the crossover probability that is defined for a general, n-dimensional search space as [63]:

$$p_\text{m} = \begin{cases} C_\text{r}\dfrac{n-1}{n} + \dfrac{1}{n} & \text{for binomial crossover,} \\ \dfrac{1 - C_\text{r}^n}{n \cdot (1 - C_\text{r})} & \text{for exponential crossover.} \end{cases} \tag{15}$$

For binomial crossover, the relationship between crossover probability p_m and crossover rate C_r is linear, whereas for exponential crossover

it is non-linear, especially in high-dimensional search spaces.

Relationships among the variances of parents, mutants, and offspring for several other DE variants are derived in papers [8,69] and summarized in article [70]. In particular, for DE/best/1, the expected variance after mutation and crossover reads:

$$\mathrm{E}\left(\widehat{\mathrm{Var}}(\mathbf{O})\right) = \left(1 + 2p_{\mathrm{m}}F^2 - p_{\mathrm{m}} - \frac{p_{\mathrm{m}}(1-p_{\mathrm{m}})}{N_{\mathrm{p}}}\right)\widehat{\mathrm{Var}}(\mathbf{X}) + p_{\mathrm{m}}(1-p_{\mathrm{m}})\frac{N_{\mathrm{p}}-1}{N_{\mathrm{p}}}\left(x_{\mathrm{best}} - \hat{\mathrm{E}}(\mathbf{X})\right)^2, \tag{16}$$

whereas for DE/either-or, the expected offspring variance is:

$$\mathrm{E}\left(\widehat{\mathrm{Var}}(\mathbf{O})\right) = \widehat{\mathrm{Var}}(\mathbf{X}) \cdot \left[p_{\mathrm{f}}^2 \cdot \left(1 + 2F^2 - \frac{1}{N_{\mathrm{p}}}\right) + 2p_{\mathrm{f}}(1-p_{\mathrm{f}})\left(\frac{N_{\mathrm{p}}-1}{N_{\mathrm{p}}} + F^2 + 3K^2 - 2K\right) + (1-p_{\mathrm{f}})^2\left(\frac{N_{\mathrm{p}}-1}{N_{\mathrm{p}}} + 2\frac{N_{\mathrm{p}}-2}{N_{\mathrm{p}}}(3K^2 - 2K)\right)\right]. \tag{17}$$

Results obtained by Zaharie show that for all considered DE variants, a linear relationship holds between the expected variances of offspring and the empirical variance of the population:

$$\mathrm{E}\left(\widehat{\mathrm{Var}}(\mathbf{O})\right) = c \cdot \widehat{\mathrm{Var}}(\mathbf{X}) + d, \tag{18}$$

where c and d are scalars [70].

Zaharie did not show how the variance of population changes after selection, i.e. after completing the whole iteration of the algorithm. Difficulties in computing this quantity are related primarily to its dependence on the objective function.

Formulas (13), (14), (16), and (17) directly depend on the population size N_{p}. For large populations, quantities $(N_{\mathrm{p}} - 1)/N_{\mathrm{p}}$ and $1/N_{\mathrm{p}}$ tend to 1 and 0 respectively. Thus the dependence weakens towards insignificance, which simplifies the diversity equations. On the other hand, for low N_{p} values, the actual variances of the mutant and offspring populations may largely differ from their expectations. This is due to the poor efficiency of small sample statistics.

Zaharie argues that her one-dimensional analysis generalizes to multi-dimensional cases because all elements of vectors that encode individuals are evolved according to the same rule, hence analysis can be conducted componentwise [71]. Therefore, the expected values and variances transform to the expectation vectors and covariance matrices in higher-dimensional spaces.

Zaharie and Micota [70] generalized the variance formulas for constrained optimization. Offspring falling outside of the feasible range for any dimension were randomly reinitialized. This led to a mixture of a trimmed offspring distribution and a uniform one.

4.2. Critical regions for parameters

Formulas describing population diversity give insight into DE operators and provide bounds for setting the parameters. For instance, comparing the population variance before and after applying mutation and crossover allows for determining the critical values of the parameters for which the offspring population will have lower diversity than the parent population. Such a diversity loss without imposing selective pressure makes DE prone to premature convergence for every objective function [12]. For DE/rand/1/bin, equating the empirical variance of population $\widehat{\mathrm{Var}}(\mathbf{X})$ to the expected empirical variance of the offspring $\mathrm{E}\left(\widehat{\mathrm{Var}}(\mathbf{O})\right)$ yields

$$2F^2 p_{\mathrm{m}} - \frac{2p_{\mathrm{m}}}{N_{\mathrm{p}}} + \frac{p_{\mathrm{m}}^2}{N_{\mathrm{p}}} = 0. \tag{19}$$

Equation (19) allows for the calculation of the critical value of the scaling factor F_{crit} below which premature convergence is inevitable:

$$F_{\mathrm{crit}} = \sqrt{\frac{1 - p_{\mathrm{m}}/2}{N_{\mathrm{p}}}}. \tag{20}$$

The critical regions were further analysed by Zaharie and Micota in paper [70] and presented as several plots in the $F - C_{\mathrm{r}}$ domain for a few DE variants. To avoid undesired behaviour, such as the premature collapse of population diversity, parameters have to be chosen outside of the critical regions.

Zaharie [8] equated the expected variances of the DE/either-or operator, described by formula (6), for two different p_{f} settings. The resulting equation was solved with respect to K. To make the operator independent on the choice of p_{f}, values of K had to be very close to the suggested dependence $K = (F + 1)/2$. This result provides a theoretical background to an empirically-derived rule.

4.3. Binomial and exponential crossover

The crossover operator in DE was investigated by Zaharie [63,65]. Her comparison of the two most common methods, binomial DE/X/Y/bin and exponential DE/X/Y/exp was based on the crossover probability p_{m} provided in equation (15). She concluded that the difference in performance of these operators mainly follows from the number of elements exchanged between the vectors that encode the offspring and the parent.

In high-dimensional search spaces, the impact of the crossover rate on crossover probability strongly depends on the operator used [63]. For binomial crossover, the difference between p_{m} and C_{r} is small and vanishes completely if the individuals taking part in DE are not required to be pairwise distinct. These parameters are therefore approximately equal $p_{\mathrm{m}} \approx C_{\mathrm{r}}$. The expected number of exchanged elements is $np_{\mathrm{m}} = (n-1)C_{\mathrm{r}} + 1$.

In the exponential crossover operator, crossover probability depends non-linearly on the crossover rate. The enumerator in formula (15) quickly vanishes due to the exponentiation of $C_{\mathrm{r}} \in [0,1]$. The expected number of exchanged elements is $np_{\mathrm{m}} = (1 - C_{\mathrm{r}}^n)/(1 - C_{\mathrm{r}})$. Consequently, exponential crossover is sensitive only to a very small range of C_{r} values, such as $C_{\mathrm{r}} \in [0.9, 1]$. This range narrows further for higher search space dimensions n. In a limit, when $n \to \infty$, mutation probability is zero for all crossover rates smaller than one [63]:

$$\lim_{n\to\infty} p_{\mathrm{m}} = \begin{cases} C_{\mathrm{r}} & \text{for binomial crossover,} \\ 0 & \text{for exponential crossover and } 0 \le C_{\mathrm{r}} < 1, \\ 1 & \text{for exponential crossover and } C_{\mathrm{r}} = 1. \end{cases} \tag{21}$$

These results show that the crossover rate $C_{\mathrm{r}} \approx 1$ is necessary for exponential crossover to avoid very low crossover probabilities p_{m} in high-dimensional search spaces.

4.4. Generalized scaling factors

Opara and Arabas [66] used the Central Limit Theorem to show that the distribution of a sum of k difference vectors (2) weakly converges for $k \to \infty$ to a multivariate Gaussian distribution. Following the approach by Zaharie [11], their analysis was based on the expectation vector and the covariance matrix of the mutant population. These statistics, which determine Gaussian distribution, are given as

$$\mathrm{E}\left(\hat{\mathrm{E}}(\mathbf{U})\right) = \mathbf{x}_{\mathrm{base}}, \tag{22}$$

$$\mathrm{E}\left(\widehat{\mathrm{Cov}}(\mathbf{U})\right) = \left(1 + 2\sum_{j=1}^{k} F_j^2\right) \cdot \widehat{\mathrm{Cov}}(\mathbf{X}), \tag{23}$$

where $\widehat{\mathrm{Cov}}(\mathbf{X})$ denotes the empirical covariance matrix of the current population. Weak convergence means that for large k differential mutation (2) can be approximated with Gaussian mutation:

$$\mathbf{u}_i \leftarrow \mathbf{x}_{r_1} + \sqrt{2k}F \cdot \mathbf{q}_\infty, \text{ where } \mathbf{q}_\infty \sim \mathcal{N}\left(\mathbf{0}, \widehat{\mathrm{Cov}}(\mathbf{X})\right). \tag{24}$$

For many DE variants, the expected covariance matrix of mutants is proportional to the covariance matrix of the parent population, i.e. $d = 0$ or $d \approx 0$ in equation (18). Consequently, DE mutation operators differ in the mutation range but keep the same search directions. This is described by a generalized scaling factor $g(F)$:

$$\mathrm{E}\left(\widehat{\mathrm{Cov}}(\mathbf{U})\right) = g(F) \cdot \widehat{\mathrm{Cov}}(\mathbf{X}). \tag{25}$$

One can compensate for the differences among mutation ranges with appropriate modification of the scaling factor [72]. The generalized scaling factors show that many variants of differential mutation can be modelled with a base variant with an appropriately transformed, equivalent scaling factor. For instance, DE/current-to-best/1, defined by formula (5), corresponds to DE/best/1 with generalized scaling factor $F^{\mathrm{DE/best/1}} = \sqrt{\frac{1}{2}[(1-F_1)^2 + 2F_2^2]}$ and to DE/rand/1 with $F^{\mathrm{DE/rand/1}} = \sqrt{\frac{1}{2}[(1-F_1)^2 + 2F_2^2 - 1]}$.

The generalized scaling factors for several differential mutation operators are listed in a table in paper [72]. This allows for a common, synthetic analysis of many DE variants. Moreover, the choice of a particular mutation operator does not change the optimization performance much when the generalized scaling factor is kept at the same value. This effect is particularly well observed for mutation operators using a larger number of random elements, such as DE/rand/2.

5. Population dynamics models

Evolutionary algorithms are often modeled as Markov chains [38,73,74], which enables the population diversity to be analysed in consecutive iterations. Most DE variants, including the classical one, satisfy the Markov property, i.e. the distribution of the next population depends only on the current population rather than on a series of the previous ones.

It is convenient to assume an infinite population that is spread over the search space, as it enables the use of calculus [75–77]. In this case, the global minimum is already found in the first iteration. Consequently, properties such as "convergence speed" must be redefined. This is typically done through analysing statistics that describe population distribution, such as its expectation vector or covariance matrix. Convergence is not understood as hitting the neighbourhood of the global optimum, but rather as the growth of sampling density in that region.

The infinite population model describes the expected location of the population midpoint, which does not necessarily coincide with observations obtained for a single run of the algorithm with a finite population. Consider a bimodal, symmetric objective function. With a low mutation range, the whole population (and its mean) converges to either of the optima. However, the expected population distribution is bimodal, with a mean located halfway between the optima [78]. The infinite population model does not describe a single run of the algorithm but rather an average of many independent runs.

5.1. Dynamics of a one-dimensional population

The mutant distribution is described by a sum of random variables corresponding to choosing individuals that take part in differential mutation. Ali [79] derived the probability density of the mutant distribution for DE. He used direct integration for a uniformly distributed population in a one-dimensional search space based on an infinite population model. Wang and Huang [67] advanced Ali's results by analysing crossover and computing the distribution of the next population for a monotonous objective function. They also provided illustrative examples showing how the population distribution changes over the first iteration.

Ali and Fatti [80] approximated differential mutation with beta distribution, which allowed them to avoid generating mutants outside the box constraints. Wang and Huang [67] explicitly analysed the projection of unfeasible solutions on the box constraints. This aspect is rarely considered in theoretical analyses of DE, despite its significant influence on optimization performance [81].

Xue et al. [82,83] used the DE/current-to-best/k operator, defined by equation (5), for multi-objective optimization. They provided a population diversity model based on componentwise Gaussian approximation of differential mutation. Next, they developed population dynamics formulas that were greatly simplified because the analysed algorithm did not use a selection operator. The populations in consecutive iterations were approximately Gaussian. The following condition on the parameters is necessary to avoid premature convergence in this model:

$$2kF^2 + (1-\lambda) > 1. \tag{26}$$

Inequality (26) was derived from a condition describing the exponential vanishing of the population covariance matrix without selective pressure.

5.2. Differential Evolution as a dynamic system

Dasgupta et al. [10,64] described the population of DE/rand/1/bin in a one-dimensional search space as a dynamic system. They used notions from probability theory and calculus. In their approach, each individual was treated as a particle moving with a certain velocity. The iterative character of DE was interpreted as a discrete approximation of continuous time. Their analysis was performed in the neighbourhood of the minimum, assuming that the objective function was twice differentiable and did not change too rapidly (i.e. fulfilled the Lipschitz condition). In this case, the expected velocity of individual x_i^t in iteration t could be approximated with

$$\mathrm{E}\left(\frac{\mathrm{d}x_i^t}{\mathrm{d}t}\right) \approx -\frac{k}{8}C_\mathrm{r} \cdot \left((2F^2+1)\widehat{\mathrm{Var}}(\mathbf{X}^t) + (x_{\mathrm{mean}}^t - x_i^t)^2\right)\frac{\mathrm{d}f(x_i^t)}{\mathrm{d}x} + \frac{1}{2}C_\mathrm{r} \cdot (x_{\mathrm{mean}}^t - x_i^t), \tag{27}$$

where $\frac{\mathrm{d}f(x)}{\mathrm{d}x}$ is a spatial differential of the objective function, and $\widehat{\mathrm{Var}}(\mathbf{X}^t)$ denotes the empirical variance of population distribution. The difference between the population midpoint and the current individual $x_{\mathrm{mean}}^t - x_i^t$ appearing in formula (27) is a symptom of the selection bias.

In the above work, a step function modelling "greedy selection" was approximated with a logistic, sigmoidal function. This was then substituted with a linear term from the Maclaurin series. Therefore, in equation (27), factor k appeared, which is a "moderate value" of the constant from the exponent of the logistic function. These transformations implicitly modify the replacement mechanism in DE that lowers the selective pressure. This introduces a possibility that a weaker offspring replaces a stronger parent.

Dasgupta et al. [10,64] underlined the similarity between the form of equation (27) and the gradient descent optimization described with formula

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -a\frac{\mathrm{d}f(x)}{\mathrm{d}x} + b, \tag{28}$$

where a is the learning rate and b is the momentum. In equation (27), the learning rate would be

$$a_{\mathrm{DE}} = \frac{k}{8}C_\mathrm{r} \cdot \left((2F^2+1)\widehat{\mathrm{Var}}(\mathbf{X}^t) + (x_{\mathrm{mean}}^t - x_i^t)^2\right), \tag{29}$$

and momentum

$$b_{\mathrm{DE}} = \frac{1}{2}C_\mathrm{r} \cdot (x_{\mathrm{mean}}^t - x_i^t). \tag{30}$$

The term $-a_{\mathrm{DE}}\frac{\mathrm{d}f(x_i^t)}{\mathrm{d}x}$ is responsible for guiding the population in the direction of the decreasing gradient, whereas b_{DE} represents velocity towards the population midpoint. In contrast to the classical gradient descent method, quantities a_{DE} and b_{DE} are not constant but depend on the population distribution. To some extent, this distorts the analogy with gradient decent, which can be noted in a reformulation of equation (27)

$$\frac{\mathrm{d}x_i^t}{\mathrm{d}t} = -a_{\mathrm{DE}}(x_i^t, \mathbf{X}^t)\frac{\mathrm{d}f(x_i^t)}{\mathrm{d}x} + b_{\mathrm{DE}}(x_i^t, x_{\mathrm{mean}}^t). \tag{31}$$

DE is invariant under monotonic (order-preserving) transformations of the objective function. Weakening this assumption through modelling selection with a linear approximation of the logistic function implicitly changes the selection mechanism from elitist to quasi-proportional. This allows the mechanisms of DE to be highlighted. Informally, these mechanisms are described as performing "gradient optimization without computing gradient" and adjusting the speed of DE using the spread of the population.

5.3. Dynamics of infinite population

For a model of infinite population whose distribution in iteration t is described with the probability density function (PDF) $h^t(\mathbf{x})$, Ghosh et al. [68] proved that the PDF of mutants $h_{\mathrm{m},i}^t(\mathbf{x})$ corresponding to the i-th parent is given by

$$h_{\mathrm{m},i}^t(\mathbf{x}) = \frac{1}{F^2 P_{3,N_{\mathrm{p}}-1}} \sum_{\substack{i_1 \\ i_1 \neq i_2 \neq i_3 \neq i}} \sum_{i_2} \sum_{i_3} h_{i_1}^t(\mathbf{x}) * h_{i_2}^t\left(\frac{\mathbf{x}}{F}\right) * h_{i_3}^t\left(-\frac{\mathbf{x}}{F}\right), \tag{32}$$

where constant $P_{3,N_{\mathrm{p}}-1}$ denotes the number of permutations of three elements from $N_{\mathrm{p}} - 1$, and symbol $*$ indicates n-dimensional convolution.

Ghosh et al. [68] also introduced a special notation to describe the result of a particular crossover execution c. Indices of elements originating from either mutant or parent are denoted as $\mathbf{a}_c$ and $\mathbf{b}_c$. Notation $\mathbf{x}_{\mathbf{a}_c}, \mathbf{y}_{\mathbf{b}_c}$ describes a vector formed from elements of vector $\mathbf{x}$ in positions $\mathbf{a}_c$ and vector $\mathbf{y}$ in positions $\mathbf{b}_c$. For instance, for a five-dimensional search space and values

$$\mathbf{x} = [x_1, x_2, x_3, x_4, x_5]^T, \tag{33}$$

$$\mathbf{y} = [y_1, y_2, y_3, y_4, y_5]^T, \tag{34}$$

$$\mathbf{a}_c = \{1, 2, 4\}, \tag{35}$$

$$\mathbf{b}_c = \{3, 5\}, \tag{36}$$

we obtain

$$\mathbf{x}_{\mathbf{a}_c}, \mathbf{y}_{\mathbf{b}_c} = [x_1, x_2, y_3, x_4, y_5]^T. \tag{37}$$

This notation facilitates the common analysis of binomial and exponential crossover. Consider a particular crossover execution in which elements with indices $\mathbf{a}_c$ originate from the mutant and elements with indices $\mathbf{b}_c$ from the parent. To generate an offspring at point $\mathbf{o} = \mathbf{o}_{\mathbf{a}_c}, \mathbf{o}_{\mathbf{b}_c}$ we need a mutant with elements of $\mathbf{o}$ at coordinates $\mathbf{a}_c$ and a parent with elements of $\mathbf{o}$ at coordinates $\mathbf{b}_c$. The remaining coordinates do not influence the offspring. Therefore, the PDF of an offspring h_{o}^t is given by an integral of a product of PDFs of the mutant $h_{\mathrm{m}}^t(\mathbf{o}_{\mathbf{a}_c}, \mathbf{u}_{\mathbf{b}_c})$ and the parent $h^t(\mathbf{y}_{\mathbf{a}_c}, \mathbf{o}_{\mathbf{b}_c})$:

$$h_{\mathrm{o}}^t\left(\mathbf{o}_{\mathbf{a}_c}, \mathbf{o}_{\mathbf{b}_c}\right) = \int_{\mathbb{R}^{|\mathbf{a}_c|} \times \mathbb{R}^{|\mathbf{b}_c|}} h_{\mathrm{m}}^t\left(\mathbf{o}_{\mathbf{a}_c}, \mathbf{u}_{\mathbf{b}_c}\right) h^t\left(\mathbf{y}_{\mathbf{a}_c}, \mathbf{o}_{\mathbf{b}_c}\right) \mathrm{d}\mathbf{y}_{\mathbf{a}_c}\, \mathrm{d}\mathbf{u}_{\mathbf{b}_c}, \tag{38}$$

where $|\mathbf{a}_c|$ and $|\mathbf{b}_c|$ denote the number of elements originating from the mutant and the parent, $|\mathbf{a}_c| + |\mathbf{b}_c| = n$. Variables $\mathbf{y}_{\mathbf{a}_c}$ and $\mathbf{u}_{\mathbf{b}_c}$ denote the irrelevant elements of the vectors encoding the parent and the mutant, which are integrated out.

The choice of an individual that survives to the next iteration depends on the comparison of the objective function values for the parent and the offspring. This can be described through integration over the level sets i.e. the regions for which the value of the objective function is better (or worse) than for the offspring $f(\mathbf{x}) = f(\mathbf{x}_{\mathbf{a}_c}, \mathbf{x}_{\mathbf{b}_c}) \geq f(\mathbf{o}_{\mathbf{a}_c}, \mathbf{x}_{\mathbf{b}_c})$. In this way, the population PDF in the next iteration becomes:

$$h_{p_c}^{t+1}(\mathbf{x}) = h^t(\mathbf{x}) \int_{f(\mathbf{x}) < f(\mathbf{o}_{\mathbf{a}_c}, \mathbf{x}_{\mathbf{b}_c})} h_{\mathrm{m}}^t(\mathbf{o}_{\mathbf{a}_c}, \mathbf{u}_{\mathbf{b}_c})\, \mathrm{d}\mathbf{o}_{\mathbf{a}_c}\, \mathrm{d}\mathbf{u}_{\mathbf{b}_c} + \int_{f(\mathbf{x}) \geq f(\mathbf{y}_{\mathbf{a}_c}, \mathbf{x}_{\mathbf{b}_c})} h_{\mathrm{m}}^t(\mathbf{x}_{\mathbf{a}_c}, \mathbf{u}_{\mathbf{b}_c}) h^t(\mathbf{y}_{\mathbf{a}_c}, \mathbf{x}_{\mathbf{b}_c})\, \mathrm{d}\mathbf{y}_{\mathbf{a}_c}\, \mathrm{d}\mathbf{u}_{\mathbf{b}_c}. \tag{39}$$

The first summand in equation (39) describes a situation in which the offspring is rejected in selection and its parent gets to the next population. The integral describes the probability of such an event. The second integral denotes the opposite case: the offspring generated in point $\mathbf{x} = \mathbf{x}_{\mathbf{a}_c}, \mathbf{x}_{\mathbf{b}_c}$ wins the competition against the parent $\mathbf{y}_{\mathbf{a}_c}, \mathbf{x}_{\mathbf{b}_c}$.

The derivation of formula (39) was conducted for a fixed choice of indices $\mathbf{a}_c$ and $\mathbf{b}_c$. To describe the general case, Ghosh et al. [68] summed density (39) over all possible executions of the crossover operator using the probability of particular crossover outcomes p_c as weights:

$$h^{t+1}(\mathbf{x}) = \sum_{c=1}^{2^n-1} p_c h_{p_c}^{t+1}(\mathbf{x}). \tag{40}$$

The summation runs over all $2^n - 1$ possible realizations of the crossover operator (at least one element was required to come from the mutant). Hence, the model's computational complexity grows exponentially with the search space dimension n, which significantly constrains the scope of its application.

After the substitution of equation (39) into (40) the final dynamics formula becomes:

$$h^{t+1}(\mathbf{x}) = \sum_{c=1}^{2^n-1} p_c \left(h^t(\mathbf{x}) \int_{f(\mathbf{x}) < f(\mathbf{o}_{\mathbf{a}_c}, \mathbf{x}_{\mathbf{b}_c})} h_{\mathrm{m}}^t(\mathbf{o}_{\mathbf{a}_c}, \mathbf{u}_{\mathbf{b}_c})\, \mathrm{d}\mathbf{o}_{\mathbf{a}_c}\, \mathrm{d}\mathbf{u}_{\mathbf{b}_c} + \int_{f(\mathbf{x}) \geq f(\mathbf{y}_{\mathbf{a}_c}, \mathbf{x}_{\mathbf{b}_c})} h_{\mathrm{m}}^t(\mathbf{x}_{\mathbf{a}_c}, \mathbf{u}_{\mathbf{b}_c}) h^t(\mathbf{y}_{\mathbf{a}_c}, \mathbf{x}_{\mathbf{b}_c})\, \mathrm{d}\mathbf{y}_{\mathbf{a}_c}\, \mathrm{d}\mathbf{u}_{\mathbf{b}_c} \right). \tag{41}$$

Equation (41) links the population distribution in subsequent iterations. The influence of the mutation operator is described by the integrands, selection is depicted in the regions of integration, and crossover is modelled by an exhaustive sum.

5.4. Convergence analysis with Lyapunov's second method

Modelling the population in DE as a dynamic system allows its stability and convergence to be investigated using Lyapunov's second method. This method provides the conditions which ensure that a system with constantly decreasing energy will reach the zero energy state identified by a stationary point. The main advantage of this method is the possibility of analysing the stability of the solution without solving the dynamics equations.

Dasgupta et al. [64] used Lyapunov's second method to show that a population located close to an isolated optimum approaches it in a stable way and without any kind of oscillatory behaviour. These results were further developed in a paper by Ghosh et al. [68]. Derivation of the Lyapunov function was based on the population dynamics model detailed in Section 5.3.

Ghosh et al. [68] did not solve equation (41). Instead, they used it to derive a non-constructive proof of the convergence of DE. They defined a Lyapunov function as the difference between the expected values of

the objective function with respect to measures given by population distributions in two consecutive populations. Next, they showed that this function equals zero in the global minimum and is positive for all other arguments. Its changes between consecutive iterations, which are an approximation of a time derivative, decrease during the execution of the algorithm. Lyapunov's second method ensures convergence of the population to a one-point distribution located in the global minimum.

The above theoretical result seems to contradict experimentally observed cases in which DE does not find the global minimum but converges to a local one. Ghosh et al. [68] use an important additional assumption of an infinite population initialized in an area containing the global minimum. Therefore, for all individuals outside of the neighbourhood of the minimum, there is a non-zero probability that an offspring falls there and then replaces the parent. The theorem proved in paper [68] means that infinite population distribution almost surely converges to the global minimum, which is hit already in the first iteration.

Similar results are derivable with the standard probability theory. However, use of Lyapunov's second method potentially opens new possibilities. For instance, one can analyse convergence speed through investigating the values of the Lyapunov function in consecutive iterations [68].

5.5. Gaussian approximation

Zhang and Sanderson devoted a chapter in their monograph [13] to analysing the dynamics of DE/rand/k/none (for $C_\mathrm{r} = 1$, so that mutants directly become offspring). They considered an n-dimensional, quadratic objective function. Without the loss of generality, it was assumed that the population midpoint was located on the first coordinate axis. This property held for the expected distributions in consecutive populations.

It was also assumed that the population in each iteration $t = 1, 2, 3, \ldots$ had uncorrelated, multivariate Gaussian distribution

$$\mathbf{X}^t \sim \mathcal{N}(\mathbf{x}^t_{\mathrm{mean}}, \mathbf{C}^t), \tag{42}$$

with expectation vector $\mathbf{x}^t_{\mathrm{mean}} = [m^t, 0, \ldots, 0]^T$ and diagonal covariance matrix

$$\mathbf{C}^t = \begin{bmatrix} v^t_1 & 0 & \cdots & 0 \\ 0 & v^t_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & v^t_2 \end{bmatrix}. \tag{43}$$

Gaussian approximation (42) accurately represents the axial symmetry of the considered case, but it neglects the skewness of the distribution of the population approaching the minimum.

Zhang and Sanderson derived equations describing the parameters of the expected population distribution in iteration $t+1$ based on their values in iteration t, which constituted a system:

$$\left(m^{t+1}\right)^2 = \left(m^t - \Delta m^t\right)^2 + \frac{v_1^{t+1} + (n-1)v_2^{t+1}}{N_\mathrm{p}}, \tag{44}$$

$$v_1^{t+1} = \frac{\pi - 1}{\pi}\frac{v_1^t + w_1^t}{2}, \tag{45}$$

$$v_2^{t+1} = v_2^t \Phi\left(-\frac{m^t_-}{\sqrt{v^t_+}}\right) + w_2^t \Phi\left(\frac{m^t_-}{\sqrt{v^t_+}}\right) + \tag{46}$$

$$- \frac{2\left(\left(v_2^t\right)^2 + \left(w_2^t\right)^2\right)^2}{\sqrt{2\pi v^t_+}}\left(-\frac{(m^t_-)^2}{2v^t_+}\right),$$

where Φ denotes the cumulative distribution function (CDF) of the standard normal distribution, w_1^t and w_2^t describe variances of the mutant population

$$w_1^t = (1 + 2kF^2)v_1^t, \tag{47}$$

$$w_2^t = (1 + 2kF^2)v_2^t, \tag{48}$$

and symbols Δm^t, m^t_- and v^t_+ depict the following quantities:

$$\Delta m^t = \frac{2m^t\left(v_1^t + w_1^t\right)}{\sqrt{2\pi}\sqrt{4(m^t)^2\left(v_1^t + w_1^t\right) + 2(n-1)\left(\left(v_2^t\right)^2 + \left(w_2^t\right)^2\right)}} \cdot \tag{49}$$

$$\cdot exp\left(-\frac{(n-1)^2\left(v_2^t - w_2^t\right)^2}{8(m^t)^2\left(v_1^t + w_1^t\right) + 4(n-1)\left(\left(v_2^t\right)^2 + \left(w_2^t\right)^2\right)}\right), \tag{49}$$

$$m^t_- = (n-1)(v_2^t - w_2^t) + (v_1^t - w_1^t), \tag{50}$$

$$v^t_+ = 4\left(m^t\right)^2(v_1^t + w_1^t) + 2\left(\left(v_1^t\right)^2 + \left(w_1^t\right)^2\right) \tag{51}$$

$$+ 2(n-1)\left(\left(v_2^t\right)^2 + \left(w_2^t\right)^2\right).$$

Equations (44)–(46) describe the elements of the expectation vector and covariance matrix of the expected population distribution. This model adequately describes the dynamics of large populations in which the empirical distribution is close to the expected distribution. Based on a simulation study, Zhang and Sanderson [13] conjectured that after a certain time, the rate of change of parameters of the Gaussian distribution stabilizes.

The results from Zhang and Sanderson allow practitioners to compute the critical value of the scaling factor F, below which the undesirable phenomenon of premature convergence occurs. Far away from the minimum, the quadratic objective function locally resembles a hyperplane and the critical value is $F = \sqrt{1/(\pi - 1)} \approx 0.683$, whereas in the neighbourhood of the minimum it drops to $F = 0.48$.

5.6. DE and Markov chain Monte Carlo

Ter Braak [84] introduced DE into a framework of Markov chain Monte Carlo methods. The goal was to draw samples with a density proportional to the objective function values rather than perform standard optimization. The objective function was assumed to be the density of some continuous probability distribution, i.e. a non-negative function with a unit integral.

The DE-MC algorithm [84] differs from classical DE in two respects. First, differential mutation uses the parent as the base vector and adds a realization of a normal random variable:

$$\mathbf{u}_i \leftarrow \mathbf{x}_i + F \cdot \left(\mathbf{x}_{r_1} - \mathbf{x}_{r_2}\right) + \mathbf{e}, \text{ where } \mathbf{e} \sim \mathcal{N}(\mathbf{0}, b \cdot \mathbf{I}), \tag{52}$$

where b is a small positive scalar and $\mathbf{I}$ is an identity matrix.

Second, greedy selection between the parent $\mathbf{x}_i$ and offspring $\mathbf{o}_i = \mathbf{u}_i$ is replaced with the Metropolis acceptance rule. The replacement probability equals

$$p_\mathrm{r} = \begin{cases} \min\left(\dfrac{f(\mathbf{u}_i)}{f(\mathbf{x}_i)}, 1\right) & \text{if } f(\mathbf{x}_i) > 0, \\ 1 & \text{if } f(\mathbf{x}_i) = 0. \end{cases} \tag{53}$$

The DE-MC algorithm yields a Markov chain, which has a unique stationary distribution with density f. In the proof, Ter Braak first shows

that the resulting chain is reversible and therefore the stationary distribution is proportional to f. To derive the uniqueness of the chain, the unboundedness of the support of $\mathbf{e}$ is necessary, so that the population can move between all of the local optima by means of macromutation.

From the Bayesian point of view, the significance of Ter Braak's results consists of the fact that the DE-MC algorithm adapts the direction and range of the sampling distribution and effectively accommodates heavy-tailed and multimodal target distributions [85].

The resulting population dynamics in DE-MC is quite different from other DE variants. Rather than converging to a single optimum, DE-MC maintains population diversity. The population traverses the whole feasible set, with sampling density proportional to the objective function. The resulting algorithm has an exploratory character and seems more appropriate for Monte Carlo sampling than for optimization.

6. Discussion and further study

6.1. Comparison of theoretical results on DE

In sections 4 and 5, we discussed theoretical results on the diversity and dynamics of the population in DE. Comparison of these models can be based on the analysis of their assumptions, the scope of applicability, obtained results and mathematical tools used. These criteria were used to summarize the current state of knowledge about population dynamics in DE in Table 1.

Most dynamics models focus on the classical DE/rand/1 variant. Sometimes, certain modifications are added, such as the introduction of soft selection. The latter makes it possible to prove global convergence. It also facilitates the use of calculus by smoothing the discontinuous step function describing greedy selection into a continuous, differentiable sigmoid.

Several of the theoretical models are derived in a one-dimensional space. This simplifies the necessary mathematical formalisms. However, crossover based on exchanging elements of vectors is trivialized in a one-dimensional space. Moreover, it is not possible to analyse the adaptation of search directions.

Many papers focus on the diversity of populations of parents, mutants, and offspring. Diversity is typically measured with the second central moment, i.e. variance (covariance matrix). Together with the population midpoint (expectation vector) we obtain parameters that uniquely characterize Gaussian distribution, which is another tool commonly used in theoretical analyses of DE.

There is still a gap between the results obtained on a theoretical basis and their practical use. However, it is worth mentioning approaches which have successfully bridged these two areas. Among the results that have influenced applications is Zaharie's analysis of critical values for the scaling factor and crossover rate. The developments of DE/either-or and BBDE by Price were also motivated by theoretical considerations. Recently, Arabas and Biedrzycki improved the efficiency of various EAs, including several types of DE, through the analysis of the midpoint dynamics [86].

6.2. Further study

Theoretical results concerning DE are still scarce and often of an abstract nature. This is clear from bibliometric analysis, which indicates huge disproportion between theory-oriented and experimental studies. Despite several valuable results obtained from previous studies, there are still many open issues within theoretical analyses of DE. Below we unfold some of the promising research directions.

1. The two features distinguishing DE from classical evolutionary algorithms are differential mutation and local, greedy selection. Although the former has been the focus of several studies, the latter seems largely unexplored. Direct modelling of selection in DE is a difficult task due to its elitist, local character and dependence on the objective function. Nevertheless, the impact of the selection mechanism on search dynamics is profound. Investigations into this issue could clarify whether DE resembles an adaptive swarm optimizer or rather a group of loosely related hill-climbers.
2. Population dynamics models for DE need further development. The ones currently available provide little operational advice. Moreover, the models are much more mathematically complicated than the original DE algorithm. Consequently, conclusions are less straightforward, which hinders their translation into new algorithmic developments. Population dynamics models are a possible way to compute statistics such as the expected first hitting time, i.e. the average time necessary to find an optimal solution.
3. Population size N_p has gained the least attention of all the control parameters of DE. In theoretical studies it is usually bypassed by analysing infinite population size. This assumption is sometimes implicitly introduced as a consequence of replacing the particular population characteristics with their expectations. The infinite size assumption is however poorly fulfilled in cases of small or dynamic populations, which often lead to improved performance [21]. It would be interesting to generalize the population diversity and dynamics models to explicitly handle the population size.
4. Over consecutive iterations, population distribution in DE adapts to the local shape of the optimized function. This phenomenon is known as "contour fitting" and plays a significant role in the optimization of highly-conditioned functions. Despite common empirical evidence, "contour fitting" is still waiting for theoretical grounding.
5. The formal analysis is best suited to simple algorithms with a clear, coherent internal structure. Classical DE is only partially characterized by these qualities because recombination is present in both mutation and crossover. Moreover, some variants of dif-

Table 1
Comparison of population dynamics models in DE.

Author	DE variant	Main results	Assumptions and techniques
Zaharie [11,63,70]	DE/rand/1/*Z*	Population diversity after mutation and crossover, critical regions for parameters	Analysis of the moments of population distribution
Zhang and Sanderson [13]	DE/rand/*k*	Dynamics for the radial objective function	Gaussian distribution of populations, geometrical approach
Dasgupta et al. [10,64]	DE/rand/1 soft selection	Analogy with gradient descent, stability near the optimum	Dynamic system, non-elitist selection
Ghosh et al. [68]	DE/rand/1	General dynamics equation, asymptotic global convergence	Dynamic system, non-elitist selection, Lyapunov's second method for stability
Ter Braak [84,85]	DE-MC soft selection	Characterization of a stationary population distribution over the feasible set	Existence and uniqueness of a stationary Markov chain
Xue et al. [82,83]	Multiobj. DE/current-to-best/*k*	Constraints on the critical region of parameters	No selection, componentwise Gaussian approximation of mutation

ferential mutation such as DE/best not only introduce diversity but also exert selective pressure. The state-of-the-art modifications of DE contain additional, non-trivial elements. The complex form of an algorithm and its numerous variants make modelling troublesome. From the theoretical perspective it is advisable to look for minimalistic, analysis-friendly variants of DE. Low complexity facilitates theoretical studies and allows for stronger conclusions with clearer interpretations to be drawn. Algorithmic simplicity does not necessarily hinder performance [31,32]. The search for a general, synthetic framework may concern not only whole algorithms but also their elements, such as genetic operators.

6. DE would benefit from transferring theoretical results developed for other metaheuristic optimization methods, such as Evolution Strategies or Estimation of Distribution Algorithms. Adaptation of these results would also show similarities and contrasts between DE and other global optimizers.
7. It would be useful to characterize the properties of optimization problem classes that would be particularly easy, or particularly hard, to be solved by DE. The existence of such classes of tasks is a consequence of the *no free lunch* theorem. Although some guidelines in this area are available [87], they are of experimental origin and have a general, descriptive character. These guidelines would benefit from being grounded in mathematical derivation. However, it is not clear to what extent it is feasible to formalize non-trivial features such as deceptiveness or a lack of clear structure [88]. Apart from the algorithm as a whole, guidelines could also address the use of particular elements of DE, for instance the choice of appropriate mutation operators.

7. Conclusions

The theory of DE is still behind empirical developments and to a larger extent explains already known phenomena rather than guides new research or provides practical advice. This situation is typical for evolutionary algorithms. The importance of theoretical studies lies in the possibility to better understand the principles and mechanics of the optimizer. Analytical results often inspire improvements and modifications to the algorithm. Finally, well-developed theoretical foundations build trust in the analysed methods.

This paper surveyed the theoretical results concerning DE. It provided an overview of the convergence studies, invariances, as well as investigations into differential mutation, crossover operators, and population diversity. A detailed presentation of the population dynamics models stressed their advantages and weaknesses. Despite huge research effort into DE, there are still many open problems waiting for analytical study, the most promising of which have been unfolded in this paper.

Acknowledgment

Karol Opara's study was supported by a research fellowship within "Information technologies: research and their interdisciplinary applications", agreement number POKL.04.01.01-00-051/10-00.

References

[1] F. Neri, V. Tirronen, Recent advances in Differential Evolution: a survey and experimental analysis, Artif. Intell. Rev. 33 (1–2) (2010) 61–106, https://doi.org/10.1007/s10462-009-9137-2.
[2] N. Pham, A. Malinowski, T. Bartczak, Comparative study of derivative free optimization algorithms, IEEE Trans. Ind. Inf. 7 (4) (2011) 592–600, https://doi.org/10.1109/TII.2011.2166799.
[3] K. Price, Differential Evolution: a fast and simple numerical optimizer, in: Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American, 1996, pp. 524–527, https://doi.org/10.1109/NAFIPS.1996.534790.
[4] R. Storn, On the usage of Differential Evolution for function optimization, in: Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American, 1996, pp. 519–523, https://doi.org/10.1109/NAFIPS.1996.534789.
[5] R. Storn, K. Price, Differential Evolution — a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359, https://doi.org/10.1023/A%3A1008202821328.
[6] S. Das, P.N. Suganthan, Differential Evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31, https://doi.org/10.1109/TEVC.2010.2059031.
[7] S. Das, S.S. Mullick, P. Suganthan, Recent advances in Differential Evolution–an updated survey, Swarm Evol. Comput. 27 (2016) 1–30, https://doi.org/10.1016/j.swevo.2016.01.004.
[8] D. Zaharie, Differential Evolution: from theoretical analysis to practical insights, in: Proceedings of 19th International Conference on Soft Computing (Mendel), 2012, pp. 126–131.
[9] K.V. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: a Practical Approach to Global Optimization, Natural Computing Series, Springer, 2005, https://doi.org/10.1007/3-540-31306-0.
[10] S. Dasgupta, A. Biswas, S. Das, A. Abraham, The population dynamics of Differential Evolution: a mathematical model, in: IEEE Congress on Evolutionary Computation, CEC 2008. (IEEE World Congress on Computational Intelligence), 2008, pp. 1439–1446, https://doi.org/10.1109/CEC.2008.4630983.
[11] D. Zaharie, On the explorative power of Differential Evolution algorithms, Analele Universitatii din Timosoara XXXIX (2) (2001) 1–12.
[12] D. Zaharie, Critical values for the control parameters of Differential Evolution algorithms, in: Proceedings of MENDEL, 2002, pp. 62–67.
[13] J. Zhang, A.C. Sanderson, Adaptive Differential Evolution: a Robust Approach to Multimodal Problem Optimization, vol. 1, Springer, 2009, https://doi.org/10.1007/978-3-642-01527-4.
[14] Y. Zhou, W. Yi, L. Gao, X. Li, Analysis of mutation vectors selection mechanism in Differential Evolution, Appl. Intell. 44 (4) (2016) 904–912, https://doi.org/10.1007/s10489-015-0738-y.
[15] K.V. Price, Eliminating Drift Bias from the Differential Evolution Algorithm, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 33–88, https://doi.org/10.1007/978-3-540-68830-3_2.
[16] T. Bäck, D.B. Fogel, Z. Michalewicz (Eds.), Handbook of Evolutionary Computation, first ed., IOP Publishing Ltd., Bristol, UK, 1997.
[17] E.-N. Dragoi, V. Dafinescu, Parameter control and hybridization techniques in Differential Evolution: a survey, Artif. Intell. Rev. 45 (4) (2016) 447–470, https://doi.org/10.1007/s10462-015-9452-8.
[18] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in Differential Evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (6) (2006) 646–657, https://doi.org/10.1109/TEVC.2006.872133.
[19] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for Differential evolution, in: 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 71–78, https://doi.org/10.1109/CEC.2013.6557555.
[20] J. Brest, M. Sepesy Maučec, Population size reduction for the Differential Evolution algorithm, Appl. Intell. 29 (3) (2008) 228–247, https://doi.org/10.1007/s10489-007-0091-x.
[21] A. Piotrowski, Review of differential evolution population size, Swarm Evol. Comput. 32 (2017) 1–24, https://doi.org/10.1016/j.swevo.2016.05.003.
[22] A. Qin, V. Huang, P. Suganthan, Differential Evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417, https://doi.org/10.1109/TEVC.2008.927706.
[23] M. Epitropakis, D. Tasoulis, N. Pavlidis, V. Plagianakos, M. Vrahatis, Enhancing Differential Evolution utilizing proximity-based mutation operators, IEEE Trans. Evol. Comput. 15 (1) (2011) 99–119, https://doi.org/10.1109/TEVC.2010.2083670l.
[24] W. Gong, Z. Cai, Differential Evolution with ranking-based mutation operators, IEEE Trans. Cybern. 43 (6) (2013) 2066–2081, https://doi.org/10.1109/TCYB.2013.2239988.
[25] A. Zhang, J. Sanderson, JADE: adaptive Differential Evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.
[26] S. Das, A. Abraham, U. K. Chakraborty, A. Konar, Differential evolution using a neighborhood-based mutation operator, IEEE Transactions on evolutionary computation 13 (3).
[27] S. Rahnamayan, H. Tizhoosh, M. Salama, Opposition-based differential evolution, IEEE Trans. Evol. Comput. 12 (1) (2008) 64–79.
[28] A.P. Piotrowski, Adaptive memetic Differential Evolution with global and local neighborhood-based mutation operators, Inf. Sci. 241 (Supplement C) (2013) 164–194, https://doi.org/10.1016/j.ins.2013.03.060.
[29] A. Caponio, F. Neri, V. Tirronen, Super-fit control adaptation in memetic Differential Evolution frameworks, Soft Comput. 13 (8–9) (2009) 811–831, https://doi.org/10.1007/s00500-008-0357-1.
[30] J.A. Nelder, R. Mead, A simplex method for function minimization, Comput. J. 7 (4) (1965) 308–313.
[31] G. Iacca, F. Neri, E. Mininno, Y.-S. Ong, M.-H. Lim, Ockham's razor in memetic computing: three stage optimal memetic exploration, Inf. Sci. 188 (Supplement C) (2012) 17–43, https://doi.org/10.1016/j.ins.2011.11.025.
[32] A.P. Piotrowski, J.J. Napiórkowski, Some metaheuristics should be simplified, Inf. Sci. 427 (Supplement C) (2018) 32–62, https://doi.org/10.1016/j.ins.2017.10.039.

[33] K. Price, How symmetry constraints evolutionary optimizers: black Box Differential Evolution – a case study, in: Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), 2017, pp. 1712–1719, https://doi.org/10.1109/CEC.2017.7969508.
[34] N. Hansen, The CMA evolution strategy: a comparing review, in: J.A. Lozano, P. Larrańaga, I. Inza, E. Bengoetxea (Eds.), Towards a New Evolutionary Computation. Advances on Estimation of Distribution Algorithms, Springer Berlin Heidelberg, 2006, pp. 75–102, https://doi.org/10.1007/3-540-32494-1_4.
[35] F. Rothlauf, Representations for Genetic and Evolutionary Algorithms, Springer, 2006.
[36] A.E. Eiben, Z. Michalewicz, M. Schoenauer, J.E. Smith, Parameter control in evolutionary algorithms, in: Parameter Setting in Evolutionary Algorithms, Springer, 2007, pp. 19–46, https://doi.org/10.1007/978-3-540-69432-8_2.
[37] C.A. Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Meth. Appl. Mech. Eng. 191 (11) (2002) 1245–1287, https://doi.org/10.1016/S0045-7825(01)00323-1.
[38] H.-G. Beyer, H.-P. Schwefel, I. Wegener, How to analyse evolutionary algorithms, Theor. Comput. Sci. 287 (1) (2002) 101–130, https://doi.org/10.1016/S0304-3975(02)00137-8.
[39] A.E. Eiben, G. Rudolph, Theory of evolutionary algorithms: a bird's eye view, Theor. Comput. Sci. 229 (1) (1999) 3–9, https://doi.org/10.1016/S0304-3975(99)00089-4.
[40] K. Zielinski, R. Laur, Stopping criteria for Differential Evolution in constrained single-objective optimization, in: U.K. Chakraborty (Ed.), Advances in Differential Evolution, Springer, Berlin, Heidelberg, 2008, pp. 111–138, https://doi.org/10.1007/978-3-540-68830-3_4.
[41] K. Opara, J. Arabas, Benchmarking procedures for continuous optimization algorithms, J. Telecommun. Inf. Technol. 2011 (4) (2011) 73–80.
[42] K. Zielinski, D. Peters, R. Laur, Run time analysis regarding stopping criteria for Differential Evolution and particle swarm optimization, in: Proc. Of the 1st International Conference on Experiments/Process/System Modelling/Simulation/Optimization, 2005, pp. 1–8.
[43] G. Rudolph, Convergence of evolutionary algorithms in general search spaces, in: Proceedings of IEEE International Conference on Evolutionary Computation, IEEE, 1996, pp. 50–54, https://doi.org/10.1109/ICEC.1996.542332.
[44] Z. Hu, Q. Su, X. Yang, Z. Xiong, Not guaranteeing convergence of Differential Evolution on a class of multimodal functions, Appl. Soft Comput. 41 (Supplement C) (2016) 479–487, https://doi.org/10.1016/j.asoc.2016.01.001.
[45] Z. Hu, S. Xiong, Q. Su, Z. Fang, Finite Markov chain analysis of classical Differential Evolution algorithm, J. Comput. Appl. Math. 268 (2014) 121–134, https://doi.org/10.1016/j.cam.2014.02.034.
[46] Y.-J. Wang, J.-S. Zhang, Global optimization by an improved Differential evolutionary algorithm, Appl. Math. Comput. 188 (1) (2007) 669–680, https://doi.org/10.1016/j.amc.2006.10.021.
[47] N. Hansen, Invariance, self-adaptation and correlated mutations in evolution strategies, in: M. Schoenauer, K. Deb, G. Rudolph, E. Yao, Xinand Lutton, J.J. Merelo, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, 2000 Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, pp. 355–364, https://doi.org/10.1007/3-540-45356-3_35.
[48] S.M. Guo, C.C. Yang, Enhancing Differential Evolution utilizing eigenvector-based crossover operator, IEEE Trans. Evol. Comput. 19 (1) (2015) 31–49, https://doi.org/10.1109/TEVC.2013.2297160.
[49] D. Wolpert, W. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82, https://doi.org/10.1109/4235.585893.
[50] M. Köppen, D.H. Wolpert, W.G. Macready, Remarks on a recent paper on the "No Free Lunch" theorems, IEEE Trans. Evol. Comput. 5 (3) (2001) 295–296, https://doi.org/10.1109/4235.930318.
[51] A.J. Lockett, R. Miikkulainen, A probabilistic reformulation of no free lunch: continuous lunches are not free, Evol. Comput. 25 (3) (2017) 503–528, https://doi.org/10.1162/EVCO_a_00196.
[52] C. Igel, No Free Lunch Theorems: Limitations and Perspectives of Metaheuristics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 1–23, https://doi.org/10.1007/978-3-642-33206-7_1, Ch. 1.
[53] T. Weise, R. Chiong, K. Tang, Evolutionary optimization: pitfalls and booby traps, J. Comput. Sci. Technol. 27 (5) (2012) 907–936, https://doi.org/10.1007/s11390-012-1274-4.
[54] T. Weise, M. Zapf, R. Chiongand, A. Nebro, Why is optimization difficult? in: R. Chiong (Ed.), Nature-inspired Algorithms for Optimisation, Vol. 193 of Studies in Computational Intelligence, Springer, 2009, pp. 1–50, https://doi.org/10.1007/978-3-642-00267-0_1. Ch. 1.
[55] T. English, Optimization is easy and learning is hard in the typical function, in: Proceedings of the 2000 Congress on Evolutionary Computation: CEC 2000, 2000, pp. 924–931, https://doi.org/10.1109/CEC.2000.870741.
[56] W.B. Langdon, R. Poli, Evolving problems to learn about particle swarm optimizers and other search algorithms, IEEE Trans. Evol. Comput. 11 (5) (2007) 561–578, https://doi.org/10.1109/TEVC.2006.886448.
[57] J.H. Holland, Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, University of Michigan Press, 1975.
[58] D.E. Goldberg, et al., Genetic Algorithms in Search, Optimization, and Machine Learning, vol. 412, Addison-wesley Reading Menlo Park, 1989.
[59] L. Altenberg, The schema theorem and Price's theorem, in: Foundations of Genetic Algorithms, vol. 3, 1994, pp. 23–49.
[60] J. He, X. Yao, Drift analysis and average time complexity of evolutionary algorithms, Artif. Intell. 127 (1) (2001) 57–85, https://doi.org/10.1016/S0004-3702(01)00058-3.
[61] J. He, X. Yao, A study of drift analysis for estimating computation time of evolutionary algorithms, Nat. Comput. 3 (1) (2004) 21–35, https://doi.org/10.1023/B:NACO.0000023417.31393.c7.
[62] P.S. Oliveto, C. Witt, Simplified drift analysis for proving lower bounds in evolutionary computation, Algorithmica 59 (3) (2011) 369–386, https://doi.org/10.1007/s00453-010-9387-z.
[63] D. Zaharie, Influence of crossover on the behavior of Differential Evolution algorithms, Appl. Soft Comput. 9 (3) (2009) 1126–1138, https://doi.org/10.1016/j.asoc.2009.02.012.
[64] S. Dasgupta, S. Das, A. Biswas, A. Abraham, On stability and convergence of the population-dynamics in Differential Evolution, AI Commun. 22 (1) (2009) 1–20.
[65] D. Zaharie, A comparative analysis of crossover variants in Differential Evolution, in: International Multiconference on Computer Science and Information Technology, 2007, pp. 171–181.
[66] K. Opara, J. Arabas, Differential mutation based on population covariance matrix, in: R. Schaefer (Ed.), Parallel Problem Solving from Nature PPSN XI, Part I, Vol. 6238 of Lecture Notes in Computer Science, Springer, 2010, pp. 114–123, https://doi.org/10.1007/978-3-642-15844-5_12.
[67] L. Wang, F. Huang, Parameter analysis based on stochastic model for Differential Evolution algorithm, Appl. Math. Comput. 217 (7) (2010) 3263–3273, https://doi.org/10.1016/j.amc.2010.08.060.
[68] S. Ghosh, S. Das, A.V. Vasilakos, K. Suresh, On convergence of Differential Evolution over a class of continuous functions with unique global optimum, IEEE Trans. Syst. Man Cybern. B Cybern. 42 (1) (2012) 107–124, https://doi.org/10.1109/TSMCB.2011.2160625.
[69] E. Zhabitskaya, Constraints on control parameters of asynchronous Differential Evolution, in: G. Adam, J. Buša, M. Hnatič (Eds.), Mathematical Modeling and Computational Science: International Conference, MMCP 2011, Stará Lesná, Slovakia, July 4-8, 2011, Revised Selected Papers, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 322–327, https://doi.org/10.1007/978-3-642-28212-6_40.
[70] D. Zaharie, F. Micota, Revisiting the analysis of population variance in Differential Evolution algorithms, in: 2017 IEEE Congress on Evolutionary Computation (CEC), 2017, pp. 1811–1818, https://doi.org/10.1109/CEC.2017.7969521.
[71] D. Zaharie, Statistical properties of Differential Evolution and related random search algorithms, in: P. Brito (Ed.), COMPSTAT 2008: Proceedings in Computational Statistics, Physica-Verlag HD, Heidelberg, 2008, pp. 473–485, https://doi.org/10.1007/978-3-7908-2084-3_39.
[72] K. Opara, J. Arabas, Comparison of mutation strategies in Differential Evolution – a probabilistic perspective, Swarm Evol. Comput. 39 (2018) 53–69, https://doi.org/10.1016/j.swevo.2017.12.007.
[73] A.E. Nix, M.D. Vose, Modeling genetic algorithms with Markov chains, Ann. Math. Artif. Intell. 5 (1) (1992) 79–88, https://doi.org/10.1007/BF01530781.
[74] G. Rudolph, Finite Markov chain results in evolutionary computation: a tour d'horizon, Fundam. Inf. 35 (1) (1998) 67–89, https://doi.org/10.3233/FI-1998-35123405.
[75] X. Qi, 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 Network. 5 (1) (1994) 102–119, https://doi.org/10.1109/72.265965.
[76] I. Karcz-Duleba, Dynamics of infinite populations evolving in a landscape of uni and bimodal fitness functions, IEEE Trans. Evol. Comput. 5 (4) (2001) 398–409, https://doi.org/10.1109/4235.942533.
[77] I. Karcz-Duleba, Dynamics of two-element populations in the space of population states, IEEE Trans. Evol. Comput. 10 (2) (2006) 199–209, https://doi.org/10.1109/TEVC.2005.856070.
[78] J. Arabas, Approximating the genetic diversity of populations in the quasi-equilibrium state, IEEE Trans. Evol. Comput. 16 (5) (2012) 632–644, https://doi.org/10.1109/TEVC.2011.2166157.
[79] M. Ali, Differential Evolution with preferential crossover, Eur. J. Oper. Res. 181 (3) (2007) 1137–1147, https://doi.org/10.1016/j.ejor.2005.06.077.
[80] M.M. Ali, L.P. Fatti, A differential free point generation scheme in the Differential Evolution algorithm, J. Global Optim. 35 (4) (2006) 551–572, https://doi.org/10.1007/s10898-005-3767-y.
[81] J. Arabas, A. Szczepankiewicz, T. Wroniak, Experimental comparison of methods to handle boundary constraints in differential evolution, in: R. Schaefer, C. Cotta, J. Kołodziej, G. Rudolph (Eds.), Parallel Problem Solving from Nature, PPSN XI: 11th International Conference, Kraków, Poland, September 11-15, 2010, Proceedings, Part II, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 411–420, https://doi.org/10.1007/978-3-642-15871-1_42.
[82] F. Xue, A.C. Sanderson, R.J. Graves, Modeling and convergence analysis of a continuous multi-objective differential evolution algorithm, in: 2005 IEEE Congress on Evolutionary Computation, vol. 1, 2005, pp. 228–235, https://doi.org/10.1109/CEC.2005.1554689. vol. 1.
[83] F. Xue, A.C. Sanderson, R.J. Graves, Multi-objective differential evolution - algorithm, convergence analysis, and applications, in: 2005 IEEE Congress on Evolutionary Computation, vol. 1, 2005, pp. 743–750, https://doi.org/10.1109/CEC.2005.1554757. vol. 1.
[84] C.J. Ter Braak, A Markov chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces, Stat. Comput. 16 (3) (2006) 239–249, https://doi.org/10.1007/s11222-006-8769-1.

[85] J.A. Vrugt, C. Ter Braak, C. Diks, B.A. Robinson, J.M. Hyman, D. Higdon, Accelerating Markov chain Monte Carlo simulation by Differential Evolution with self-adaptive randomized subspace sampling, Int. J. Nonlinear Sci. Numer. Stimul. 10 (3) (2009) 273–290, https://doi.org/10.1515/IJNSNS.2009.10.3.273.

[86] J. Arabas, R. Biedrzycki, Improving evolutionary algorithms in a continuous domain by monitoring the population midpoint, IEEE Trans. Evol. Comput. 21 (5) (2017) 807–812, https://doi.org/10.1109/TEVC.2017.2673962.

[87] R. Storn, Real-world applications in the communications industry – when do we resort to Differential Evolution? in: 2017 IEEE Congress on Evolutionary Computation (CEC), 2017, pp. 765–772, https://doi.org/10.1109/CEC.2017.7969387.

[88] J. He, T. Chen, X. Yao, On the easiest and hardest fitness functions, IEEE Trans. Evol. Comput. 19 (2) (2015) 295–305, https://doi.org/10.1109/TEVC.2014.2318025.