Neural Computing and Applications (2020) 32:12363–12379
https://doi.org/10.1007/s00521-020-04832-8

REVIEW ARTICLE

Evolutionary algorithms and their applications to engineering problems

Adam Slowik[1] · Halina Kwasnicka[2]

Received: 27 November 2018 / Accepted: 5 March 2020 / Published online: 16 March 2020

Abstract
The main focus of this paper is on the family of evolutionary algorithms and their real-life applications. We present the following algorithms: genetic algorithms, genetic programming, differential evolution, evolution strategies, and evolutionary programming. Each technique is presented in the pseudo-code form, which can be used for its easy implementation in any programming language. We present the main properties of each algorithm described in this paper. We also show many state-of-the-art practical applications and modifications of the early evolutionary methods. The open research issues are indicated for the family of evolutionary algorithms.

Keywords Nature-inspired methods · Genetic algorithm · Genetic programming · Differential evolution · Evolution strategy · Evolutionary programming · Real-life applications

1 Introduction

These days, in the area of soft computing research we can observe a strong pressure to search for new optimization techniques which are based on nature. Figure 1 presents some approaches in optimization techniques with a concentration on evolutionary approaches. Today, the whole family of *evolutionary optimization algorithms* is referred to as *evolutionary computation* (EC) algorithms. In the *evolutionary computation* domain, we can mention the following main algorithms: the *genetic algorithm* (GA) [1], *genetic programming* (GP) [2], *differential evolution* (DE) [3], the *evolution strategy* (ES) [4], and *evolutionary programming* (EP) [5]. Each of these techniques has many different varieties and is used in many different industrial applications.

This paper is a state-of-the-art paper which topic is connected mainly with evolutionary algorithms (EAs) such as GA, GP, DE, ES, and EP. (In the other paper [6], we have presented swarm intelligence algorithms (SIAs) such as ant colony optimization (ACO), particle swarm optimization (PSO), and others in which social collaboration between agents exist.) The other nature-based methods, like family of physical algorithms (e.g., simulated annealing, extremal optimization, harmony search, cultural algorithm, gravitational search, river formation dynamics, black hole algorithm), or family of plant intelligence algorithms (e.g., flower pollination algorithm, invasive weed optimization, paddy field algorithm, artificial plant optimization algorithm, photosynthetic algorithm, plant growth optimization, rooted tree optimization), are not considered here due to their less popularity.

The aim of this paper is to present a short overview of the practical applications of evolutionary algorithms (EAs). The paper is the complement to [6] where a state of the art of industrial (real-life) applications of swarm intelligence is presented. The paper is organized as follows. In Sect. 2, we briefly present the main EAs, namely *genetic algorithm*, *genetic programming*, *differential evolution*, *evolution strategies*, and *evolutionary programming*. Section 3 describes the various uses of the considered methods in

✉ Adam Slowik
aslowik@ie.tu.koszalin.pl

[1] Department of Electronics and Computer Science, Koszalin University of Technology, Sniadeckich 2 Street, 75-453 Koszalin, Poland

[2] Department of Computational Intelligence, Wroclaw University of Science and Technology, Wybrzeze Wyspianskiego 27 Street, 51-370 Wrocław, Poland

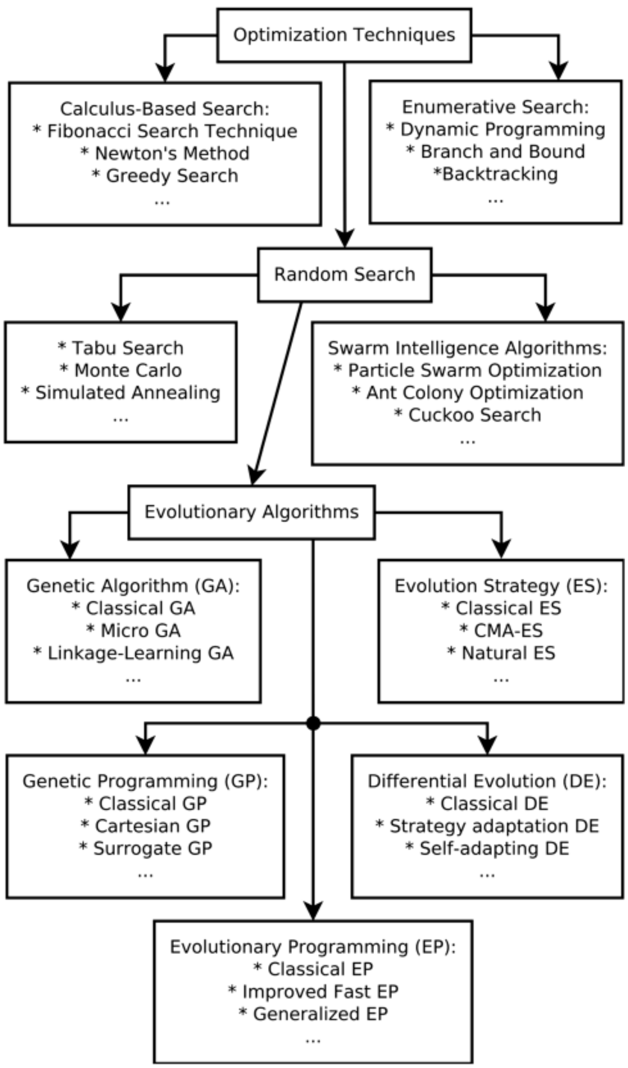

Fig. 1 Taxonomy of nature-inspired methods

selected areas. Finally, recent advances and the current trends of the EAs are described.

2 Brief presentation of the EAs

2.1 Genetic algorithms

The *genetic algorithm* (GA) [1] is one of the oldest and most known optimization techniques, which are based on nature. In the GA, the search for solution space imitates the natural process which takes place in the environment, and the Darwinian theory of species evolution is taken into consideration. In GAs, we have a population of individuals; each, called a chromosome, represents a potential solution to the problem. The problem being solved is defined by the objective function. Depending on how "good" the given individual is fitted to the objective function, the value which represents its quality is attributed to it. This value is referred to as the fitness of the individual, and it is a main evaluating factor. Highly valued individuals have a better chance to be selected to the new generation of the population. In GAs, we have three operators: selection (a new population of individuals is created based on the fitness values of individuals from the previous generation), crossover (typically parts of individuals are exchanged between two individuals selected to the crossover), and mutation (the values of particular genes are changed randomly). Algorithm 1 presents the standard GA in the pseudo-code form (for more details see [7]).

Algorithm 1 Genetic Algorithm

```
determine objective function (OF)
assign number of generation to 0 (t=0)
randomly create individuals in initial population P(t)
evaluate individuals in population P(t) using OF
while termination criterion is not satisfied do
    t=t+1
    select the individuals to population P(t) from P(t-1)
    change individuals of P(t) using crossover and muta-
tion
    evaluate individuals in population P(t) using OF
end while
return the best individual found during the evolution
```

Many modifications of the standard GA have been developed; some of them are listed in Table 1.

2.2 Genetic programming

Genetic programming (GP) [2] is relatively new; it is a specialized form of a GA which operates on very specific types of solution, using modified genetic operators. The GP was developed by Koza [2] as an attempt to find the way for the automatic generation of the program codes when the evaluation criteria for their proper operation is known. Because the searched solution is a program, the evolved potential solutions are coded in the form of trees instead of linear chromosomes (of bits or numbers) widespread in GAs. As GP differs from GA the used coding schema, the main loop of GP is the same as in Algorithm 1. Of course, the genetic operators are specialized for working on trees, e.g., crossover as exchanging the subtrees, mutation as a change of node or leaf. Some modifications of the GP are shown in Table 1.

2.3 Differential evolution algorithm

The *differential evolution* (DE) is a type of evolutionary algorithm useful mainly for the function optimization in continuous search space. Although a version of DE algorithm for combinatorial problems has also been discussed [51], the principal version of the DE algorithm was

Table 1 The 46 algorithms selected from the whole family of the GA, GP, and DE

Algorithm, references	Author	Short description	Year
Genetic algorithm (GA)			
GA [1]	Holland	Simulate the natural evolution process—survival of the fittest individuals	1975
Micro GA [8]	Krishnakumar	Evolve very small populations that are efficient in locating promising areas	1989
Cellular GA [9]	Manderick et al.	Based on the concept of structured populations and GAs	1989
Non-dominated sorting GA [10]	Srinivas et al.	Extension of the GA for multiple-objective function optimization	1994
Contextual GA [11]	Rocha	Inspired by the biological system of RNA editing found in organisms	1995
Grouping GA [12]	Falkenauer	Developed to solve clustering problems	1996
Quantum-inspired GA [13]	Narayanan et al.	Concepts and principles of quantum mechanics are used in algorithm	1996
Linkage learning GA [14]	Harik	Algorithm is capable of learning genetic linkage in the evolutionary process	1997
Island GA [15]	Whitley et al.	Multiple subpopulations helps to prevent genetic diversity	1998
Non-dominated sorting GA II [16]	Deb et al.	Extension of NSGA with fast non-dominated sorting approach	2000
Interactive GA [17]	Takagi	The fitness of individuals is assigned by human rather by a function	2001
Jumping gene GA [18]	Man et al.	Inspired by the biological mobile genes mechanism existing in chromosome	2004
Dynamic rule-based GA [19]	He et al.	Based on heuristic rules which are crucial to cut down the solution space	2006
Hierarchical cellular GA [20]	Janson et al.	Population structure is augmented with a hierarchy according to the fitness	2006
Non-dominated sorting GA III [21]	Deb et al.	Extension of NSGA-II by using a reference point approach	2014
Tribe competition-based GA [22]	Ma et al.	Population of individuals is divided into multiple tribes	2017
Fluid GA [23]	Jafari-Marandi et al.	Biologically, the fluid GA is closer to what happens in the genetic world	2017
Block-based GA [24]	Tseng et al.	GA more suitable for construct the disassembly sequence planning	2018
Genetic programming (GP)			
GP [2]	Koza	Populations of computer programs are genetically bred using natural selection	1992
Cartesian GP [25]	Miller et al.	Represents a program using two-dimensional grid of nodes	2000
Grammar-guided GP [26]	Ratle et al.	Initialization procedure is based on dynamic grammar pruning	2000
Gene expression programming [27]	Ferreira	Chromosomes encode expression trees which are the object of selection	2001
Multi-gene GP [28]	Kaydani et al.	Structure selection combined with a classical method for parameter estimation	2012
Geometric semantic GP [29]	Moraglio et al.	Searches directly the space of the underlying semantics of the programs	2012
Surrogate GP [30]	Kattan et al.	One of the two populations is evolved with the aid of meta-models	2015
Memetic semantic GP [31]	Ffrancon et al.	Taken into account the semantics of a GP tree	2015
Statistical GP [32]	Haeri et al.	Uses statistical information to generate some well-structured subtrees	2017
Multi-dimensional GP [33]	La Cava et al.	GP with novel program representation for multi-dimensional features	2018
Differential evolution (DE)			
DE [3]	Storn et al.	Differential mutation operator is used to perturb vectors in population	1997
SaDE [34]	Qin et al.	Suitable learning strategy and parameter settings are gradually self-adapted	2005
jDE [35]	Brest et al.	An efficient technique for adapting control parameter settings is used	2006
Chaotic DE [36]	Wang et al.	Properties of chaotic system are used to spread the individuals in search space	2007
JADE [37]	Zhang et al.	Updating control parameters in an adaptive manner	2009
EPSDE [38]	Mallipeddi et al.	Pool of mutation strategies with a pool of control parameter values coexists	2011
CoDE [39]	Wang et al.	Combining trial vector generation strategies with control parameter settings	2011
Multi-population DE [40]	Yu et al.	Multiple subpopulations exchange information via mutation operation	2011
ACDE [41]	Choi et al.	Each individual has its own control parameters	2013
Improved JADE [42]	Yang et al.	Extension of JADE algorithm in which the adaptation of *CR* is improved	2014
Extended adaptive Cauchy DE [43]	Choi et al.	Modification of ACDE by attaching bias strategy adaptation mechanism	2014
jDErpo [44]	Brest et al.	Uses a gradually increasing mechanism for controlling control parameters	2014
RDEL [45]	Ali	Based on a couple of local search mutation and a restart mechanism	2014

Table 1 (continued)

Algorithm, references	Author	Short description	Year
Colonial competitive DE [46]	Ghasemi et al.	Algorithm is based on mathematical modeling of sociopolitical evolution	2016
Memory-based DE [47]	Parouha et al.	Two "swarm operators" have been introduced into DE algorithm	2016
SQG-DE [48]	Sala et al.	Combines aspects of stochastic quasi-gradient methods within DE algorithm	2017
UDE [49]	Trivedi et al.	Unifying the main idea of CoDE, JADE, SaDE, and ranking-based mutation	2017
OCSinDE [50]	Draa et al.	Use a compound sinusoidal formula for scaling factor and crossover rate	2018

discussed by Storn and Price [3]. The main advantages of DE over a traditional GA are: It is easy to use, and it has efficient memory utilization, lower computational complexity (it scales better when handling large problems), and a lower computational effort (faster convergence) [52]. The standard DE procedure is shown in Algorithm 2. Presented there DE optimizes the problem with n decision variables. Parameter F scales the values added to the particular decision variables (mutation), and CR parameter represents the crossover rate [52] ($x_{i,j}$ is the value of jth decision variable stored in ith individual in the population). More detailed information on how the parameters should be tuned can be found in [53]. The main idea of the DE algorithm is connected with computing the difference between two individuals chosen randomly from the population. (The DE determines the function gradient within a given area—not at a single point.) Therefore, the DE algorithm prevents the solution of sticking at a local extreme of the optimized function [52]. Twenty years of DE development resulted in many modifications. Some of them are shortly presented in Table 1.

Algorithm 2 Differential Evolution

determine objective function (OF)
assign number of generation to 0 (t=0)
randomly create individuals in initial population P(t)
while termination criterion is not satisfied **do**
    t=t+1
    for each i-th individual in the population P(t) **do**
        randomly generate three integer numbers:
        $r_1, r_2, r_3 \in$[1; population size], where $r_1 \neq r_2 \neq r_3 \neq$i
        for each j-th gene in i-th individual (j∈[1; n]) **do**
            $v_{i,j} = x_{r1,j} + F \cdot (x_{r2,j} - x_{r3,j})$
            randomly generate one real number $rand_j \in [0; 1)$
            if $rand_j < CR$ **then** $u_{i,j} := v_{i,j}$
            else
                $u_{i,j} := x_{i,j}$
            end if
        end for
        if individual u_i is better than individual x_i **then**
            replace individual x_i by child u_i individual
        end if
    end for
end while
return the best individual in population P(t)

2.4 Evolution strategies

The *evolution strategies* (ESs) are different when compared to the GAs, mainly in the selection procedure. In the GA, the next generation is created from the parental population by choosing individuals depending on their fitness value, keeping a constant size of the population. In the ES, a temporary population is created; it has the different size than the parental population (depending on the assumed parameters λ and μ). In this step, the fitness values are not important. Individuals in the temporary population undergo crossover and mutations. From such populations, an assumed number of the best individuals are selected to the next generation of the population (in a deterministic way). ESs operate on the vectors of the floating point numbers, while the classical GA operates on binary vectors. The primary types of ESs are ES$(1+1)$, ES$(\mu+\lambda)$, and ES(μ, λ) [7].

2.4.1 Evolution strategy ES(1 + 1)

It is the oldest approach; only one individual x is evolved. The initial individual x is randomly generated. In each iteration, only one new individual y is created. The crossover operator does not exist, and the mutation operator creates the individual y by adding a randomly generated number to each gene of the individual x. The normal distribution N with a mean value equal to zero and a standard deviation equal to one is used. The value of ith gene in the individual y is computed as follows: $y_i = x_i + \sigma \cdot N_i(0, 1)$, where σ is a parameter which determines the range of the mutation. Based on the fitness value of individuals x and y, the better one is selected for the new generation and becomes a new individual x. Parameter σ undergoes adaptation by the so-called rule of 1/5 successes. According to this rule, the best results are obtained when the relation R between successful mutations and all mutations is equal to 1/5. When during k successive generations, the relation R is higher than 1/5, then the value of the σ parameter is increased. When the relation R is lower than 1/5, then the value of the σ parameter is decreased. The σ parameter does not change when the relation R is equal to 1/5 [7].

2.4.2 Evolution strategy ES($\mu + \lambda$)

This is an extension of the ES(1 + 1). The ES($\mu + \lambda$) has a self-adaptive mutation range, which replaces the 1/5 success rule implemented in ES(1 + 1). In the ES($\mu + \lambda$), each individual in the population contains additional chromosome σ, consisting of values of standard deviation for each gene. These values are used during mutation procedure. The crossover operator operates before the mutation. Both chromosomes (consisting of the value of variables, and of the value of σ parameters) undergo mutation and crossover processes [7]. Algorithm 3 presents the pseudo-code of the ES($\mu + \lambda$).

Algorithm 3 ES(μ+λ) Evolution Strategy

```
1: determine objective function (OF)
2: assign number of generation to 0 (t=0)
3: randomly create μ individuals in the initial population P(t)
4: evaluate individuals in population P(t) using OF
5: while termination criterion is not satisfied do
6:     t=t+1
7:     create the population T(t) by reproduction λ individuals from population P(t-1)
8:     create the population M(t) by crossover and mutation of individuals from population T(t)
9:     evaluate individuals in population M(t)
10:    select μ the best individuals to population P(t) from the populations P(t-1) and M(t)
11: end while
12: return the best individual in population P(t)
```

2.4.3 ES(μ,λ) evolution strategy

This type of the ES is used more often than ES($\mu + \lambda$). The operation of both algorithms is almost identical. The only one difference is that in the ES(μ, λ), the new population *P*(*t*) is created using only the best individuals from the "children" population *M*(*t*). In this case, μ has to be greater than λ. Such selection gives the advantage of ES(μ, λ) over the ES($\mu + \lambda$); in the latter, the population can be dominated by one individual which is much better than others and the values of standard deviations σ are not well tuned. The ES(μ, λ) does not have this disadvantage because the individuals from the parental population $P(t-1)$ are not copied to the new generation $P(t)$ [7].

The pseudo-code of the ES(μ, λ) is almost the same as Algorithm 3. The only one difference is line 10; here, it is: "10: select μ the best individuals to population $P(t)$ from the population $M(t)$."

Today, the covariance matrix adaptation evolution strategy (CMA-ES) is perceived as a state-of-the-art ES [54, 55]. Several variants of CMA-ES were developed [55] to enhance the efficiency or robustness of the method by different techniques. In the CMA-ES algorithm, the adaptation of the population size or other parameters was presented in papers [56]. The CMA-ES algorithm employs global weighted recombination for both, strategy and object variables, adapts the full covariance matrix for mutation and, in general, is based on the scheme of the ES(μ, λ). The CMA-ES algorithm can handle poorly scaled functions, and its performance remains invariant under rotation of the search space [54]. Some modifications of ESs are mentioned in Table 2.

2.5 Evolutionary programming

Evolutionary programming (EP) was developed as a tool for discovering the grammar of the unknown language. However, EP became more popular when it was proposed as the numerical optimization technique. The EP is similar to the ES($\mu + \lambda$), but with one essential difference [7]. In EP, the new population of individuals is created by mutating every individual from the parental population, while in the ES($\mu + \lambda$), every individual has the same probability to be selected to the temporary population on which the genetic operations are performed. In the EP, the mutation is based on the random perturbation of the values of the particular genes of the mutated individual. The newly created and the parental populations are the same sizes ($\mu = \lambda$). Finally, the new generation of the population is created using the ranking selection of the individuals from both, the parental and the mutated populations. The pseudo-code of the standard EP method is presented in Algorithm 4. EP, like other evolutionary methods, has many modifications. Some of them are listed in Table 2.

Algorithm 4 Evolutionary Programming

```
determine objective function (OF)
assign number of generation to 0 (t=0)
randomly create individuals in initial population P(t)
evaluate individuals in population P(t) using OF
while termination criterion is not satisfied do
    t=t+1
    create population M(t) by the mutation of every individual from the population P(t-1)
    evaluate individuals in population M(t) using OF
    select the individuals to population P(t) from the sum of individuals in P(t-1) and M(t) using ranking selection
end while
return the best individual in population P(t)
```

2.6 Evolutionary algorithms: problems and challenges

EAs are a very interesting research area. There are many open research problems such as: control of the balance between the exploration and exploitation properties; the

Table 2 The 24 algorithms selected from the whole family of ES, and EP

Algorithm, references	Author	Short description	Year
Evolution strategy (ES)			
ES [4]	Rechenberg	Primarily mutation and selection are used as search operators	1973
Derandomized self-adaptation ES [57]	Ostermeier et al.	Derandomized scheme of mutative step size control is used	1994
CSA-ES [58]	Ostermeier et al.	Adaptation concept uses information accumulated from the old generations	1994
CMA-ES [54]	Hansen et al.	Mutation strategy scheme called covariance (COV) matrix adaptation is used	2001
Weighted multi-recombination ES [59]	Arnold	Weighted recombination is used for improving the local search performance	2006
Meta-ES [60]	Jung et al.	Based on incremental aggregation of partial semantic structures	2007
Natural ES [61]	Wierstra et al.	Use the natural gradient to update a parameterized search distribution	2008
Exponential natural ES [62]	Glasmachers et al.	Significantly simpler version of natural ES algorithm	2010
Limited memory CMA-ES [55]	Loshchilov	Reduction in time–memory complexity by covariance matrix decomposition	2014
Fitness inheritance CMA-ES [63]	Liaw et al.	Computational cost reduction at fitness evaluation using fitness inheritance	2016
RS-CMSA ES [64]	Ahrari et al.	Several subpopulations explore the search space in parallel	2017
MA-ES [65]	Beyer et al.	COV update and COV matrix square root operations are no longer needed	2017
Weighted ES [66]	Akimoto et al.	ES with weighted recombination of general convex quadratic functions	2018
Evolutionary programming (EP)			
EP [5]	Fogel et al.	Inspired by macro-level or species-level process of evolution	1966
Improved fast EP [67]	Yao et al.	Uses a Cauchy instead of Gaussian mutation as the primary search operator	1992
Generalized EP [68]	Iwamatsu	A Levy-type mutation is used as the primary search operator	2002
Diversity-guided EP [69]	Alam et al.	Guides the mutation step size using the population diversity information	2012
Adaptive EP [70]	Das et al.	Strategy parameter is updated based on the number of successful mutations	2013
Social EP [71]	Nan et al.	Algorithm is based on a social cognitive model	2014
Immunised EP [72]	Gao	Mutation operation and selection operation based on artificial immune system	2015
Mixed mutation strategy EP [73]	Pang et al.	Employs Gaussian, Cauchy and Levy mutation operators	2016
Fast Convergence EP [74]	Basu	Developed to boost convergence speed and solution quality in EP	2017
Immune log-normal EP [75]	Mansor et al.	Combination of log-normal-based mutation EP with artificial immune system	2017
ADM-EP [76]	Hong et al.	EP with automatically designed mutation operators	2018

self-adaptive (or adaptive) control of steering parameters; reducing the number of CMA-ES algorithm parameters; introducing new selection schemes; and increasing their effectiveness. The latter is important especially in the area of evolutionary design and in evolvable hardware. Also, new more efficient techniques for constraint handling are needed.

Additionally, more investigation into the application of EAs to dynamic optimization problems, to the optimization in noisy and non-stationary environments, and to multi-objective optimization problems (especially with a large number of decision variables) is required. Also, further research is needed in the population size adaptation in different optimization scenarios. Novel strategies should be developed to deal with expensive problems more competitively. Among these matters, there is the open question of constraint handling in EAs specifically to solve engineering optimization problems. As we know, the constraint handling methods can be classified into six main categories: penalty methods, methods evolving in the feasible region, methods using parallel population approaches, methods based on the assumption of superiority of feasible individuals, methods using multi-objective optimization techniques, and hybrid methods. Of course, each of these categories can be divided into several subcategories. The taxonomy of the constraint handling techniques with EAs can be found in the paper [77] by Petrowski et al. If we want to use the proper constraint handling method in EAs for real-world application, we should find the answer to the several questions such as is the objective function defined in the unfeasible domain (if not, the penalization methods cannot be used, for example): are there any active constraints at the optimum? (if not, the methods based on the search on the feasible region boundaries are irrelevant); what is the nature of the constraints? (if only one of the constraints is a nonlinear inequality, the methods for linear constraints are excluded). Moreover, in the real-world application of EAs with constraint handling techniques the effectiveness of a method is often dominated by two other decision criteria such as complexity and difficulty of implementation. Currently, penalty methods, feasibility rules, and stochastic ranking methods are used in real-world applications very often due to their simplicity [77]. Therefore, as we can see, there is no general approach for handling the constraints with EAs able to deal with any real-world problem, so the research on constraint handling techniques in EAs for real-world application is still a hot topic.

In EAs, there are many open research problems, which are discussed in more detail in [53, 78].

Despite these weaknesses, we observe the growing popularity of EAs (please see Figs. 2, 3). If we analyze the number of publications in the Web of Science (WoS) database (years 2000–2018) for particular EAs, we can see that their number is growing from year to year for the algorithms: GA, GP, DE, and ES. Only for EP algorithm, the number of published articles has been decreasing since

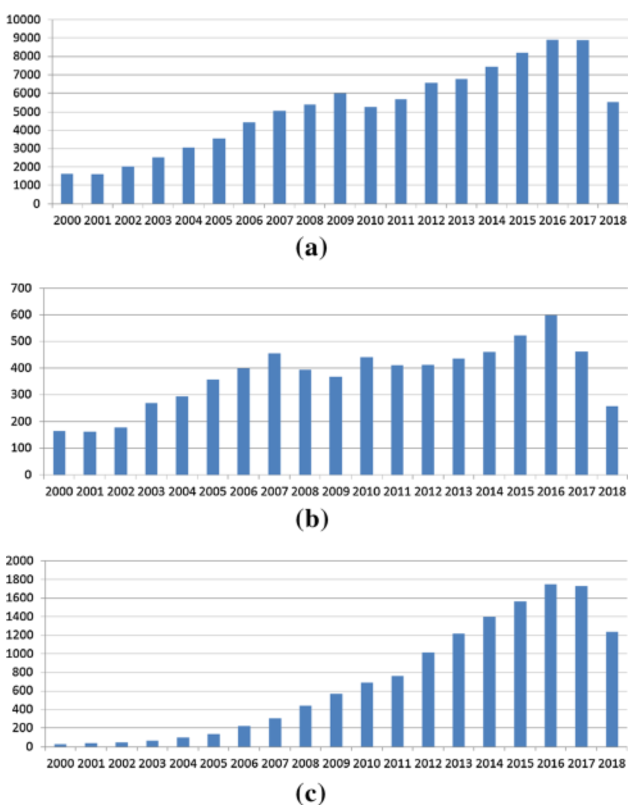

Fig. 2 Number of publications in the WoS database (years 2000–2018): GA (**a**), GP (**b**), DE (**c**)

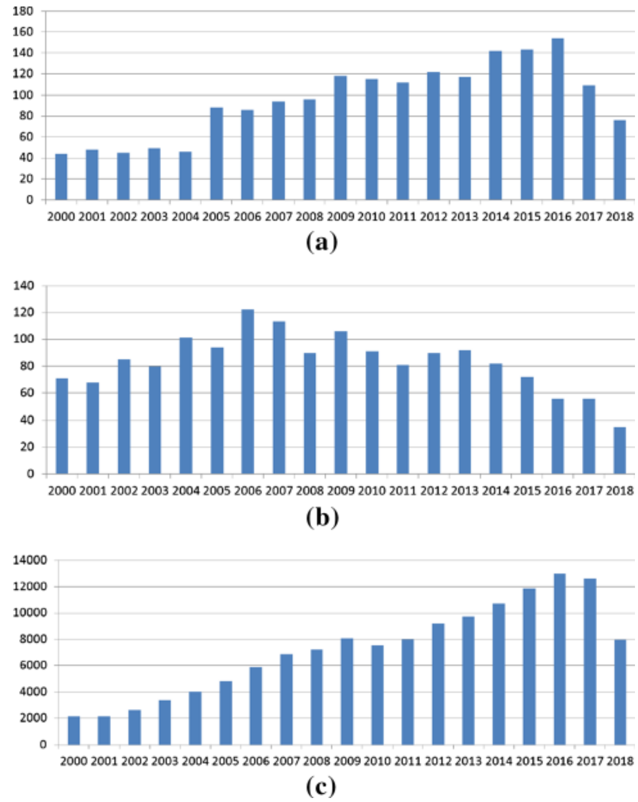

Fig. 3 Number of publications in the WoS database (years 2000–2018): ES (**a**), EP (**b**), sum of publications for all listed algorithms GA, GP, DE, ES, EP (**c**)

the 2013 year. The total number of papers published in WoS database (years 2000–2018) which are related to these algorithms was equal to 98,596 for GA, 7038 for GP, 13,308 for DE, 1804 for ES, and 1585 for EP. Also, we have study the popularity of some EA methods in the selected scientific databases such as Google Scholar, Springer, IEEE Xplore, ACM, Scientific, Science Direct, Sage, Taylor, and Web of Science. The total number of the papers was equal to 1,304,205 for GA, 186,791 for GP, 119,668 for DE, 53,254 for ES, and 73,716 for EP.

Also, many practical applications of EAs methods have been patented by such corporations like Caterpillar Inc., Yamaha Motor Co. Ltd., Fujitsu Limited, International Business Machines Corporation, Lsi Logic Corporation, Honda Research Institute Europe Gmbh, Prometheus Laboratories Inc., Siemens. The total number of patents registered in the Google Patents database (in years 2000–2018) for the particular EA methods was equal to 43,284 for GA, 2960 for GP, 2039 for DE, 1191 for ES, and 1583 for EP. More detailed information is presented in Table 3.

We believe that over the next few years researchers will focus on the above areas.

Table 3 Number of patents registered in the Google Patents database

Method	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
GA	401	618	846	892	995	1098	1309	1405	1611	1789	1999	2307	2841	3230	3476	3857	4401	5886	4323
GP	43	54	72	129	108	131	141	129	171	179	188	190	223	224	271	273	302	345	296
DE	3	1	2	4	4	10	6	26	39	49	58	64	97	143	207	231	320	410	365
ES	17	19	27	32	30	45	44	48	54	47	65	69	74	87	84	97	102	123	127
EP	22	24	37	58	49	54	53	53	55	55	49	52	65	68	83	77	88	77	55
Total	486	716	984	1115	1186	1338	1553	1661	1930	2119	2359	2682	3300	3752	4121	4535	5213	6841	5166
Total number of patents in years 2000–2018 for the particular methods																			
GA: 43,284				GP: 2960				DE: 2039				ES: 1191				EP: 1583			

A number of patents in general, and in selected areas: *GA* genetic algorithm, *GP* genetic programming, *DE* differential evolution, *ES* evolution strategy, *EP* evolutionary programming

3 Evolutionary algorithms in real-life problems

Similar to swarm intelligence algorithms [6], a major reason is a growing demand for smart optimization methods in many business and engineering activities. EAs are suitable mainly for optimization, scheduling, planning, design, and management problems. These kinds of problems are everywhere, in investments, production, distribution, and so forth. If we analyze, the results obtained from the WoS database (popularity of only ten first WoS categories for each method—for more detailed information see Table 4), we can see that the EAs methods are mainly used in the area such as:

- Engineering electrical electronics,
- Computer science artificial intelligence,
- Computer science theory methods,
- Computer science interdisciplinary applications,
- Automation control system,
- Computer science information systems,
- Operations research management science.

When in WoS we will select a field *Highly Cited in Field*, we can see that the highly cited papers (in which EA methods are used) are from the following industry areas for the particular EA methods:

- GA—energy fuels (EF), engineering electrical electronic (EEE), operations research management science (ORMS), engineering civil (EC),
- GP—engineering civil (EC), water resources (WR), energy fuels (EF), automation control systems (ACS),
- DE—energy fuels (EF), automation control systems (ACS), engineering electrical electronics (EEE), engineering civil (EC),
- ES—construction building technology (CBT), energy fuels (EF), engineering civil (EC), engineering electrical electronic (EEE),
- EP—construction building technology (CBT), engineering civil (EC), computer science software engineering (CSSE), transportation science technology (TST).

Therefore, in this paper, we will concentrate only on the real-world applications of particular EA methods in above areas of industry. In the next subsections, the abbreviation of industry area will be given in parenthesis after reference number to currently discussed paper.

Table 4 Number of the papers registered in the WoS database for ten the most popular area of applications for each algorithm

Area of applications	GA	GP	DE	ES	EP	ALL
Engineering electrical electronic	26,961	1364	3902	516	755	33,498
Computer science artificial intelligence	23983	3376	4294	655	634	32,942
Computer science theory methods	13,253	2460	2071	451	330	18,535
Computer science interdisciplinary applications	11,355	888	1669	216	177	14,305
Automation control systems	9217	437	1027	120	176	10,977
Computer science information systems	8318	673	996	136	170	10,293
Operations research management science	7397	282	791	106	74	8650
Engineering mechanical	6197	–	–	–	–	6197
Telecommunications	5850	–	725	–	98	6673
Energy fuels	5325	–	835	–	149	6309
Computer science software engineering	–	688	–	111	79	878
Mathematical computational biology	–	329	–	–	–	329
Engineering civil	–	304	–	–	–	304
Engineering multi-disciplinary	–	–	785	–	–	785
Physics applied	–	–	–	125	–	125
Mathematics applied	–	–	–	98	–	98

GA Genetic algorithm, *GP* genetic programming, *DE* differential evolution, *ES* evolution strategy, *EP* evolutionary programming, *ALL* sum of the papers for all listed algorithms in given area of applications

3.1 Genetic algorithms in real-life problems

The GAs are a universal optimization tool. Using GAs, we can solve constrained optimization problems, multimodal optimization problems, continuous optimization problems, combinatorial optimization problems, and multi-objective optimization problems. Thus, there is a wide range of real-world applications of GAs. In this short section, we show only a few of them.

The paper [79] (EF) by Lv et al. presents the solar array layout optimization problem which is solved by GA. The presented numerical method is based on rotating model of a stratospheric airship to optimize the solar array layout. The results demonstrate that the proposed method is helpful in the preparation stage for installing large area flexible solar arrays. Also, it is shown that due to solar array optimization the output power of solar panel is significantly improved.

In paper [80] (EF) by Ma et al., the optimization model based on the GA, developed to reduce the energy consumption of high-sulfur natural gas purification process, is presented. A case study was performed in a high-sulfur natural gas purification plant with the capacity of $300 \cdot 10^4\,\mathrm{N\,m^3/d}$. The results demonstrate that the energy consumption of the purification plant was reduced by 12.7%.

In [81] (EEE), Yin et al. report a GA-inspired strategy designed and incorporated in the sequential evolutionary filter. Due to this strategy, the resampling used in most of existing particle filters is not necessary, and the particle diversity can be maintained. The experimental results show that the proposed sequential evolutionary filter offers better state estimation results than three other comparative filters.

The authors of [82] (EEE) investigate the pros and cons of hybridization of a GA and local search on the basis of a hard practical and up-to-date problem, namely the routing and spectrum allocation of multi-cast flows (RSA/M) in elastic optical networks (EONs). They proposed an efficient optimization method for solving the RSA/M problem in EONs. The proposed method outperformed all other competing methods. Additionally, introduction of Baldwin effects helped to preserve the population diversity in GA.

In the paper [83] (ORMS), the local-inventory-routing model for perishable products is presented. The proposed model integrates the three levels of a decision in the supply chain such as the number and location of required warehouses, the inventory level at each retailer, and the routes traveled by each vehicle. It is shown that the model developed in this paper is NP-hard; therefore, the authors develop a GA-based approach to solve this problem efficiently. It is shown that presented approach achieves a high-quality near-optimal solution in reasonable time.

The paper [84] (ORMS) by Ramos et al. presents new container loading algorithm with load balance, weight limit, and stability constraints which use a load distribution diagrams. This algorithm is based on multi-population biased random-key GA, with a new fitness function that takes static and loads balance into account. Due to incorporate weight balance goal with stability guaranteed by full base support and by the mechanical equilibrium conditions, the proposed approach is very effective.

In [85] (EC) by Yan et al., a framework to determine the investment plan to strengthen a railway system to earthquake hazard is proposed. This framework consists of four parts. In the third part, an investment optimization model is formulated, and in part four, this model is solved using GA. The proposed approach has been applied to the real Chinese railway system. The obtained results show that the presented framework is more responsive to the earthquake impact on railway system compared to topology-based methods.

In the paper [86] (EC) by Ascione et al., the multi-objective optimization of operating cost for space conditioning and thermal comfort to achieve a high level of building energy performance is presented. The main objective of proposed GA is to optimize the hourly set point temperatures with a day-ahead horizon, based on a forecast of weather conditions and occupancy profile. In comparison with the standard control strategy, the presented approach generates a reduction of operating cost up to 56%.

In [87] (EC) by Lin et al., a time-optimal train running reference curve is designed with least time-consuming, but highest energy consumption, and it is optimized by adding multi-point coasting control to realize energy saving with a relative rise in time. Multi-population GA is adopted to solve this multi-point combinatorial optimization problem. Simulation results, based on real line condition and train parameters of Shanghai line 7, demonstrate the advancement of multi-point coasting control with the proposed approach.

In [88] (EC), the application of GA to minimization of average delay for an urban signalized intersection under the oversaturated condition is presented. Relieving urban traffic congestion is an urgent call for traffic engineering. One of the key solutions to reduce congestion is the effectiveness of traffic signalization. The current traffic signal control system is not fully optimized for handling the oversaturated condition. Simulation results show that GA is able to control the traffic signals for minimizing the average delay to 55 s/vehicle.

Zhang et al. [89] (EEE) use the flexible GA for node placement problems. Node placement problems are encountered in various engineering fields, e.g., the deployment of radio-frequency identification systems or wireless sensor networks. The flexible GA with variable length encoding, subarea-swap crossover, and Gaussian mutation is able to adjust the number of nodes and their corresponding properties automatically. Experimental results show that the flexible GA offers higher performance than existing tools for solving node placement problems.

In the paper [90] (EF) by Reddy, the scheduling problem considering the hybrid generation system is presented. The new strategy based on GA for the optimal scheduling problem taking into account the impact of uncertainties in the wind, solar photovoltaic modules with batteries, and load demand forecast is proposed. From simulation results (for IEEE 30 and 300 bus test systems), it can be noticed that with a marginal increase in the cost of day-ahead generation schedule, a significant reduction in real-time mean adjustment cost is obtained.

3.2 Genetic programming in real-life problems

The GP possesses many practical applications.

In [91] (ACS) et al. the handwriting character recognition system for inertial-sensor-equipped pens is presented. In this system, the characteristic function is calculated for each character using a GP algorithm. The experimental results show that the performance of the proposed method is superior to that one of the state-of-the-art works in the area of recognizing Persian/Arabic handwriting characters.

Bagatur and Onen [92] (WR) propose novel models for the prediction of flood routing in natural channels using the gene expression programming (GEP) algorithm, which is one of the extensions of GP algorithm. The GEP method makes use of few hydrologic parameters such as inflow, outflow, and time. The performance of the proposed models is evaluated by two goodness-of-fit measures. The proposed GEP models are tested for the three datasets taken from the literature. It is proved that the GEP models show superior performance to the other solution techniques based on the Muskingum model.

In the paper [93] (EF) by Abkenar et al., an intelligent fuel cell (FC) power management strategy is proposed. The main objective of the proposed approach is to improve FC performance at different operating points without employing DC/DC interfacing converters. A hybrid all-electric ships (AES) driveline model using GP is utilized to formulate operating FC voltage based on the load current, FC air, and fuel flow rates. The proposed approach maintains FC performance and reduces fuel consumption, and therefore ensures the optimal power sharing between the FC and the lithium-ion battery in AES application.

The authors of [94] (EC) use GP algorithm to develop models to predict the deterioration of pavement distress of the urban road network. Five models for the prediction of pavement distress progression such as cracking, raveling, pothole, rutting, and roughness are created. In order to obtain a training dataset, and validation dataset, the real data from the roads of Patiala City, Punjab, India, have been collected. It was shown that GP models predict with high accuracy for pavement distress and help the decision makers for adequate and timely fund allocations for the preservation of the urban road network.

3.3 Differential evolution in real-life problems

The DE algorithm also found many real-world applications.

In [95] (EF), Ramli et al. present an application of multi-objective SaDE algorithm for optimal sizing of a photovoltaic (PV)/wind/diesel hybrid microgrid system (HMS) with battery storage. The multi-objective optimization is used to analyze the loss of power supply probability, the cost of electricity, and the renewable factor in relation to HMS cost and reliability. The proposed approach is tested using three case studies involving differing house numbers for the city Yanbu, Saudi Arabia. The results obtained are useful in investigating optimal scheduling of HMS components and can be used as a power reference for the economic operation of PV and wind turbine generators.

Yao et al. [96] (EF) use a multi-objective DE algorithm for optimizing a novel combined cooling, heating, and power-based compressed air energy storage system. The system combines a gas engine, ammonia–water absorption refrigeration system, and supplemental heat exchangers. The proposed optimization technique is used to find a trade-off between the overall exergy efficiency and the total specific cost of final product. The best trade-off solution which was selected possesses a total product unit cost of 20.54 cent/kWh and an overall exergy efficiency of 53.04%.

In the paper [97] (ACS) by Wang et al., the DE algorithm is applied for wind farm layout optimization with the aim of maximizing the power output. Due to a new encoding mechanism in DE, the dimension of the search space is reduced to two, and a crucial parameter (i.e., the population size) is eliminated. In comparison with seven other methods, the proposed approach is able to obtain the best overall performance, in terms of the power output and execution time.

The authors of [98] (EEE) investigate the problem of linear dipole array synthesis. Dynamic DE algorithm is proposed for synthesizing shaped power pattern by using element rotation and phase optimization for a linear dipole array. Based on two experiments for synthesizing flattop and cosecant squared pattern, the effectiveness and advantages of the proposed approach were verified in comparison with the phase-only optimization and the amplitude-phase joint optimization.

Tian et al. [99] (EC) use a multi-objective hybrid DE+PSO algorithm in order to create a set of Pareto solutions for the problem of dual-objective scheduling of rescue vehicles to distinguish forest fires. The novel multi-objective scheduling model to handle forest fires subject to limited rescue vehicles constraints, in which a fire spread model is introduced into this problem to better describe practical forestry fire is presented. Results show that the proposed approach is able to quickly produce satisfactory Pareto solutions in comparison with GA and PSO algorithms.

3.4 Evolution strategies in real-life problems

Studying the literature, we can find fewer papers with the real-life applications of ESs than those with GAs. Below we shortly present some of them.

The paper [100] (CBT) by Hasancebi presents ES integrated parallel optimization algorithm to minimize the total member weight in each test steel frame. Steel frames with various beam–column connection and bracing configuration are considered for comparative cost analyzes. Three multi-story buildings are chosen (10, 20, and 30-story buildings) as examples for numerical verification of proposed method. The results collected are utilized to reach certain recommendations regarding the selection of economically feasible frames for the design of multi-story steel buildings.

In [101] (EF), Fadda et al. consider the usage of electric batteries in order to mitigate it. In energy distribution systems, uncertainty is the single major cause of power outages; therefore, the authors propose intelligent battery able to maximize its lifetime while guaranteeing to satisfy all the electric demand peaks. The battery exploits a customized steady-state ES to dynamically adapt its recharge strategy to changing environments. Experimental results on both synthetic and real data demonstrate the efficacy of the proposed approach.

In the paper [102] (EC) by Ogidan et al., the enhanced non-dominated sorting ES algorithm that uses a specialized operator to guide the algorithm toward known sanitary sewer overflows (SSOs) locations is presented. The main objectives of the proposed method are the maximization of SSO reduction and minimization of rehabilitation cost. The proposed method was tested in an existing network in the eastern San Antonio Water System network. The presented approach improves the convergence rate by approximately 70% over the tested alternative algorithms.

The authors of [103] (EEE) investigate the problem of wireless sensor fault diagnosis based on fusion data analysis. The fault diagnosis model is proposed based on the hierarchical belief rule-based model, and the CMA-ES algorithm is used to optimize the initial parameters of the proposed model. In order to validation of presented approach, a case study based on Intel laboratory dataset of sensors is designed. The experiments prove the effectiveness of the proposed method in comparison with back propagation neural network model and the fuzzy expert system.

In the paper [104] (EEE), the problem of subsurface inverse profiling of a 2-D inhomogeneous buried dielectric target is presented and solved using proposed iterative optimization method which is based on CMA-ES algorithm. In relation to the results obtained using EP and PSO, the results obtained using CMA-ES significantly outperform the other two optimization techniques in the inhomogeneous imagining.

The paper [105] (EEE) by Emadi et al. presents CMA-ES algorithm for tasks scheduling in the cloud computing environment. The need for planning the scheduling of the user's jobs is an important challenge in the field of cloud computing. The causes are manifold; the most important are: ever-increasing advancements of information technology, an increase in applications and user needs for these applications with high quality, also the popularity of cloud computing among user, and rapid growth of them during recent years. The results obtained indicate that presented algorithm, led to a reduction in execution time of all tasks, compared to the shortest processing time algorithm, longest processing time algorithm, and GA and PSO algorithms.

3.5 Evolutionary programming in real-life problems

In the literature, we can find the applications of EP in many different areas. However, in WoS the number of papers in which EP algorithm is used is decreasing since the 2013 year. Below we shortly present some of them.

The paper [106] (TST) by Yan et al. presents bi-subgroup self-adaptive EP algorithm for seeking the Pareto optimal solution of the multi-objective function of the hybrid electric vehicle (HEV) and the best degree of hybrid (DOH) for this vehicle. In the proposed algorithm, the evolution of Cauchy operator and Gauss operator are parallel performed with different mutation strategies. Moreover, the Gauss operator owns the ability of self-adaptation according to the variation of adaptability function. The simulation results show that the optimal DOH is equal to 0.311 for given HEV. Also, the validity of simulating method was proved, and the fuel saving effect was consistent with authors' expectations.

In the paper [107] (CBT) by Gao, a new evolutionary neural network whose architecture and connection weights simultaneously evolve is proposed. This neural network is based on immunized EP algorithm and is used in the novel inverse back analysis for underground engineering. As a numerical example, an underground roadway of the Huainan coal mine in China is chosen for the verification of the accuracy of the presented inverse back analysis. The results obtained show that using the proposed method, the computed displacements agree with the measured ones. Therefore, it is demonstrated that the new inverse back analysis method is a high-performance method for usage in underground engineering.

Jiang et al. [108] (EC) use EP algorithm to find weights and the threshold value in the neural network which is applied to the traffic signal light control. According to the historical traffic flow data of a crossroad, the next node's traffic flow data are predicted. Due to predicted data, the traffic signal light frequency can be re-adjusted in order to improve traffic congestion and other traffic problems. The results obtained show that the connection of EP algorithm with the neural network has a good effect on traffic signal light optimization.

The authors of [109] (CSSE) propose a novel approach to navigate over 3-D terrain using best viewpoints. The concept of viewpoint entropy is exploited for best view determination, and greedy n-best view selection is used for visibility calculation. In order to connect the calculated viewpoints, the authors use an EP algorithm for the traveling salesman problem. It was shown that the computed and planned viewpoints reduce human effort when used as starting points for scene tour. The proposed method was tested on real terrain and road network datasets.

3.6 Which EA should be used for a given problem?

What lesson for a potential user of evolutionary computation emerges from the above overview? The question is simple, but the answer is hard. All discussed methods are from the same family—evolutionary approaches to optimization problems. The principal question could be: Which of the discussed methods is suitable for the given problem? Expanding the answer to all heuristic methods in general, not just evolutionary algorithms, the best answer seems to be: take the method you know best, you can define your problem well in terms required by this method, you understand the sensitivity of this method to parameters, you can fine-tune this method. Let us see, for example, on energy fuels area. Numerous evolutionary approaches are applied within this scope. It is not possible to indicate one of them as the best for this particular subject. The similar situation is with other areas of industry.

As we can see, the literature on the evolutionary algorithms in general and in their industrial applications is plentiful, but very rarely this literature concerns applications that have been used in practice. Following [110], we can say that the theory does not support the practice; there is a big gap between theory and practice. Theoretical results on properties such as convergence, diversity, exploration, exploitation, deceptiveness, and epistasis are not useful enough for practice. Significant topics from the practice point of view are constraint and noise handling methods, robustness, or multi-objective optimization. The progress in the above matters is also observed; however, these methods are tested mainly on simple silo problems or standard sets of numerical functions, so their usefulness to practitioners working on EA-based software applications is very limited.

It is worth mentioning that the real usefulness of EAs could be not only in industry. The spectacular achievement of EA is presented in [111]. The artificial intelligence system, with the use of EA, the first time discovered a new theory, namely a mechanism of planar regeneration. The remarkable ability of these small worms to regenerate body parts made them a research model in human regenerative medicine.

4 Summary and future trends

As it is shown in this paper, the evolutionary algorithms are a popular research domain. Each year many new modifications of these algorithms are proposed. Some of these modifications are shortly described in Tables 1 and 2. The EAs are applied to solve many industry problems. When we cannot use a dedicated algorithm for a given problem, one of the EAs will be a good choice. Of course, we must remember about specific issues the user can face when dealing with EAs. Here, we can mention two main problems. First of this problem is a premature convergence (the population converging to a suboptimal solution instead of an optimal one). We can solve this problem by introducing the mechanism which will provide a lower transfer rate of the genetic material between individuals—the whole population is divided into several subpopulations (so-called islands) and periodically migrate an individual between islands [15]. Another solution of the premature convergence problem is a cooperation of EAs with branch and bound algorithm endowed with interval propagation techniques, as it was shown in [112]. The second problem is related to the optimal trade-off between exploration and exploitation properties of EA. One of the solutions to this problem is control of the level of selection pressure [113]. We can do this by introducing specialized genetic operators which will guarantee high population diversity at the start of the algorithm operation (high exploration property–small exploitation property) and a low population diversity at the end of the algorithm operation (low exploration property–high exploitation property). A survey about exploration and exploitation in EAs can be found in [114].

As future trends in EAs, we can mention some main directions. The first of current trend is a hybridization of two or more algorithms to obtain better results. Currently, in the literature, we can find an increasing number of papers where hybrid algorithms are presented. Also, many researchers work on modifications of EAs to improve their computational performance. In many recently published papers, we can find modifications of GA [22, 23, 115, 116], GP [31, 32, 117, 118], DE [48, 49, 119, 120], ES [64, 65], and EP [74, 75]. An interesting domain of future research in EAs is also memetic algorithms. The term memetic algorithm is widely used as a synergy of the evolutionary algorithm or any other population-based approach with separate local search techniques as the Nelder–Mead method. We can find very interesting information about future trends in EAs in the paper [121] written by Eiben et al. As one of the future trends in EAs, the authors point out the increasing interest in applying EAs to embodied or embedded systems, that is, employing evolution in populations for which the candidate solutions are controllers or drivers that implement the operational strategy for some situated entities, and are evaluated within the context of some rich, dynamic environment: not for what they are, but for what they do. Finally, there is another one important issue especially in the industrial application of EA methods. Very often in real-world problems, we must optimize a function in a high-dimensional domain. This process usually is very complex and takes a lot of computational time. Therefore, in real applications, the EAs designed for this type of problems should be designed to be implemented easily to run in parallel (or easy to run in GPU) to reduce their computational time. A greater effort in this feature should be in future proposals because this could be a crucial feature to decide whether an algorithm is useful in real applications. Some research in the area of EAs can be connected with the so-called surrogate models (computationally cheaper models of real-world problems) which can be used in the place of full fitness evaluation, and that refine those models through occasional full evaluations of individuals in the population [121]. Also, very often industry problems have many objectives. In tandem with algorithmic advances, the interactive evolutionary algorithms are used to increase the efficiency of EAs in multiobjective optimization [121]. As we know, each engineering problem is defined by the different objective function and has a different landscape of search space. The values of EAs parameters which are "good" in one problem cannot be sufficient in another one. Therefore, searching for new techniques in such area as automated tuning and adaptive parameter control is still a hot topic in EAs. Another important issue in the industrial application of EA methods is a proper definition of an objective function. The industrial problems are very complex. Therefore, a definition of a good mathematical model (good objective function for EAs) for a given industry process is also a very demanding task. The "quality" of the chosen objective function will have a great influence on the results obtained using EA methods. The next issue which we want to mention in discussing is repeatability of the EA methods. As we know, the EAs are stochastic techniques. Each time the EA method is run, a different result can be obtained. Therefore, the main focus should be on ensuring repeatability of the results generated by EA techniques. This issue is very important for application on EA methods in industry.

In summary, we believe that in the future, new evolutionary algorithms will be developed, and the research problems connected with evolutionary algorithms will always be a hot topic for researchers.

Compliance with ethical standards

Conflict of interest In the present work, we have not used any material from previously published. So we have no conflict of interest.

References

1. Holland JH (1975) Adaptation in natural and artificial systems. MIT Press, Cambridge
2. Koza J (1992) Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge
3. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359
4. Rechenberg I (1973) Evolutionstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution. Frommann-Holzboog Verlag, Stuttgart
5. Fogel LJ, Owens AJ, Walsh MJ (1966) Artificial intelligence thorough simulated evolution. Wiley, New York
6. Slowik A, Kwasnicka H (2018) Nature inspired methods and their industry applications—swarm intelligence algorithms. IEEE Trans Ind Inf 14(3):1004–1015
7. Rutkowski L (2008) Computational intelligence: methods and techniques. Springer, Berlin
8. Krishnakumar K (1989) Micro-genetic algorithms for stationary and non-stationary function optimization. In: SPIE proceedings: intelligent control and adaptive systems, pp 289–296
9. Manderick B, Spiessens P (1989) Fine-grained parallel genetic algorithm. In: Proceedings of the third international conference on genetic algorithms, pp 428–433
10. Srinivas N, Deb K (1994) Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evol Comput 2(3):221–248
11. Rocha LM (1995) Contextual genetic algorithms: evolving developmental rules. In: Advances in artificial life, pp 368–382
12. Falkenauer E (1996) The grouping genetic algorithm. In: State of the art in global optimization, pp 249–265
13. Narayanan A, Moore M (1996) Quantum-inspired genetic algorithms. In: Proceedings of the IEEE international conference on evolutionary computation, pp 61–66
14. Harik GR (1997) Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms, Ph.D. thesis. University of Michigan
15. Whitley D, Rana S, Heckendorn RB (1998) The island model genetic algorithm: on separability, population size and convergence. J Comput Inf Technol 7:33–47
16. Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Proceedings of the international conference on parallel problem solving from nature, pp 849–858
17. Takagi H (2001) Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. Proc IEEE 89(9):1275–1296
18. Man KF, Chan TM, Tang KS, Kwong S (2004) Jumping-genes in evolutionary computing. In: Proceedings of the IEEE IECON 2004, pp 1268–1272
19. He Y, Hui C-W (2006) Dynamic rule-based genetic algorithm for large-size single-stage batch scheduling. Comput-Aided Chem Eng 21:1911–1916
20. Janson S, Alba E, Dorronsoro B, Middendorf M (2006) Hierarchical cellular genetic algorithm. In: Proceedings of the European conference on evolutionary computation in combinatorial optimization, EvoCOP, pp 111–122
21. Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans Evol Comput 18(4):577–601
22. Ma B, Xia Y (2017) A tribe competition-based genetic algorithm for feature selection in pattern classification. Appl Soft Comput 58:328–338
23. Jafari-Marandi R, Smith BK (2017) Fluid genetic algorithm (FGA). J Comput Des Eng 4:158–167
24. Tseng H-E, Chang C-C, Lee S-C, Huang Y-M (2018) A block-based genetic algorithm for disassembly sequence planning. Expert Syst Appl 96:492–505
25. Miller JF, Thomson P (2000) Cartesian genetic programming. In: Proceedings of the 3rd European conference on genetic programming, pp 121–132
26. Ratle A, Sebag M (2000) Genetic programming and domain knowledge: beyond the limitations of grammar-guided machine discovery. In: Proceedings of the 6th international conference on parallel problem solving from nature—PPSN VI, pp 211–220
27. Ferreira C (2001) Gene expression programming: a new adaptive algorithm for solving problems. Complex Syst 13(2):87–129
28. Gandomi AH, Alavi AH (2012) A new multi-gene genetic programming approach to nonlinear system modeling. Part I: materials and structural engineering problems. Neural Comput Appl 21(1):171–187
29. Moraglio A, Krawiec K, Johnson CG (2012) Geometric semantic genetic programming. In: Proceedings of the international conference on parallel problem solving from nature—PPSN XII, pp 21–31
30. Kattan A, Ong Y (2015) Surrogate genetic programming: a semantic aware evolutionary search. Inf Sci 296:345–359
31. Ffrancon R, Schoenauer M (2015) Memetic semantic genetic programming. In: Proceedings of the annual conference on genetic and evolutionary computation, GECCO, pp 1023–1030

32. Haeri MA, Ebadzadeh MM, Folino G (2017) Statistical genetic programming for symbolic regression. Appl Soft Comput 60:447–469
33. La Cava W, Silva S, Danai K, Spector L, Vanneschi L, Moore JH (2019) Multidimensional genetic programming for multiclass classification. Swarm Evol Comput 44:260–272
34. Qin AK, Suganthan PN (2005) Self-adaptive differential evolution algorithm for numerical optimization. In: IEEE Congress on evolutionary computation, vol 2, pp 1785–1791
35. Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evol Comput 10(6):646–657
36. Wang Y-J, Zhang J-S (2007) Global optimization by an improved differential evolutionary algorithm. Appl Math Comput 188:669–680
37. Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. IEEE Trans Evol Comput 13(5):945–958
38. Mallipeddi R, Suganthan PN, Pan QK, Tasgetiren MF (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. Appl Soft Comput 11(2):1679–1696
39. Wang Y, Cai Z, Zhang Q (2011) Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans Evol Comput 15(1):55–66
40. Yu W, Zhang J (2011) Multi-population differential evolution with adaptive parameter control for global optimization. In: Proceedings of the GECCO 2011, pp 1093–1098
41. Choi TJ, Ahn CW, An J (2013) An adaptive cauchy differential evolution algorithm for global numerical optimization. Sci World J 2013, article ID: 969734, 12 pages
42. Yang M, Cai Z, Li C, Guan J (2014) An improved JADE algorithm for global optimization. In: IEEE congress on evolutionary computation (CEC), pp 806–812
43. Choi TJ, Ahn CW (2014) An adaptive cauchy differential evolution algorithm with bias strategy adaptation mechanism for global numerical optimization. J Comput 9(9):2139–2145
44. Brest J, Zamuda A, Fister I, Boskovic B (2014) Some improvements of the self-adaptive jDE algorithm. In: IEEE symposium on differential evolution (SDE), pp 1–8
45. Ali WM (2014) RDEL: restart differential evolution algorithm with local search mutation for global numerical optimization. Egypt Inf J 15(3):175–188
46. Ghasemi M, Taghizadeh M, Ghavidel S, Abbasian A (2016) Colonial competitive differential evolution: an experimental study for optimal economic load dispatch. Appl Soft Comput 40:342–363
47. Parouha RP, Das KN (2016) A memory based differential evolution algorithm for unconstrained optimization. Appl Soft Comput 38:501–517
48. Sala R, Baldanzini N, Pierini M (2017) SQG-differential evolution for difficult optimization problems under a tight function evaluation budget. In: Proceedings of the international workshop on machine learning, optimization, and big data, pp 322–336
49. Trivedi A, Sanyal K, Verma P, Srinivasan D (2017) A unified differential evolution algorithm for constrained optimization problems. In: Proceedings of the IEEE Congress on evolutionary computation (CEC), pp 1231–1238
50. Draa A, Chettah K, Talbi H (2019) A compound sinusoidal differential evolution algorithm for continuous optimization. Swarm and Evol Comput 50:100450
51. Li H, Zhang L, Jiao Y (2016) Discrete differential evolution algorithm for integer linear bilevel programming problems. J Syst Eng Electron 27(4):912–919
52. Slowik A (2011) Application of adaptive differential evolution algorithm with multiple trial vectors to artificial neural networks training. IEEE Trans Ind Electron 58(8):3160–3167
53. Das S, Mullick SS, Suganthan PN (2016) Recent advances in differential evolution—an updated survey. Swarm Evol Comput 27:1–30
54. Hansen N, Ostermeier A (2001) Completely derandomized self-adaptation in evolution strategies. Evol Comput 9(2):159–195
55. Loshchilov I (2014) A computationally efficient limited memory CMA-ES for large scale optimization. In: Proceedings of the genetic and evolutionary computation conference, GECCO, pp 397–404
56. Liao T, de Octa MAM, Stutzle T (2013) Computational results for an automatically tuned CMA-ES with increasing population size on the CEC'05 benchmark set. Soft Comput 17(6):1031–1046
57. Ostermeier A, Gawelczyk A, Hansen N (1994) A derandomized approach to self-adaptation of evolution strategies. Evol Comput 2(4):369–380
58. Ostermeier A, Gawelczyk A, Hansen N (1994) Step-size adaption based on non-local use of selection information. In: Parallel problem solving from nature—PPSN III, pp 189–198
59. Arnold DV (2006) Weighted multirecombination evolution strategies. Theor Comput Sci 361(1):18–37
60. Jung JJ, Jo G-S, Yeo S-W (2007) Meta-evolution strategy to focused crawling on semantic web. In: International conference on artificial neural networks, ICANN, pp 399–407
61. Wierstra D, Schaul T, Peters J, Schmidhuber J (2008) Natural evolution strategies. In: Proceedings of the IEEE congress on evolutionary computation, pp 3381–3387
62. Glasmachers T, Schaul T, Sun Y, Wierstra D, Schmidhuber J (2010) Exponential natural evolution strategies. In: Proceedings of the genetic and evolutionary computation conference, pp 393–400
63. Liaw RT, Ting CK (2016) Enhancing covariance matrix adaptation evolution strategy through fitness inheritance. In: Proceedings of the IEEE congress on evolutionary computation (CEC), pp 1956–1963
64. Ahrari A, Deb K, Preuss M (2017) Multimodal optimization by covariance matrix self-adaptation evolution strategy with repelling subpopulations. Evol Comput 25(3):439–471
65. Beyer HG, Sendhoff B (2017) Simplify your covariance matrix adaptation evolution strategy. IEEE Trans Evol Comput 21(5):746–759
66. Akimoto Y, Auger A, Hansen N (2018) Quality gain analysis of the weighted recombination evolution strategy on general convex quadratic functions. Theor Comput Sci. https://doi.org/10.1016/j.tcs.2018.05.015
67. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3(2):82–102
68. Iwamatsu M (2002) Generalized evolutionary programming with Levy-type mutation. Comput Phys Commun 147(1–2):729–732
69. Alam MS, Islam MM, Yao X, Murase K (2012) Diversity guided evolutionary programming: a novel approach for continuous optimization. Appl Soft Comput 12(6):1693–1707
70. Das S, Mallipeddi R, Maity D (2013) Adaptive evolutionary programming with p-best mutation strategy. Swarm Evol Comput 9:58–68
71. Nan L, Xiaomin B, Shouzhen Z, Jinghong Z (2014) Social evolutionary programming algorithm on unit commitment in wind power integrated system. IFAC Proc 47(3):3611–3616
72. Gao W (2015) Slope stability analysis based on immunised evolutionary programming. Environ Earth Sci 74(4):3357–3369

73. Pang J, Dong H, He J, Feng Q (2016) Mixed mutation strategy evolutionary programming based on Shapley value. In: IEEE congress on evolutionary computation (CEC), pp 2805–2812
74. Basu M (2017) Fast convergence evolutionary programming for multi-area economic dispatch. Electr Power Compon Syst 45(15):1629–1637
75. Mansor MH, Musirin I, Othman MM (2017) Immune log-normal evolutionary programming (ILNEP) for solving economic dispatch problem with prohibited operating zones. In: 4th international conference on industrial engineering and applications, ICIEA, pp 163–167
76. Hong L, Drake JH, Woodward JR, Ozcan E (2018) A hyper-heuristic approach to automated generation of mutation operators for evolutionary programming. Appl Soft Comput 62:162–175
77. Petrowski A, Ben-Hamida S (2017) Constrained continuous evolutionary optimization. Evol Algorithms 1:93–133
78. Vanneschi L, Mussi L, Cagnoni S (2011) Hot topics in evolutionary computation. Intell Artif 5:5–17
79. Lv M, Li J, Du H, Zhu W, Meng J (2017) Solar array layout optimization for stratospheric airships using numerical method. Energy Convers Manag 135:160–169
80. Ma L, Hu S, Qiu M, Li Q, Ji Z (2017) Energy consumption optimization of high sulfur natural gas purification plant based on back propagation neural network and genetic algorithms. Energy Proc 105:5166–5171
81. Yin S, Zhu X, Qiu J, Gao H (2016) State estimation in nonlinear system using sequential evolutionary filter. IEEE Trans Ind Electron 63(6):3786–3794
82. Przewozniczek MW, Walkowiak K, Aibin M (2017) The evolutionary cost of Baldwin effect in the routing and spectrum allocation problem in elastic optical networks. Appl Soft Comput 52:843–862
83. Hiassat A, Diabat A, Rahwan I (2017) A genetic algorithm approach for local-inventory-routing problem with perishable products. J Manuf Syst 42:93–103
84. Ramos AG, Silva E, Oliveira JF (2018) A new load balance methodology for container loading problem in road transportation. Eur J Oper Res 266:1140–1152
85. Yan Y, Hong L, He X, Ouyang M, Peeta S, Chen X (2017) Pre-disaster investment decisions for strengthening the Chinese railway system under earthquakes. Transp Res Part E-Logist Transp Rev 105:39–59
86. Ascione F, Bianco N, De Stasio C, Mauro G, Vanoli G (2016) Simulation-based model predictive control by the multi-objective optimization of building energy performance and thermal comfort. Energy Build 111:131–144
87. Lin C, Fang X, Zhao X, Zhang Q, Liu X (2017) Study on energy-saving optimization of train coasting control based on multi-population genetic algorithm. In: Proceedings of the 3rd international conference on control, automation and robotics (ICCAR), pp 627–632
88. Tan MK, Chuo HSE, Chin RKY, Yeo KB, Teo KTK (2016) Optimization of urban traffic network signalization using genetic algorithm. In: Proceedings of the IEEE conference on open systems (ICOS), pp 87–92
89. Zhang Y-H, Gong Y-J, Gu T-L, Li Y, Zhang J (2017) Flexible genetic algorithm: a simple and generic approach to node placement problems. Appl Soft Comput 52:457–470
90. Reddy S (2017) Optimal scheduling of thermal-wind-solar power system with storage. Renew Energy 101:1357–1368
91. Sepahvand M, Abdali-Mahammadi F, Mardukhi F (2017) Evolutionary metric-learning-based recognition algorithm for online isolated Persian/Arabic characters, reconstructed using inertial pen signals. IEEE Trans Cybern 47(9):2872–2884
92. Bagatur T, Onen F (2018) Development of predictive model for flood routing using genetic expression programming. J Flood Risk Manag 11:444–454
93. Abkenar A, Nazari A, Jayasinghe S, Kapoor A, Negnevitsky M (2017) Fuel cell power management using genetic expression programming in all-electric ships. IEEE Trans Energy Convers 32(2):779–787
94. Chopra T, Parida M, Kwatra N, Chopra P (2018) Development of pavement distress deterioration prediction models for urban road network using genetic programming. Adv Civ Eng. https://doi.org/10.1155/2018/1253108
95. Ramli MAM, Bouchekara HREH, Alghamdi AS (2018) Optimal sizing of PV/wind/diesel hybrid microgrid system using multi-objective self-adaptive differential evolution algorithm. Renew Energy 121:400–411
96. Yao E, Wang H, Wang L, Xi G, Marechal F (2017) Multi-objective optimization and exergoeconomic analysis of a combined cooling, heating and power based compressed air energy storage system. Energy Convers Manag 138:199–209
97. Wang Y, Liu H, Long H, Zhang Z, Yang S (2018) Differential evolution with a new encoding mechanism for optimizing wind farm layout. IEEE Trans Ind Inf 14(3):1040–1054
98. Li M, Liu Y, Guo Y (2018) Shaped power pattern synthesis of a linear dipole array by element rotation and phase optimization using dynamic differential evolution. IEEE Antennas Wirel Propag Lett 17(4):697–701
99. Tian G, Ren Y, Zhou M (2016) Dual-objective scheduling of rescue vehicles to distinguish forest fires via differential evolution and particle swarm optimization combined algorithm. IEEE Trans Intell Transp Syst 17(11):3009–3021
100. Hasancebi O (2017) Cost efficiency analyses of steel frameworks for economical design of multi-storey buildings. J Constr Steel Res 128:380–396
101. Fadda E, Perboli G, Squillero G (2017) Adaptive batteries exploiting on-line steady-state evolution strategy. In: Proceedings of the European conference on the applications of evolutionary computation, LNCS, vol 10199, pp 329–341
102. Ogidan O, Giacomoni M (2017) Enhancing the performance of a multiobjective evolutionary algorithm for sanitary sewer overflow reduction. J Water Resour Plan Manag 143(7):1–9
103. He W, Qiao P, Zhou Z, Hu G, Feng Z, Wei H (2018) A new belief-rule-based method for fault diagnosis of wireless sensor network. IEEE Access 6:9404–9419
104. Hajebi M, Hoorfar A, Bou-Daher E, Tavakoli A (2018) Inverse profiling of inhomogeneous subsurface targets with arbitrary cross sections using covariance matrix adaptation evolution strategy. IEEE Geosci Remote Sens Lett 14(5):612–616
105. Emadi G, Rahmani AM, Shahhoseini H (2017) Task scheduling algorithm using covariance matrix adaptation evolution strategy (CMA-ES) in cloud computing. J Adv Comput Eng Technol 3(3):135–144
106. Yan W, Sun J, Liu Z, Hu Y (2017) A novel bi-subgroup adaptive evolutionary algorithm for optimizing degree of hybridization of HEV bus. Cluster Comput J Betwroks Softw Tools Appl 20(1):497–505
107. Gao W (2016) Inverse back analysis based on evolutionary neural networks for underground engineering. Neural Process Lett 44(1):81–101
108. Jiang L, Li Y, Liu Y, Chen C (2017) Traffic signal light control model based in evolutionary programming algorithm optimization BP neural network. In: 7th international conference on electronics information and emergency communication, pp 564–567
109. Serin E, Adali S, Balcisoy S (2012) Automatic path generation for terrain navigation. Comput Graph 36(8):1013–1024

110. Michalewicz Z (2012) Evolutionary computation and the processes of life: the emperor is naked: evolutionary algorithms for real-world applications. In: Ubiquity symposium, pp 3:1–3:13
111. Lobo D, Levin M (2015) Inferring regulatory networks from experimental morphological phenotypes: a computational method reverse-engineers planarian regeneration. PLOS Comput Biol 11(6):e1004295
112. Vanaret C, Gotteland J-B, Durand N, Alliot J-M (2013) Preventing premature convergence and proving the optimality in evolutionary algorithms. In: Proceedings of the international conference on artificial evolution, pp 29–40
113. Slowik A (2010) Steering of balance between exploration and exploitation properties of evolutionary algorithms—mix selection. In: Lecture notes in artificial intelligence, vol 6114, pp 213–220
114. Crepinsek M, Liu S-H, Mernik M (2013) Exploration and exploitation in evolutionary algorithms: a survey. ACM Comput Surv 45(3):1–33
115. Chmiel W, Kwiecien J (2018) Quantum-inspired evolutionary approach for the quadratic assignment problem. Entropy 20(10):781
116. Marjani A, Shirazian S, Asadollahzadeh M (2018) Topology optimization of neural networks based on a coupled genetic algorithm and particle swarm optimization techniques (c-GA-PSO-NN). Neural Comput Appl 29:1073–1076
117. Fajfar I, Tuma T (2018) Creation of numerical constants in robust gene expression programming. Entropy 20(10):756
118. Chen J, Zeng Z, Jiang P, Tang H (2018) Application of multi-gene genetic programming based on separable functional network for landslide displacement prediction. Neural Comput Appl 27(6):1771–1784
119. Guo Z, Yue X, Zhang K, Wang S, Wu Z (2014) A thermodynamical selection-based discrete differential evolution for the 0–1 knapsack problem. Entropy 16(12):6263–6285
120. Civicioglu P, Besdok E, Gunen MA, Atasever UH (2018) Weighted differential evolution algorithm for numerical function optimization: a comparative study with cuckoo search, artificial bee colony, adaptive differential evolution, and backtracking search optimization algorithms. Neural Comput Appl, in-press, first-online 26 October
121. Eiben AE, Smith J (2015) From evolutionary computation to the evolution of things. Nature 521:476–482

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.