Ecological Modelling 392 (2019) 179–195

Contents lists available at ScienceDirect

Ecological Modelling

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

Review

Evolutionary algorithms for species distribution modelling: A review in the context of machine learning

Sacha Gobeyn[a,*], Ans M. Mouton[a], Anna F. Cord[b], Andrea Kaim[b], Martin Volk[b], Peter L.M. Goethals[a]

[a] Ghent University, Department of Animal Sciences and Aquatic Ecology, Coupure Links 653, B-9000 Ghent, Belgium
[b] UFZ – Helmholtz Centre for Environmental Research, Department of Computational Landscape Ecology, Permoserstr. 15, 04318 Leipzig, Germany

ARTICLE INFO

Keywords:
Evolutionary algorithms
Machine learning
Species distribution modelling
Model development
Automated model selection

ABSTRACT

Scientists and decision-makers need tools that can assess which specific pressures lead to ecosystem deterioration, and which measures could reduce these pressures and/or limit their effects. In this context, species distribution models are tools that can be used to help asses these pressures. Evolutionary algorithms represent a collection of promising techniques, inspired by concepts observed in natural evolution, to support the development of species distribution models. They are suited to solve non-trivial tasks, such as the calibration of parameter-rich models, the reduction of model complexity by feature selection and/or the optimization of hyperparameters of other machine learning algorithms. Although widely used in other scientific domains, the full potential of evolutionary algorithms has yet to be explored for applied ecological research. In this synthesis, we study the role of evolutionary algorithms as a machine learning technique to develop the next generation of species distribution models. To do so, we review available methods for species distribution modelling and synthesize literature using evolutionary algorithms. In addition, we discuss specific advantages and weaknesses of evolutionary algorithms and present a guideline for their application. We find that evolutionary algorithms are increasingly used to solve specific and challenging problems. Their flexibility, adaptability and transferability in addition to their capacity to find adequate solutions to complex, non-linear problems are considered as main strengths, especially for species distribution models with a large degree of complexity. The need for programming and modelling skills can be considered as a drawback for novice modellers. In addition, setting values for hyperparameters is a challenge. Future ecological research should focus on exploring the potential of evolutionary algorithms that combine multiple tasks in one learning cycle. In addition, studies should focus on the use of novel machine learning schemes (*e.g.* automated hyperparameter optimization) to apply evolutionary algorithms, preferably in the context of open science. This way, ecologists and model developers can achieve an adaptable and flexible framework for developing tools useful for decision management.

1. Introduction

Innovations in remote sensing and micro-control units used for (near-)real-time monitoring of ecosystems are challenging ecologists to deal with a great number of data coming from different sources (Hampton et al., 2013). Machine learning plays an increasingly important role to deal with this challenge, not only in the field of ecology (Meyers et al., 2017). However, ecologists often struggle with the interpretation of models developed with novel machine learning algorithms (Araújo and Guisan, 2006). As a consequence, scientists are required to search for new approaches to increase reliability, transparency and flexibility of the models they are developing (Elith and Leathwick, 2009).

The models developed for ecological research and management with machine learning are mainly classified under species distribution models (SDMs). These models aim to define the species–environment relation and from this, estimate the species' geographical distribution. SDMs rely on the concept of an 'ecological niche', described by Hutchinson (1957). This theory conceptualizes the relation between a species' environment and its occurrence (Hirzel and Le Lay, 2008). Niche theory (Hutchinson, 1957) states that a species can only exist if the local combination of environmental gradients, the niche, allows a positive population growth rate, in the absence of immigration. In addition, the theory states that a difference in species' traits allows them

[*] Corresponding author.
E-mail address: Sacha.Gobeyn@UGent.be (S. Gobeyn).

https://doi.org/10.1016/j.ecolmodel.2018.11.013
Received 19 June 2018; Received in revised form 12 November 2018; Accepted 20 November 2018
0304-3800/

Box 1
Terminology machine learning

Machine learning: Field of research using computer programs or algorithms that have the ability to adapt or change (*i.e.* train) from an experience (data) given a performance measure.

Supervised learning: Learning from an experience by providing the machine learning algorithm with a set of inputs together with the corresponding outputs (labels).

Unsupervised learning: Learning from an experience by providing the machine learning algorithm with only inputs (and no labels) or in other words, making the algorithm search for patterns in data without having any labels to test it.

Training data: Observations of an experience used by the machine learning algorithm to train a model.

Testing data: Observations of an experience used to test the trained model. It is important to note test data are not used during the training phase.

Feature space: Set of all possible combinations of features of the input dataset. Both training and testing data are samples from the input dataset.

Feature selection or (input) variable selection: Process of selecting a subset of *relevant* features. The objectives of feature selection are to avoid the risk of overfitting by reducing model complexity (1), improve cluster detection (2) and reduce computational cost (3).

Ensemble learning: Training with multiple machine learning methods to obtain better predictive performance than by only using one machine learning method.

Model parameters: Model elements that are internal to a model, whose value can be estimated with data and are context-dependent. The action of estimating parameters is referred to as 'parameter estimation', in which a unique set of model parameter values are estimated with data.

Hyperparameters: Machine learning algorithm free options that need to be set beforehand, determining the training strategy and related efficiency of the algorithm. The settings of these hyperparameters can influence an algorithm's and a model's performance. In this review, hyperparameters refer specifically to parameters related to the algorithm free options, whereas 'parameters' refer to model elements. To clarify the difference between parameters and hyperparameters, a number of examples are given in Section 3.2. For a good mathematical introduction to hyperparameters and their role in machine learning, we refer to Bergstra and Bengio (2012).

Evolutionary algorithms (EAs) (or evolutionary computing): Metaheuristic search algorithms (strategy) inspired by processes observed in evolution, *i.e.* selection, crossover and mutation.

to occupy a different niche and coexist in a given spatial unit. To inspect these species' traits, machine learning algorithms are used to classify a species as present, or absent, given the environmental conditions. Often a probabilistic framework is used to express the chance for a species to occur. The environmental conditions are quantified in a number of environmental features, for instance, temperature, precipitation, soil moisture (He et al., 2015). The number of input features, after preprocessing, reach up to 20, while studies using over 30 features are found in literature (Bennetsen et al., 2016). Machine learning is used to train a model estimating a response variable, species occurrence, based on these environmental predictors.

Species distribution model development with machine learning embeds little ecological hypotheses in the training process as the machine learning algorithms primarily aim to uncover patterns in data (Saeys et al., 2007; Mount et al., 2016). At first instance, this development could be considered as "black box" modelling, which is the case in a number of techniques, *e.g.* artificial neural networks. Yet, there are techniques, such as decision trees, that present interpretable models. In these models, the user can interpret why estimations were done in that way. For example, in decision trees, a set of hierarchical rules can be analysed that lead to an estimated species presence. In specific applications, for instance in freshwater management, embedding hypotheses and assumptions is of major importance to preserve ecological interpretability of the developed models (Adriaenssens et al., 2004). Also, incorporating species dispersal and interactions in the next generation of species distribution models (SDMs) requires a hypothesis-driven approach. Evolutionary algorithms (EAs), a collective of machine learning techniques inspired on the concept of evolution, allow embedding these hypotheses by separating model performance evaluation from solution searching (Rauch and Harremoës, 1999). In these EAs, models with parameters and input variables are encoded in so-called 'chromosomes'. Specific algorithm functions, called genetic operators, are applied to these chromosomes to search for well-performing models.

Because EAs algorithms offer this transparency and flexibility, we evaluate in this synthesis the role of EAs as a machine learning method for species distribution modelling. We aim to answer the following questions: What is the current role of EAs in species distribution modelling (Section 2)?; How do EAs work and what are their specific strengths (Section 3)?; What opportunities do these EAs have in the field of species distribution modelling (Section 4)?; What guidelines can be given to applying an EA or another metaheuristic as a machine learning method to identify transparent and accurate models (Section 5)? It is important to note that this paper focuses on EAs, having in mind that other metaheuristics such as particle swarm optimization, ant colony optimization and simulated annealing also exist. For an overview of these other methods, we refer to supportive information 1. Differences between EAs and other metaheuristics are discussed throughout this manuscript. For an extensive description of EAs and other metaheuristics, we refer readers to Gendreau and Potvin (2010) and Kacprzyk and Pedrycz (2015).

2. Machine learning in species distribution modelling

Many ecological researchers rely on machine learning for the development of SDMs. The use of machine learning has introduced new concepts important for ecologists to understand. In contrast to previous reviews discussing the development of SDMs (*e.g.* Guisan and Thuiller, 2005; Araújo and Guisan, 2006; Austin, 2007), this review focusses on SDM learning and its technical challenges, rather than development through ecological reasoning. That is why we focus on the discussion of EAs in the context of machine learning.

Machine learning can be defined as the field of research using computer programs or algorithms that have the ability to adapt or

Table 1
Machine learning approaches developed and used in species distribution modelling. In the table, WOS = web of science, and C.n. = cumulative number of publications. A hyphen in the first column indicates that no acronyms/full names are found in the literature.

Approach/technique (acronym)	Short description	C.n. in WOS (08/11/2017)	Notable references
Artificial neural networks (ANNs)	Non-linear mapping structures inspired on the biological system of the brain.	2000: 6; 2010: 77; 2017: 166	Fukuda et al. (2013); D'heygere et al. (2006)
BIOCLIM (–)	Delineates a rectangular environmental (bioclimate) hyperspace (or envelope) to estimate the response of species to a number of bioclimatic input variable.	1990/2000: 2; 2010: 1; 2017: 33	Carpenter et al. (1993); Elith et al. (2006)
Biodiversity modelling (BIOMOD)	Ensemble modelling platform/software combining several techniques.	2010: 11; 2017: 53	Thuiller (2003); Thuiller et al. (2009)
CLIMEX (–)	Model based on GROWEST, using a growth and stress index	sporadicaly used before 1990	Sutherst and Maywald (1985)
Decision trees (DT)	Classifiers expressed as a recursive partition or tree of the feature space.	2000: 5; 2010: 126; 2017: 421	Iverson and Prasad (1998)
Fuzzy logic (FL)	Method that deals with linguistic uncertainty by generalizing classical logic.	2000: 2; 2010: 20; 2017: 61	Adriaenssens (2004); Van Broekhoven et al. (2006)
Genetic algorithm for rule set production (GARP)	EA-inspired method used to produce a rule-bank SDM.	2010: 12; 2017: 224	Peterson et al. (2002); Stockwell and Noble (1992)
Generalized linear models (GLMs)	Collection of parametric techniques based on a random component, a systematic component, and a link function describing a relation between the former the random and systematic component.	2000: 23; 2010: 230; 2017: 600	Nelder and Wedderburn (1972); Zuur et al. (2010)
Generalized additive models (GAMs)	Extension of GLMs which relate the response variable to a linear combination of smoother functions.	2000: 43; 2010: 313; 2017: 738	Zuur et al. (2010)
GROWEST (–)	Model using a growth index based temperature, light, moisture	Sporadicaly used before 1980	Nix et al. (1977)
Maximum entropy method (Maxent)	Technique using the principle of maximum entropy to make predictions from incomplete knowledge.	2010: 145; 2017: 1391	Phillips et al. (2006); Elith et al. (2011)
Random forest (RF)	Technique using bootstrap aggregation to create a set of decision trees.	2010: 27; 2017: 283	Prasad et al. (2006); Cutler et al. (2007)

change from an experience (data) given a(n) performance or objective measure. Algorithms to facilitate machine learning are widely applied in many scientific fields such as artificial intelligence, telecommunication and engineering of electronics (web of science, accessed on 13/09/2018). Machine learning algorithms can be used to train models with data so these models can make as good as possible predictions on new, 'unseen', data. Machine learning algorithms can be categorized based on whether output labels are (not) used for training, *i.e.* (un-)supervised learning (Box 1). To guide the readers, Box 1 shows a number of definitions used in the field of machine learning, also used in this review paper.

In species distribution modelling, supervised learning is typically applied to classify species occurrence in geographical, and possibly the temporal, dimensions. To train SDMs, binary labelled data (species presence or absence) together with environmental input data are used by the machine learning algorithm. In Table 1, an overview of methods used in species distribution modelling are shown, together with a short explanation, the cumulative number of papers mentioning the method (web of science, accessed 08/11/2017), and key references. In addition, in Fig. 1, the first report of the method in scientific literature is shown. The remainder of this section aims to shortly introduce these methods to guide readers to the most used ones. Acronyms of these methods can also be found in Table 1.

Generalized linear and adaptive models (respectively GLMs, and

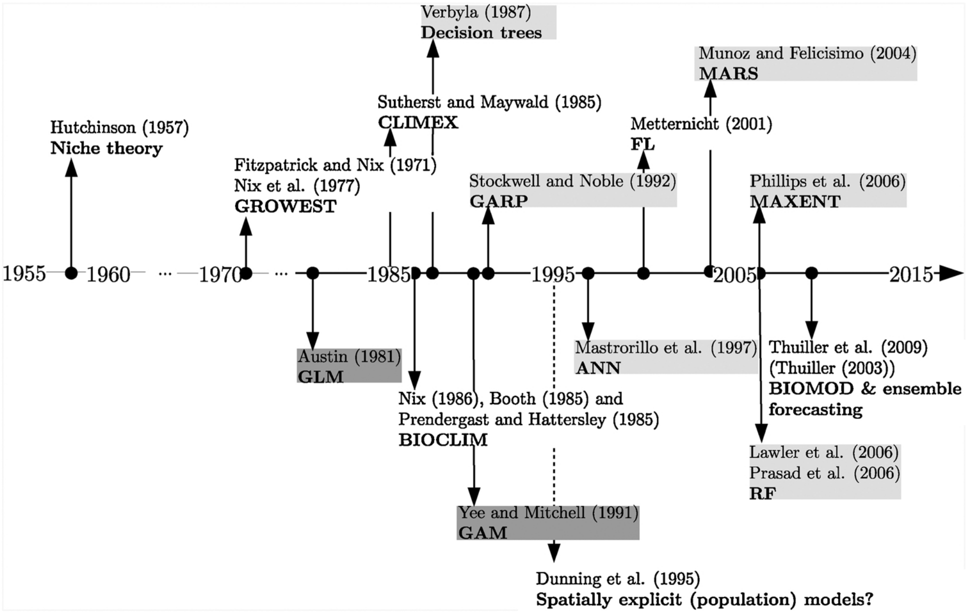

Fig. 1. Chronological overview of first use or report of SDM modelling technique. Methods classified before 2000 under machine learning are indicated in a light grey boxes. Other methods that are often categorized under machine learning are shown in a dark grey boxes. This graph is a result of a web of science search on 08/11/2017 based on an intersection of literature on the shown methods and the field of 'species distribution modelling' (see supportive information 2). The acronyms found in this figure are listed in Table 1.

GAMs) and decision trees are the first machine learning techniques used in species distribution modelling (Fig. 1). GLMs (and GAMs) are a collection of (semi-)parametric techniques based on three elements: a random component that assumes a probability distribution of a response variable (1), a systematic component specifying linear combination of the explanatory variables with their respective slopes (2) and a link function describing the relation between the random and systematic component (3) (Nelder and Wedderburn, 1972). Decision trees are classifiers expressed as recursive partitions or trees of the feature space (Rokach and Maimon, 2015). These are tree-like representations of a rule-induction, *i.e.* a set of 'if-then' rules that are followed leading to a (probability of) species presence or absence. Different algorithms are available to develop decision trees, such as CART (Breiman et al., 1984) and C4.5 (Quinlan Ross, 1993), using a Gini index and entropy measure, respectively. A robust approach based on decision trees is a random forest (RF) (Breiman, 2001). RF uses bootstrap aggregation to generate on a number of decision or regression trees. In bootstrap aggregation, several bootstrap samples from a training data set (objects in the instance space) are taken to develop a number of models. RF is used often in species distribution modelling, and has shown to be an interesting technique to model complex systems, including species interactions (Vezza et al., 2015).

The genetic algorithm for rule set production (GARP) is an EA-inspired method used to produce a rule-bank SDM (Stockwell and Noble, 1992). A rule-bank SDM is a model based on a set of hierarchical rules estimating species presence or absence. In this sense, the model structures obtained with GARP are similar to those obtained with decision trees. GARP is the first SDM software package using EAs, knowing a number of successful applications, for instance, to estimate the effect of global change on species distributions (Peterson et al., 2002). Another technique often used in combination with EAs are artificial neural networks (Ding et al., 2013). Artificial neural networks which are nonlinear mapping structures inspired by the biological system of the brain have been successfully used for freshwater applications (Goethals et al., 2007; Muñoz-Mas et al., 2017). From these two examples, it is clear that EAs are often used to support the development of the model structure, *i.e.* search for the most optimal model structure. Despite their success, the popularity of artificial neural networks and also GARP is recently declining compared to that of maximum entropy modelling (Maxent) (Table 1). Introduced in 2006, Maxent uses the principle of maximum entropy to make predictions from incomplete knowledge (Phillips et al., 2006). The maximum entropy method (Maxent) approach uses the principle of maximum entropy to make predictions from incomplete knowledge. The principle of maximum entropy states that the best approximation to an unknown distribution, given a number of constraints, is the distribution which only satisfies these constraints and no others. In other words: Maxent aims to model everything that is known (constraints) but assumes nothing about what is unknown. Maxent is currently the most used package to train SDMs (200+ papers in 2016, based on abstract search, web of science, 08/11/2017). Its theoretical basis, the default use of regularisation (*i.e.* penalize model complexity), flexibility and performance are the main factors explaining Maxent's popularity (Elith et al., 2011). Model complexity can also be penalized in many other algorithms, however, Maxent was the first approach to include it by default. This default inclusion stressed the importance of regularization among users. Peterson et al. (2007) compared the transferability (test on unseen data) of Maxent and GARP and showed that both have their specific advantages. This is not surprising as the 'No Free Lunch Theorem' (Wolpert and Macready, 1997) depicts that no algorithm will work well on all problems. A good approach to deal with the 'No Free Lunch Theorem' is to combine and/or compare several approaches in one modelling effort. Thuiller (2003) developed a platform for implementing different techniques. His initial aim was to present a framework able to simultaneously fit different models to data. It was only six years later that Thuiller et al. (2009) presented a new version of BIOMOD, including the concepts of uncertainty estimation, and ensemble forecasting (Araújo and New, 2007). A weakness of the BIOMOD is that the platform is bound by specific implementations of techniques. Golding et al. (2017) deals with this issue by implementing a modular framework for species distribution modelling. He argues that algorithm success is partly depicted by method transparency empowered by clear encoding and guidelines of use, *i.e.* which method and specific settings are suitable to solve the problem at hand? It is important to note that besides algorithm guidelines, also appropriate data cleansing techniques can considerably improve results. An excellent guide for data cleansing of ecological data is given by Zuur et al. (2010).

Other techniques are available but are not categorized under machine learning as their origins are rooted in niche theory of Hutchinson (1957). These methods, specifically GROWEST (Nix et al., 1977), CLIMEX (Sutherst and Maywald, 1985) and BIOCLIM (Booth, 1985; Nix, 1986) were used in the early days of mapping a species' niche (Fig. 1). They are currently less used because of their simplicity, lack of accuracy and inability to account for variable interaction (Booth et al., 2014) (Table 1). Another less-used approach is the development of models with fuzzy logic. Fuzzy logic models allow integrating expert knowledge in their model structure. Specifically, fuzzy models allow reflecting uncertainty present in linguistic information (Adriaenssens et al., 2004). Although fuzzy logic model development might not be classified under machine learning, it is often used in conjunction with a machine learning algorithm (Chen et al., 2003).

The methods discussed above generally only consider the species-environment relationship to estimate species occurrence. The spatial structure of the relations is implicitly included in the scale. Spatially explicit models, which incorporate the spatial space in their structure, allow describing processes such as migration and dispersal in a spatial context (DeAngelis and Yurek, 2017; Dunning et al., 1995). Up until today, these explicit methods are less popular, mainly because of their complexity, and need for detailed information to parametrize them (DeAngelis and Yurek, 2017). Given these disadvantages, spatially explicit models do hold a lot of potential to help uncover species behaviour and distribution as a function of environmental pressures.

Although the role of GARP is recently declining, a number of specific applications of EAs are observed in the literature since 2000 (see Table 2). These novel techniques are mainly applied in the context of freshwater management and used to estimate a link between river modification and freshwater species occurrence. They facilitate feature selection (feature selection, for definition, see Box 1) for artificial neural network and decision tree models (D'heygere et al., 2006) or to estimate model parameters of fuzzy logic SDMs (Fukuda et al., 2011). The results of our literature review (for methodology, see supportive information 2) show that these algorithms are often 'tailor-made', and characterized by a specific algorithm formulation. In the next section, we discuss this 'EA-literature' with the aim to identify which specific applications and workflows are mainly adopted. Before we do so, we introduce the basic principles of evolutionary algorithms.

3. Evolutionary algorithms

3.1. Introduction to evolutionary algorithms

EAs aim to solve complex problems by incorporating elements of structured randomness in their search behaviour motivated by principles in evolution such as selection, mutation and crossover (Maier et al., 2014). EAs distinguish from single-point based methods by iterating a population of candidate solutions to an optimum. These solutions are quantified in a fitness mimicking the evolutionary concept.

EAs have been successfully applied to solve specific problems in

Table 2

Overview of literature review. The ‘subject of training’ is either parameter estimation, feature selection and/or hyperparameter optimization. Parameter estimation refers to the estimation of a unique set of model parameter values (see also Box 1). Hyperparameter optimization refers to the search for values of algorithm settings which influence an algorithm's performance. ‘Training robustness’ indicates the robustness of the algorithm towards different samples of the data (by cross-validation, or bootstrapping) whereas ‘algorithm robustness’ is the robustness of the algorithm tested on the same data sample. The acronyms found in the column ‘objective function’ are found in Table 3. * = or no improvement 50 generations. ‘?’ = information was unclear or uncertain. Prev. = prevalence, *i.e.* number of species occurence over number of samples. x = number of input features.

Author; ecosystem	Prev.	subject of training	Objective functions	Type of EA and operators	Problem size	Hyperparameters	Hyperparameter optimization	Training robustness	Algorithm robustness	Resampling scheme

D’Angelo et al. (1995); freshwater	?	Parameter estimation; feature selection	ρ;
			SSE
				Genetic algorithms genetic programming; mixed string genotype	$\pm 10^{10}$	# chromosomes: 200;
						crossover rate: 0.8;
						mutation rate: 0.1;
						# generations: 20000
							Iterative	no	10 runs	No

Whigham (2000); terrestrial	?	Feature selection; parameter estimation	?	Genetic programming; tree-genotype	?	?	?	?	100 runs	?

McKay (2001); terrestrial	?	Feature selection; parameter estimation	?	Genetic programming; tournament selection; half ramped initialization; tree-genotype	$\pm 10^{10}$	# chromosomes: 50;
						crossover rate: 0.9;
						mutation rate: 0.1;
						# generations: 50
							No	?	?	?

D’heygere et al. (2003); freshwater	?	Feature selection	CCI	Simple genetic algorithm; roulette wheel selection; binary string	2^{15}	# generations: 20;
						crossover rate: 0.6;
						mutation rate: 0.033;
						# generations: 20
							?	10-fold cross-validation	?	?

McClean et al. (2005); terrestrial	?	Parameter estimation	AUC	Bayesian genetic algorithm; continuous string	$\pm 10^{2*9}$	?	?	?	?	?

D’heygere et al. (2006); freshwater	?	Feature selection	CCI	Simple genetic algorithm; roulette wheel selection; binary string	2^{17}	# chromosomes: 20;
						crossover rate: 0.6;
						mutation rate: 0.03;
						# generations: 40
							Iterative	10-fold cross-validation	x runs	Stratified

Termansen et al. (2006); terrestrial	?	Parameter estimation	AUC	Bayesian genetic algorithm; continuous string genotype	10^{2*9}	# chromosomes: 100; crossover rate: ?; mutation rate: < 0.01; # generations: 90	Iterative	?	x runs	?

Van Broekhoven et al. (2007); freshwater		Parameter estimation	% CFCI	Simple genetic algorithm; tournament selection; elitism; binary string and continuous	?	# chromosomes: 100;
						crossover rate: 0.95;
						mutation rate: ~ 1/(length chromosome) (< 0.01);
						# generations: 1000*
							iterative	?	100 runs	?

Fukuda and Hiramatsu (2008); freshwater	?	Parameter estimation	MSE	Simple genetic algorithm; binary string	?	?	?	10 runs	?	
Fukuda (2009); freshwater	?	Parameter estimation	MSE	Simple genetic algorithm; elitism	?	?	?	50 runs	?	
Tirelli and Pessani (2009); freshwater	0.7	Feature selection	Kappa (?)	?	?	?	?		?	
Hoang et al. (2010); freshwater	0.12 to 0.72	Feature selection	CCI; Kappa	Binary string	2^{21}	?	?	3-fold cross-validation	5 runs	No

Favaro et al. (2011); freshwater	0.56	Feature selection	Sn; Sp;
			CCI;
			Kappa;
			AUC
				?	?	?	?	10-fold cross-validation	?	?

Fukuda et al. (2011); freshwater	0.27	Parameter estimation	MSE	Simple genetic algorithm;	2^{4*35}	# chromosomes: 100; crossover rate: ?;	?	3-fold cross-validation	20 runs	Stratified based on prevalence

(*continued on next page*)

Table 2 (*continued*)

Author; ecosystem	Prev.	subject of training	Objective functions	Type of EA and operators	Problem size	Hyperparameters	Hyperparameter optimization	Training robustness	Algorithm robustness	Resampling scheme

				roulette wheel selection;
				elitisme;
				binary string
						mutation rate: 0.05;
						# generations: 5000				

Jeong et al. (2011); terrestrial (marine)	?	Feature selection; parameter estimation	RMSE	Genetic algorithm genetic programming; tree-genotype	± 10^{10}	# chromosomes: 200;
						crossover rate: 0.6–0.9;
						mutation rate: 0–0.3;
						# generations: 100
							?	Bootstrapping	?	80 % training data
										20 % test data

Fukuda et al. (2012); freshwater	0.5	Parameter estimation	MSE	binary string	2^{4*17}	# chromosomes: 100; crossover rate: ?; mutation rate: ?; # generations: 2000	?	5-fold cross-validation	20 runs	?

Zarkami et al. (2012); freshwater	?	Feature selection	CCI;
			Kappa
				Simple genetic algorithm;
				binary string
					?	# chromosomes: 20;
						crossover rate: 0.6;
						mutation rate: 0.033;
						# generations: 20
							?	3-fold cross-validation	?	?

Boets et al. (2013); freshwater	0.38	Feature selection	Kappa;
			AUC;
			CCI
				Simple genetic algorithm;
				tournament selection;
				binary string
					2^{11}	# chromosomes: 20
						crossover rate: 0.6;
						mutation rate: 0.033;
						# generations: 20
							Iterative	3-fold cross-validation	?	Random

Boets et al. (2013); freshwater	0.29	Feature selection	Kappa;
			AUC;
			CCI
				Simple genetic algorithm;
				tournament selection;
				binary string
					2^{11}	# chromosomes: 20;
						crossover rate: 0.6;
						mutation rate: 0.033;
						# generations: 20
							iterative	3-fold cross-validation	?	Random

Sadeghia et al. (2013); freshwater	?	Feature selection	?	Simple genetic algorithm;
				binary string
					± 10^{33}	# generations: 20;
						crossover rate: 0.6;
						crossover rate: 0.033;
						# generations: 20
							Iterative	3-fold cross-validation	x runs	?

Sadeghi et al. (2014); freshwater	?	Feature selection	?	Binary string	± 10^{33}	?	?	4-fold cross-validation	5 runs	?
Zarkami et al. (2014); freshwater	0.5	Feature selection	CCI; Kappa	Simple genetic algorithm; binary string	2^{10}	# chromosomes: 20; crossover rate: ?; mutation rate: ?; generations: 20	?	3-fold cross-validation	?	Random

Muñoz-Mas et al. (2016a); freshwater	0.37	Feature selection; hyperparameter optimization (decision tree)	TSS	GA with a derivative quasi-Newton method; mixed string genotype	± 10^{2x}	# chromosomes: 500;
						crossover rate: 0.75;
						mutation rate: 0.75;
						# generations: 500
							?	3 times 3-fold cross-validation	?	Stratified based on prevalence

Muñoz-Mas et al. (2016); freshwater	0.62
	0.66
		Feature selection	TSS penalty complexity	GA with a derivative quasi-Newton method; mixed string genotype	?	# chromosomes: 1000;
						crossover rate: 0.75;
						mutation rate: 0.75;
						# generations: 1000
							?	3 times 3-fold cross-validation	?	?

Vayghan et al. (2016); freshwater	0.57	feature selection	?	Binary string	2^{9}	# generations: 20;
						crossover rate: 0.6;
						mutation rate: 0.033;
						# generations: 20
							?	?	?	?

Gobeyn and Goethals (2017); freshwater	?	Feature selection parameter estimation	AUC
			Kappa
				Simple genetic algorithm
				tournament selection
				variable length binary string
					> 2^{112}	# chromosomes: 100;
						crossover rate: 1.;
						mutation rate: 0.05;
						# generations: 50 to 2000
							iterative
							and guidelines of Gibbs et al. (2008)
								bootstrapping	?	?

Muñoz-Mas et al. (2017); freshwater	0.21
	0.42
		Feature selection	TSS stimulated overprediction	GA with a derivative quasi-Newton method; binary string	?	# chromosomes: function of ensemble size;
						crossover rate: 0.6;
						mutation rate: 0.6;
						# generations: function of ensemble size
							iterative?	?	?	Stratified based on prevalence

water resources management (Maier et al., 2014), astrophysics, bioinformatics (Pal et al., 2006; Sirbu et al., 2010) and software engineering (Eiben and Smith, 2015). Even more, the use of EAs in artificial intelligence is pushing advances in evolving digital objects (software) towards physical embodied artificial evolution (*i.e.* hardware, robots, 3D-printers). Numerous examples exist and the number of applications is expected to increase in the coming years (Eiben and Smith, 2015). EAs have mainly been used as a machine learning method, also to train other methods such as artificial neural networks and decision trees. Their success can be explained by a number of reasons (Eiben and Smith, 2015; Maier et al., 2014):

1 EAs are assumption-free which make them generally applicable and easily transferable to other problems.
2 They are flexible and can easily be used in combination with other methods, for instance, other search methods or fuzzy logic.
3 They are capable to solve complex problems without the need for model simplification often required by traditional optimization methods. Moreover, they are able to uncover less obvious or even unexpected patterns.
4 The found solutions with EAs allow for an in-depth analysis since a number of near-optimal solutions are generated.

EAs iterate a population of chromosomes over a number of generations with genetic operators, *i.e.* selection, crossover and mutation (Box 2 and Fig. 2, panel A). This process is inspired by the concept of evolution where genetic information and characteristics in a population are passed on generation by generation. The chromosome is the algorithms' building block storing the formulation and performance of a candidate solution to a problem, *i.e.* the genome and fitness. The fitness can be defined as a quantification measure of how good a solution to a problem is. The formulation of the candidate solution is stored in a specific data type, also called genome. For genetic algorithms (GAs), binary or real-valued strings are coded as data type whereas tree-like structures are used for genetic programming (GP) (Fig. 2, panel B). Both GAs and GP are classified under evolutionary algorithms. Other types of EAs, such as evolutionary programming and evolution strategies exist (Weise, 2009). GAs differ from other evolutionary algorithms in the way they are designed as problem-independent solvers, whereas other EAs are designed and implemented to solve specific problems. We consider GP to be developed for specific problems. The conceptual difference between GAs and evolutionary programming is that the basic object is considered to be a species rather than a chromosome. In evolutionary programming, recombination (crossover) is often not considered. Evolutionary strategies show resemblance to real-valued GAs, but with a focus on the selection and mutation operator. These problem-specific techniques are used in species distribution modelling, but not often. In the next section, we explore the use of this problem-(in)dependent EAs.

The initialisation of a population with a number of chromosomes (population size, *PS*) and their genomes is the first step. For binary string GAs, this consists of creating a random string of bits (example lower left panel, Fig. 2) with every bit either having the value zero or one. For real-valued strings, a uniform value within a defined interval is chosen for every bit. After initialisation, the fitness is evaluated by mapping the genome to a model with a mapper function. As an example, for feature selection in SDMs, a '011' string is translated to the exclusion of the first feature and inclusion of the last two in the model (D'heygere et al., 2006). For parameter estimation, a binary string is translated to an integer or decimal (respectively '011′ → (0*1 + 1*2 + 1*4) → 6 or 1/6) for the value of the model parameters (Van Broekhoven et al., 2007). The models are then evaluated with a user-defined objective function and training data leading to a fitness value. Usually, a measure of agreement between the model output and training data is calculated. After fitness evaluation, selection, crossover and mutation operators are applied to the population. The selection operator selects a number of chromosomes from the population as parents to generate offspring based on their fitness value and a selection procedure (*e.g.* tournament selection, roulette wheel selection). In tournament selection, tournaments are organised in which two candidate parents are (randomly) selected and the candidate with the highest fitness is selected as a parent. For roulette wheel selection, parents are selected with a chance proportional to their relative fitness. The crossover operator generates offspring by inheriting a part of the parents' genomes. For instance, in a GA one-point crossover operator, a position in the genome is randomly chosen as a breakpoint. The parents' substrings are then combined to form a new string for the offspring (Fig. 2, lower panel). A crossover rate determines the probability that crossover between parents occurs. The mutation operator changes the values in random positions in the genomes (or alleles) of the offspring with a rate equal to the mutation rate. After the application of the three operators, the fitness of the new chromosomes is evaluated. Next, a new generation is produced by applying the before-introduced operators. This procedure is repeated until a certain stopping criterion is met. Typically, this criterion is a maximum number of generations or a fitness convergence criterion.

Besides the context of use of GAs and GP, the way of problem encoding and consequently the implementation of the crossover and mutation operator is different (Mcdermott et al., 2015; Rowe, 2015). For GAs, an example of a crossover of two binary strings with a one-point uniform operator is shown in Fig. 2 (lower left panel). Here, a random number between two (one) and the length (length minus one) of the two parents' genome is chosen and before (after) this position a breakpoint is appointed. The genome for the first offspring is formed by merging the part before and after the breakpoint of parent one and two, respectively. Similarly, for the second offspring the parts before and after the breakpoint are used but now the genomes of parent two and one are used. For GP (Fig. 2, lower right panel), breakpoints are chosen between nodes of the tree and these are switched between the parents' genomes. For mutation in a binary string, a random position is chosen and the value for the allele at that position is flipped to the other value (0 → 1 or 1 → 0). In case of real-valued strings, a new random value bounded by a predefined interval is chosen at a random position. For tree-like structures, a random terminal or non-terminal node is chosen and replaced with a terminal node or random initiated subtree (see Fig. 2, lower right panel) (Mcdermott et al., 2015).

3.2. Application to species distribution modelling

In order to obtain an insight into the use of EAs in species distribution modelling, literature abstracts were scanned in the web of science catalogue. The followed methodology to conduct this literature review can be found in supportive information 2. Here, implementations that differ from GARP are discussed, as these implementations vary as a function of the context of the problem. The results of this literature review are shown in Table 2. In this table, we make a clear distinction between 'parameter estimation' and 'hyperparameter optimization'. Parameter estimation refers to the estimation of a unique set of model parameter values (Box 1). With respect to species distribution and ecological modelling, this implies that model parameters that describe the limits of a species' environmental range are estimated, *e.g.* what are the threshold river temperatures in which a fish can survive? Or what are temperature tipping points at which species reproduction declines? As such, parameters are an element of the SDM. Hyperparameter optimization refers to the search for values of algorithm settings which influence an algorithm's performance. In other words, a hyperparameter can be considered as an algorithms' free option available for the user. The number of neurons and hidden layers are examples of

Box 2
Terminology EAs

Phenotype: Candidate solution to a problem, here represented by a model.

Genotype: Representation of a phenotype in a data type. Typically used genotypes are binary, real-valued strings or tree structures.

Genome: A specific formulation of the genotype (*e.g.* 1111011010).

Allele: A single element of the genotype (*e.g.* one bit).

Fitness: A measure of how good a solution to a problem is.

Chromosome: Object containing a genome and fitness.

Mapper: User-defined function which translates the genotype to phenotype.

Selection: Process of selecting chromosomes as parents for crossover, typically based on their fitness values.

Crossover: Process of combining the parents' genome to form genomes for the offspring.

Mutation: Process of randomly altering the parts of the genome.

Genetic algorithm: Evolutionary algorithms that use selection, crossover and mutation operators to solve an optimization problem. An explicit difference between GAs and other EAs is that GAs are designed as problem-independent solvers, whereas other EAs are designed to solve specific problems.

two hyperparameters that need to be set in order to develop an artificial neural network. Or for a RF, on has to set the maximum depth of a tree and the number of trees. As such, values for hyperparameters can be considered as choice elements of the algorithm, and can thus not be directly estimated with data. In the context of Table 2, 'hyperparameter optimization' in the column 'subject of training' refers to the action of using an EA to perform hyperparameter optimization of another machine learning approach. It is important to differentiate between the column 'subject of training' from the columns 'hyperparameters' and 'hyperparameter optimization', as the latter two refer to (the setting of) hyperparameters of the EA itself.

Table 2 shows the general characteristics of studies in which EAs are applied: generally they are applied in freshwater management to estimate model parameters, perform feature selection or hyperparameter optimization of other machine learning techniques. For 14 of the 27, an EA is used for only feature selection whereas in seven studies for only parameter estimation. In five studies, an EA is used for parameter estimation and feature selection and in the remaining study, an EA is used for feature selection and optimization of hyperparameters of a decision tree. In case of feature selection, the EAs are used as wrapper methods for other methods; artificial neural networks, *e.g.* D'heygere et al. (2006), and decision trees, *e.g.* Boets et al. (2013). In this approach, the genomes are translated to features for another machine learning technique fitting the response patterns to environmental conditions. In the case of parameter estimation, the EAs are used to estimate the model parameter values of fuzzy logic models, *e.g.* Fukuda (2009), Van Broekhoven et al. (2007). In other words, the model parameters describing suitability range of environmental conditions for a species are estimated.

A noteworthy observation is that 22 papers presented in Table 2 situate within the domain of freshwater science. The data used in these case studies are often characterized by a high degree of uncertainty and noise, and observation bias, *i.e.* more (less) presence instances are available than absence (see column prev. in Table 2). The latter, causing a bias in model training (Mouton et al., 2010), is the reason why a number of classification measures are typically used in these studies. Often-used measures are listed in Table 3, together with their acronyms. In these measures, species occurrence estimated by the classifier is tested to observations. According to the study objectives, and available data, a set of measures is selected and analysed, each weighting a degree of correct estimation of species presence, on the one hand, and absence, on the other (Mouton et al., 2010). Non-binary measures, such as the (root) mean of squared error, correlation and sum of squared errors, are used for regression. In these cases, the probability of occurrence is not estimated, but species numbers (D'Angelo et al., 1995) or density (Fukuda, 2009). In addition, the (root) mean square error and linear correlation is used. In one case, the mean squared error between the non-classified preference (between 0 and 1) and observed presence/absence is computed (Fukuda et al., 2012). In another case, models are penalized for their complexity (Muñoz-Mas et al., 2016a). The trade-off between omission and commission errors are never explicitly considered in model training, although they are considered implicitly by weighting objectives. Assumed prevalence-independent measures like Cohen's Kappa or the true skill statistic are used to cope with this trade-off, however, there is no agreement whether these are truly prevalence independent (Mouton et al., 2011). In three of the studies reported in Table 2, the training data are stratified by sampling an equal number of presence and absence instances in order to deal with this prevalence dependency (Mouton et al., 2009).

A number of different implementations of EAs have been used in species distribution modelling. Simple genetic algorithms are generally used and are considered problem-independent. These algorithms apply a GA with uniform crossover and mutation operators, in conjunction with a tournament selection operator (*e.g.* Boets et al. (2013)). Derivative methods have been used in combination with GAs allowing to improve the local search performance of the GAs (Muñoz-Mas et al., 2016; Muñoz-Mas et al., 2017). In addition, GPs are used, but only limited (Jeong et al., 2011; McKay, 2001; Whigham, 2000). Another interesting application is the use of Bayesian theory in GAs (McClean et al., 2005; Termansen et al., 2006). Feature selection is always implemented in binary strings whereas binary and continuous strings are used for parameter estimation. Crossover rates vary from 0.6 to 0.95 whereas mutation rate are generally lower, between 0.1 and 0.3, with the exception of 0.6 (Muñoz-Mas et al., 2017) and 0.75 (Muñoz-Mas et al., 2016). A number of 20–200 chromosomes are reported to iterate over generally 20–100 generations. However, a larger number of generations (≥ 1000) are observed in four studies. Selection rates are never reported, as these are typically equal to 50%.

Model robustness is tested by applying cross-validation and repeated learning with the same or different samples of the data. Cross-validation is generally used (14 of 27 cases) to test robustness. In this approach, the data are partitioned in a number of samples, i.e. folds. Next, the model is identified with $n - 1$ folds and validated with the remaining fold. In a number of publications, the EA analysis is repeated a number of times with the same data starting from different initial populations in order to test the robustness of the EA (see D'Angelo et al. (1995), Fukuda (2009)). This is because the obtained near-optimal solution might not be equal in every EA run since the search behaviour is characteristic by random choices. An interesting application of this repeated EA analysis is the multilayer perceptron ensembles (a type of

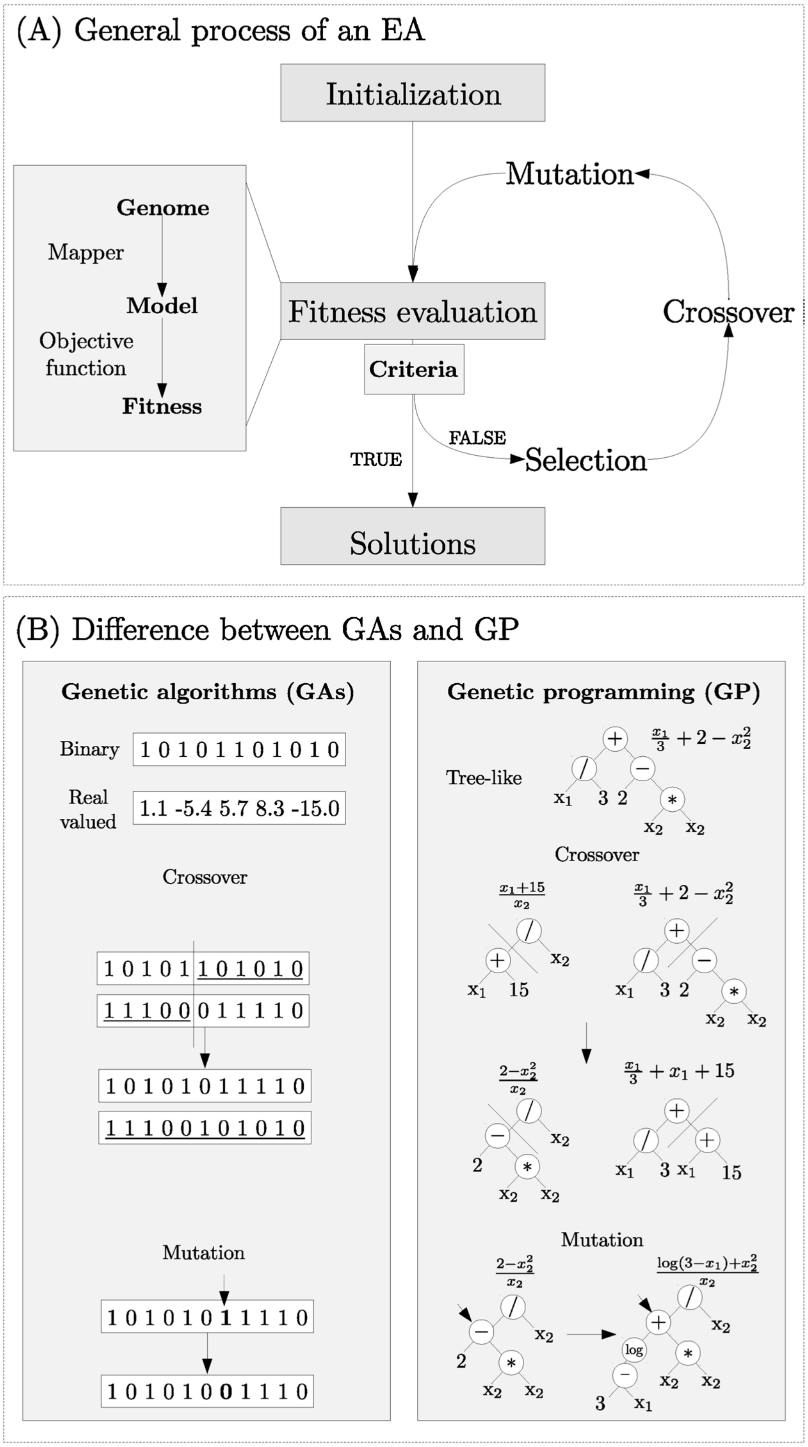

Fig. 2. General process of iterating a number of chromosomes in an EA with a number of operators (panel A), and the difference between GAs and GP (panel B). The scheme in the upper panel is applicable for both GAs and GP, for which the encoding and way the operators are implemented differ (lower left and right panel respectively). + = plus operator, − = minus operator, * = multiplication operator, / = division operator.

artificial neural network) for the modelling of the redfin barbel (Muñoz-Mas et al., 2017). Here, a derivative GA analysis is repeated a number of times to increase ensemble size. By checking convergence of the solutions determined with the GA for an increased ensemble size, one can determine an optimal set of solutions. With this, the authors showed the potential of using multilayer perceptron ensembles and EAs for the identification of multiple models reflecting simulation uncertainty (i.e. ensemble forecasting (Araújo and New, 2007)).

An interesting observation is that EA hyperparameters are reported more consistently in recent years. In addition, testing the training robustness as a practice has increased. The latter is probably due to the availability of growing computational resources. In addition, species prevalence is increasingly reported, suggesting that practioners are more aware of the effect of prevalence bias on model training. As such,

Table 3
Overview of often-used measures to define an objective function. For a in-depth review and formulation of regression coefficients, we refer to Mouton et al. (2010).

Measure	Classification (C) or regression (R)	Acronym symbol	Reference
Correctly classified instances	C	CCI	Mouton et al. (2010)
Cohen's Kappa	C	Kappa	Mouton et al. (2010)
Sensitivity	C	Sn	Mouton et al. (2010)
Sensitivity	C	Sp	Mouton et al. (2010)
True skill statistic	C	TSS	Mouton et al. (2010)
Area under the receiver operator characteristic curve	C	AUC	Mouton et al. (2010)
Correctly fuzzy classified instances	C	% CFCI	Van Broekhoven et al. (2006)
(Root) mean squared errors	R	(R)MSE	Based on species density, see Fukuda (2009) Based on suitability, see Fukuda et al. (2012)
Sum of squared errors	R	SSE	Based on population size: D'Angelo et al. (1995)
Linear correlation	R	ρ	Based on population size: D'Angelo et al. (1995)

it is observed that more detailed and robust approaches are presented. Finally, most studies rely on an extension of simple genetic algorithms, whereas genetic programming has not been employed in recent years.

4. Strengths, weaknesses, opportunities and threats analysis

To explore the potential of EAs in species distribution modelling, we performed a strength, weaknesses, opportunities and threats (SWOT) analysis. In this review, strengths and weaknesses refer to *current* characteristic of EAs that offer respectively advantages or disadvantages compared to other techniques. Opportunities and threats refer to *future* (dis-)advantages. To perform this analysis, the literature of EAs in species distribution modelling was scanned and the specific strengths and weaknesses of the use of EAs were compiled. In addition, opportunities and threats were assessed by testing compliance with known challenges in species distribution modelling (based on Araújo and Guisan, 2006; Austin, 2007; Araújo and New, 2007; Guisan and Zimmermann, 2000; Guisan and Rahbek, 2011). In the past, specific advantages of EAs and metaheuristics were mainly derived from experiments as the true functioning of the algorithms was poorly understood (Boussaïd et al., 2013; Maier et al., 2014). That is why the analysis in this section is based on specific examples rather than theoretical studies.

EAs are particularly useful in situations where solutions to complex problems have to be found for which little information is available to characterize the optimal solutions. In these situations, it is not possible to do a grid-search, *i.e.* check all candidate solutions one by one, as it would take an exponential amount of computational time. Cases characterised by a high degree of non-linearity to which little information is available to bound the search, such as incorporating interactions in SDMs (Kissling et al., 2012), can be classified as complex problems. The ability to deal with this complexity is an advantage over other methods. EAs also have notable weaknesses and pitfalls which are discussed in this section. It is important to note that other potentially suitable metaheuristic algorithms exist such as particle swarm optimization and ant colony optimization. They are also shortly discussed as they share a number of characteristics with EAs making them interesting for machine learning in species distribution modelling.

4.1. Problem encoding and flexibility

A clear strength is the flexibility of EAs offering the chance to implement any type of machine learning problem by using the encoding-model interface. Specifically, the ability of (1) encoding the model in a computation element, the chromosome, and (2) using mappers to translate chromosomes to models allows separating the process of training (with operators) from fitness calculation (model performance evaluation). This flexibility has already been illustrated in GARP, where different relations, *e.g.* logistic, linear or Boolean, can be used in the software (Olden et al., 2008).

The mentioned flexibility allows to define various ways of model training, *i.e.* estimating model parameters and/or reducing model complexity. Reducing model complexity in conjunction with learning can present an opportunity for the use of individual- and agent-based to support species distribution modelling. Indeed, the structure of these individual- and agent-based models can be complex (Grimm et al., 2010) and model simplification with flexible machine learning algorithms could allow for a further automation of model development. To reduce model complexity, the most relevant features of a model are selected by encoding embedded or wrapper feature selection (Saeys et al., 2007). Wrapper feature selection is concerned with selecting features for other data-driven or an already parameterized model. As opposed to this, embedded feature selection estimates model parameters and selects features simultaneously. For wrapper feature selection, a binary string encoding the inclusion (1) or exclusion (0) can be used (D'heygere et al., 2006) whereas, for embedded feature selection, a 'list of list' approach can be used (Gobeyn, 2018). For encoding a binary or continuous string, the reader is referred to Haupt and Haupt (2004). The list of list, a first order list is implemented in the genotype to represent a feature in- or exclusion. If an inclusion for a feature is considered then a second-order list is defined, holding the value for the model parameters coupled to the input feature, *i.e.* coefficients of the response curve. The list of list approach seems to be promising, however, additional research is required to verify its performance. For parameter estimation, a string of continuous values (for example, values of model parameters describing the species' niche, see (Fukuda et al., 2011)) can be implemented in the genotype of the chromosomes. These are then translated to model parameters and – after model execution – a fitness value.

A disadvantage of the chromosome encoding and the use of a mapper function is that a certain amount of programming skills is required. This might hinder novel users to use EAs or other metaheuristics for their machine learning application. However, since machine learning with EAs is specifically applicable to increase transparency of complex models, it is expected that the initial investment in programming will be the better option – especially in the long run. Even more, open science is challenging ecological informaticians to increase code flexibility, modularity and transparency (Golding et al., 2017) leading to a more user-friendly experience in programming languages such as R and Python. In addition, a number of initiatives are taken in the field of computer science to help non-expert and expert users to deal with feature selection and hyperparameter optimization. For example, Auto-WEKA (Feurer et al., 2015) and Auto-skLearn (Kotthoff et al., 2016) are initiatives that consider the problem of simultaneously selecting a learning algorithm and values for the hyperparameters through Bayesian learning. As such, these tools offer the opportunity to investigate the position of EAs in comparison to other machine learning methods for specific problems tackled in species distribution modelling and ecology in general.

4.2. Population-based approach

The population-based approach of EAs is considered the second advantage for species distribution modelling since ecological phenomena characterised by a large amount of noise are too complex to describe by one model (Fukuda et al., 2013; Merow et al., 2014; Muñoz-Mas et al., 2017; Vezza et al., 2015). Using multiple models in the context of ensemble learning is useful to reflect model uncertainties (Araújo and New, 2007). Practically, the EA would be run a couple of times preferably with other samples of the training data (*i.e.* cross-validation or bootstrapping) and track the best models found in each run. Ensemble learning has shown to be valuable to avoid SDM overfitting, especially for modelling rare species (Breiner et al., 2015). The population-based approach of EAs allows providing an informative ensemble of near-optimal solutions rather than just one optimal solution. In this perspective, EAs can be used to generate ensembles comparable (Muñoz-Mas et al., 2017) and possibly serve as an alternative for the RF method.

The combination of iterating a number of solutions and the crossover and mutation operators offers the opportunity to explore multiple areas of the search space (Holland, 2000). Applied to feature selection, it allows tracking interesting combinations of features over several generations. This is considered a great strength and a competitive alternative to stepwise selection procedures usually used in species distribution modelling. In stepwise selection procedures, an alternative model is tested to data by iteratively excluding (including) a feature (Zuur et al., 2009). These approaches are considered greedy because they make locally optimal decisions with the assumption that a (near-) optimal solution will be found in the vicinity of this local solution. Although the forward selection approach is computationally efficient, this procedure may ignore informative combinations of features which are individually only marginally relevant. The search behaviour of EAs is totally different: They combine and test solutions that are located in various regions of the search space.

EAs and ant colony optimization are population-based approaches able to deal with combinatorial optimization problems (Boussaïd et al., 2013) whereas particle swarm optimization was initially designed to solve continuous problems (Kennedy and Eberhart, 1997). Combinatorial optimization problems are a class of discrete optimization problems in which the input arguments encode permutations, combinations or variations (Scheerlinck et al., 2009). The way the candidate solutions are generated is the main difference between EAs and ant colony optimization. In EAs, candidate solutions are encoded as strings of bits or real numbers of the chromosomes whereas for ant colony optimization the potential solutions are encoded in the environment of ants. That is, the ants or agents propagate through the search space and new candidate solutions are constructed from the information in this environment. This way, the memory of the system is embedded in the environment rather than the objects. This property makes ant colony optimization more appealing for modelling dynamically changing systems (Maier et al., 2003; Szemis et al., 2012; Zecchin et al., 2006).

For now, the application of ant colony optimization in species distribution modelling might seem less interesting since data are often not available over multiple time instances. As depicted in the introduction, near-real-time data are expected to arrive as technologies in species-tracking and remote sensing are continuously improved (Cord et al., 2014; Pauwels et al., 2014; Bastille-Rousseau et al., 2017). As ant colony optimization is able to deal with dynamic constraints without reinitialisation, it is expected to be appropriate to deal with these type of dynamic data. In these cases, the use of ant colony optimization for model identification could be assessed as superior to EAs.

A fairly novel class of population-based methods are ‘Estimation of distribution algorithms’ (EDAs). These algorithms guide the search for an optimal solution by sampling probabilistic models of candidate solutions, and by using selection operators also applied in EAs. The aim of EDAs is not only to optimize models, but also to provide a series of probabilistic models revealing characteristics of the problem being solved (Pelikan et al., 2015). Other examples of newly developed population-based methods to obtain this type of information are ‘irace’ (López-Ibá nez et al., 2016) and sequential model-based optimization (Hutter et al., 2011). They all share the aim of automatic algorithm configuration, defined as finding good algorithm settings (values for hyperparameters, operators) for solving unseen problem instances by learning on a set of training problem instances (López-Ibá nez et al., 2016). Applying this type of algorithms to train SDMs could be interesting to further learn about the characteristics of the training problem at hand. Sample prevalence is a typical example of a characteristic of a training data set (see also Table 2). The mentioned techniques could thus be used to train models on data sets with varying sample prevalence so to provide interesting insights on the effect of sample prevalence on – not only the objective measure – but also algorithms' functioning.

4.3. Hyperparameters

The standard application of an EA requires five hyperparameters to be optimized or tuned (population size, a stopping criterion, selection rate and crossover and mutation rate). This can be considered as a disadvantage since the performance of the EA depends on the choice of these hyperparameters (Grefenstette, 1986; Feurer et al., 2015). Guidelines for (automated) tuning these hyperparameters are found in the literature (Gibbs et al., 2008, 2010; López-Ibá nez et al., 2016). Yet, it is important to note that the ‘No Free Lunch Theorem’ states that there is no global set of hyperparameters effective for every problem (Wolpert and Macready, 1997). Consequently, every class of problems will require hyperparameter testing. The results our literature review show that a limited number of studies (8 out of 25) used an iterative approach to obtain hyperparameter values. In addition, no significant relation between hyperparameters could be identified (see supportive information 3). This is in line with the findings of Gibbs et al. (2008) who empirically determined the degree of interaction between hyperparameters for a list of optimization problems. Only the population size shows a strong inverse relationship with the mutation rate whereas the interaction between other hyperparameters was found not to be as relevant for the GA performance. Within our analysis, we could not determine a relation between the number of chromosomes and the mutation rate. The reason for this observation is that hyperparameters are rarely optimized in the field of species distribution modelling. Algorithms are used, and settings seem to be copied from other publications without explicit reasoning (see Table 2: Boets et al. (2013), D'heygere et al. (2003, 2006 and Zarkami et al. (2012)). As such, we suspect that hyperparameters values in the studies in Table 2 are suboptimal. Here, we advocate the practice of testing and reporting the values for hyperparameters and their effect on the objective function, so readers can assess which hyperparameters might be useful for a specific application in species distribution modelling. We promote the use of guidelines to have a good estimate of optimal values for the hyperparameters as those in Gibbs et al. (2008) and Gibbs et al. (2010). Although the number of hyperparameters to be determined may be a weakness of EAs, many metaheuristic algorithms (*i.e.* ant colony optimization, particle swarm optimization, simulated annealing) and machine learning algorithms share this shortcoming. As noted at the end of section 4.1, a number of tools developed in computer science are being developed to automate the hyperparameter optimization problem (Feurer et al., 2015).

4.4. Multiobjective machine learning

An opportunity of EAs in species distribution modelling is their potential use as multiobjective machine learning methods which aim to train a model based on multiple – potentially conflicting – objectives. Typically, the purpose of species distribution modelling is to train models which estimate species presence and absence well. In many cases, it is desired – for instance in decision management – to give a higher weight to one or the other (Mouton et al., 2009). A number of evaluation criteria based on the classification of the occurrence probability (*e.g.* Cohen's Kappa or True Skill Statistics) are being used to pool the degree of correct estimation of species presence and absence (Mouton et al., 2010). Unfortunately, these evaluation measures depend on sample prevalence. Consequently, training models with these data having varying sample prevalence are biased. A pragmatic approach to solve this issue is to keep sample prevalence equal over all data samples and/or to define a trade-off between commission and omission errors in the objective function (Allouche et al., 2006; Manel et al., 2001; Mouton et al., 2010).

The trade-off between omission and commission errors can be viewed as a multiobjective problem. EAs have proven to be adequate techniques to identify trade-offs between objectives (Penn et al., 2013; Sweetapple et al., 2014). In general, EAs can be used to determine the entire set of Pareto optimal solutions or at least a representative subset. A Pareto optimal set is a set of solutions that are nondominated when compared with other solutions of the solution space (Deb et al., 2000). For example, for species distribution modelling, a Pareto optimal set could be a set of equally valid solutions to a problem presenting the trade-off between commission and omission errors. This way, decision makers obtain a set of solutions that can be very valuable for different aspects of ecosystem decision management (Guisan et al., 2013). A well-known example of a multiobjective optimizer using an EA is the non-dominated sorting GA II of Deb et al. (2000). In this algorithm, a simple GA with uniform crossover and mutation but with specific selection operators is used. For the selection operator, different nondominant fronts are identified. These nondominant fronts are estimates of the Pareto front defined by two or more objectives. The chromosomes in each non-dominant front have the same assigned dummy fitness value, ranked according to the 'strength' of the front. These dummy fitness values are used to select chromosomes (Deb et al., 2000). This process is repeated until a nondominant front equal or close to the Pareto optimal front is found. An example of the use of the non-dominated sorting GA II in ecology is presented by Côté et al. (2007).

5. Recommendations for application

EAs and other metaheuristic algorithms are particularly useful to solve problems such as feature selection, parametrisation of complex models, and optimization of other learning algorithms. These algorithms are likely not suited to solve every problem as the development of a specific EA will require high investment costs - in terms of programming and algorithm understanding - returning little improvement in model insight and predictive performance. In these cases, the use of machine learning methods, such as decision trees, GLMs or Maxent would be more appropriate. However, we recommend to consider EAs when the problem at hand has one of the following characteristics:

- The problem and search for a solution is expected to be complex (*e.g.* includes species interactions or many features) and little information is available to *a priori* reduce complexity (see, for example, Kissling et al. (2012)).
- Many (complex) boundaries *can* be formulated for the problem. These could, for instance, be obtained from experts or ecological databases (Verberk et al., 2012).
- Solutions to the problem are required to be transparent and flexible for model (re-)analysis, for instance for decision management (Adriaenssens et al., 2004).
- The input data set has a high number of features, and manual feature reduction is no longer possible (*e.g.* pesticide database of river sediment in Flanders counts more than 200 identified pesticides (VMM, 2018)).
- The model knows many parameters which have to be calibrated (Van Broekhoven et al., 2007).
- A trade-off between objective functions is required for decision management applications. This can, for instance, be the trade-off between model complexity and performance, or between the correct estimation of species presence and absence.

For a specific problem, one can select from a number of EA implementations. In Table 4, a suggestion for the type of EA are provided for a number of problems. Two 'trivial' problems are listed, parameter estimation and feature selection (see row one and two), whereas other applications are less obvious, and often problem-specific implementations. For the calibration of parameter-rich models (> 10 parameters), a binary or real GA encoded can be used, since both are expected to perform equally well (Van Broekhoven et al., 2007). The second case involves the reduction of the number of input features with the help of EAs. Typically, this applies to data sets for which a large number of potential input variables can explain species occurrence. This type of learning could be particularly interesting when remote sensing products are used, in order to reduce the amount of input data required to estimate species distributions from spatial input data (Hampton et al., 2013). Automated variable selection with EAs can be helpful to steer model development, but as noted by Araújo and Guisan (2006), this should not replace a selection based on expert knowledge. In the case of feature selection, a binary encoded GA is implemented, encoding the in- or exclusion of input features (D'heygere et al., 2006). This feature selection can be helpful for the optimization of stacked SDMs or population-based SDMs. In these SDMs, models for different species are coupled with each other (Guisan and Rahbek, 2011), and are allowed to interact. With the number of species considered in these stacked SDMs, the number of model elements increases exponentially (due to one-on-one interaction). Binary GAs can be used to simplify these models, preventing an overly complex model to be fitted to a limited number of species occurrence observations. In addition, binary GAs can be used to optimize artificial neural networks. In this case, different layers or neurons can be implemented in the binary string, and the structure of the ANN can be optimized (see Muñoz-Mas et al. (2017)).

For simple binary and real-coded GAs, a selection rate of 0.5, a crossover rate above 0.8, mutation rate lower than 0.2 and 100 generations will in general work well, when the number of chromosomes is between 30 and 200, independent of the chromosome length (Gibbs et al. (2008) and Table 2). For the choice of the selection operator, we advise using tournament or roulette wheel selection. Both are simple to understand, and generally give satisfying results when compared to other selection operators (Goldberg and Deb, 1991). The use of elitism is advised, however, it is important to note that the use of elitism can decrease the population diversity, and facilitate faster convergence. The need for fast convergence, motivated by limited available computation resources can be an important boundary condition in choosing the number of model evaluations. This number is determined by the number of chromosomes multiplied by a number of generations. A limited number of model evaluations, 400 and 2500, have been used, and have presumably led to satisfying results. Increasing the number of evaluations can be useful, however, it is possible that gains in accuracy or precision are marginal. As discussed by Gibbs et al. (2008), the number of evaluations should vary as a function of the available computational resources. As a rule of thumb, we advise to focus on a cross-validation resampling strategy and on repeated execution of the EA/cross-validation strategy to increase robustness, rather than employing a larger number of model evaluations.

Users are advised to consider stratification according to sample prevalence to a design cross validation strategy. As discussed above, accuracy measures can vary as a function of this prevalence. To make

Table 4
Suggested EA or metaheuristic algorithm useful in species distribution modelling. Algorithms followed by a '*' are suggested in this review, but have yet to be tested, or have only tested in a number of experiments within ecology.

Learning problem	Suggested algorithm	Example reference
Feature selection	Binary SGA	D'heygere et al. (2006)
	Ant colony optimization*	–
Parameter estimation	Real-coded SGA	Van Broekhoven et al. (2007)
	Particle swarm optimization*	–
	Simulated annealing*	–
Parameter estimation and feature selection	GP and GAs	Jeong et al. (2011)
	Variable length GAs*	Gobeyn and Goethals (2017)
Hyperparameter optimization of other machine learning algorithms	Evolutionary optimization (problem-specific)	Muñoz-Mas et al. (2016)
Multi-objective optimization	NSGA-II*	–
Problem characteristics identification	EDA*	–

results comparable, it is of importance that data are stratified according to this prevalence. The choice for a number of folds, and repetitions will depend on the available computation resources, and the size of the data set. Precision will increase with a higher number of repetitions and folds, leading to a longer runtime. When data sets are small, and models are learned fast, a higher number is thus preferred. In contrast, when learning is slow, one can opt to choose fewer folds and repetitions (Kohavi, 1995). As discussed above, one can also consider lowering the number of function evaluations.

When further fine-tuning of the hyperparameters is desired, we recommend using the guidelines of Gibbs et al. (2008), as every specific problem can have a unique set of optimal hyperparameters. The choice of the objective function, which the GA has to optimize, depends on the study objectives: does one aim to estimate species presence well, or rather absence? If the former is true than a higher weight should be given to sensitivity, in the case of the latter, specificity (see Table 3). In case a trade-off between both should be identified, one can consider a multi-objective EA. In these algorithms, a specific selection operator is implemented in the GA to weight different objectives (see for example the non-dominated sort in NSGA-II (Deb et al., 2002)).

Three main points need to be taken into account when machine learning or other algorithms are considered to solve a hypothesis. First, specifying the model and its structural component, the subject of model training and the objective of the model and the study (see Box 3) is important (Guisan and Zimmermann, 2000). For example, is the aim of the model to understand a specific theoretical assumption about species interaction? Or is the aim to develop a predictive model for estimating species occurrence in an ecosystem with many interactions? In a second step, an algorithm to train the model(s) needs to be selected (Box 3, second part). Specifically, algorithm operators, problem encoding and operators need to be defined. Here, it is important to make a distinction between algorithms which make use of explicit encoding (EAs, ant colony optimization) and those which do not (decision trees, GLMs). It is empirically found that algorithms making use of encoding work well to train models with hypotheses embedded in the model structure (Maier et al., 2014). The initial choice for a type of algorithm and use of encoding will hence determine the choice for hyperparameters and operators.

Finally, a platform to implement the approach for the machine learning application is required. GUI packages can be used, however, adopting these packages can considerably limit the options which make EAs interesting in the first place. For that reason, we advise to use a high-level scripting language such as Python or R and search for existing codes implemented in these languages. An additional advantage of using high-level scripting languages for machine learning applications is their transferability to high performance and cloud computing infrastructure. Preferably, the scripting is done in an open science context allowing for continuous code improvement and validation through modular scripting. For a good introduction on modular scripting applied to ecology, we refer to Golding et al. (2017). Typically, open science is performed on code sharing platforms such as GitHub (https://github.com/).

6. Future perspectives and conclusions

Recent advances in theoretical ecology (Leibold et al., 2004) and conceptual modelling (Guisan and Rahbek, 2011) are challenging scientists to continuously develop new ways to deal with this increasing complexity. The field of machine learning has proven to be useful to tackle these questions, despite that researchers are struggling to identify the appropriate approach to address increasing complexity (Kissling et al., 2012).

Maxent is currently the most used technique to model species distributions when considering terrestrial cases. For freshwater system case studies, innovative methods such as artificial neural networks and EAs are increasingly being used to solve less straightforward problems. Model developers will be required to deal with this increased complexity, preferably in an open science context. This depicts full transparency in the methodology, but also the practical encoding (Golding et al., 2017; Phillips et al., 2017). In addition, it requires the developed algorithms to be easily transferable and adaptable to new problems. Considering these aspects, EAs and other metaheuristic algorithms are found to be of particular use since they split the training process from the objective function evaluation (model run). In addition, EAs allow dealing with complex cases (Eiben and Smith, 2015) making them appropriate candidates to train the next generation of species distribution models. Dealing with hyperparameters optimization and the requirement of programming and modelling skills are considered disadvantageous, hindering the use of EAs and other metaheuristic algorithms. For the first, hyperparameter optimization methods are already available giving satisfying results for multiple problems (Gibbs et al., 2008; Gobeyn et al., 2017). The second, the need for modelling know-how will require standardization, documentation and refinement of the algorithm development and application process (Jakeman et al., 2006; Grimm et al., 2010) going hand in hand with the philosophy of open science. In this review paper, a number of suggestions with respect to the definition of the model, algorithm and implementation are given. With this, we aim to stimulate ecologists to use and further refine the development of EAs applied to species distribution modelling, hypothesis testing and preferably ecology in general.

As technological advances in machine learning are reshaping the way scientist develop models and analyse data, researches are increasingly aware that one specific algorithm won't offer a tailor-made solution to every problem (Chatfield, 1993). With this synthesis, an

Box 3
General guidelines for applying a metaheuristic machine learning algorithm

Model:

1 **Model formulation**: First the model scale, resolution, model inputs, states, parameters and boundary conditions relevant to the species and case study need to be considered. At this stage, it is inspected if the model can be simplified by making specific assumptions (*e.g.* only consider specific species interactions) and/or identifying correlating features. For the latter, this can be done a prior model fitting with filtering methods based on input data (using, for instance, the Spearman rank correlation, see Saeys et al. (2007) and Dormann et al. (2012)) or during model fitting (by computing, for instance, a mutual information criterion, see May et al. (2008)). It is important to note that the use of automated procedures to select features should not serve as a replacement for an expert-based selection (Araújo and Guisan, 2006).

2 **Objective**: Define specific objectives and criteria to which the model should comply. For instance, is the aim of to obtain models which estimate primarily species presence well or rather species absence? Other aspects involved can be related to model complexity (for example see (Phillips et al., 2006)). Many options are available to define a measure. It is important to note that these are sensitive to the sample prevalence (Mouton et al., 2009, 2010). In other words: the measure used for model training can be varied as a function of the sample prevalence. This can cause a bias in the obtained model.

3 **Subject of training**: Defining which specific elements of the models are perturbed to maximize or minimize an objective function. It can be aimed to estimate model parameters or/and reduce the number of input variables (and thus model elements) (Fukuda et al., 2011). When the goal is to decrease the number of model structural elements, both wrapper and embedded feature selection methods can be used. feature selection selects input features which are most relevant – given an objective – to explain patterns in the data. In embedded feature selection, parameters are estimated while performing feature selection. On the contrary, in wrapper feature selection, parameters are estimated for each feature subset (or prior feature selection) (Saeys et al., 2007). For the latter, another machine learning algorithm is typically run within the feature selection procedure. **Algorithm:**

(a) **Type of algorithm**: Users are advised to consider EAs and other metaheuristics in order to train models to test complex ecological hypotheses. A very good example using a simulated annealing metaheuristic algorithm to inspect the effect of species interactions and stress tolerance on biodiversity is presented by Baert et al. (2016). For relatively simple questions mainly aiming to get a first insight into the problem, we advise using Maxent, GLMs and/or decision trees.

(b) **Encoding**: A binary (string of zeros and ones) encoding can be considered for wrapper feature selection. Applied to EAs, real-valued encoding (string of continuous values) can be considered for parameter estimation and a list of list encoding for embedded feature selection. Haupt and Haupt (2004) provide background hints and tips for the implementation of a binary and real-values encoding in evolutionary algorithms. To implement embedded feature selection in EAs, a list of list approach can be used (Srikanth et al., 1995; Gobeyn and Goethals, 2017). In addition, boundary conditions need to be addressed in the encoding and functioning of the operators (*e.g.* implementing repair operators for genome in EAs).

(c) **Operators**: Metaheuristic algorithms use a number of operators depicting efficiency of the algorithm. Many implementations are available and depend on the encoding of the solutions. For EAs, typically tournament selection, uniform crossover and uniform mutation operators are implemented (Haupt and Haupt, 2004). The implementation of multiobjective machine learning with EAs requires specific selection operators (Deb et al., 2002).

(d) **Hyperparameters**: All machine learning methods have a number of hyperparameters which need to be set. For standard EAs, selection, crossover and mutation rate needs to be set together with the number of iterations. Guidelines are available for many algorithms (*e.g.* EA, Gibbs et al. (2008) or decision trees, Everaert et al. (2016)). It is important to note that setting hyperparameters is for many machine learning algorithms required to acquire satisfying results with the lowest computational effort (Gibbs et al., 2015). **Implementation:**

i **Programming language**: A number of programming languages exist to implement algorithms. For data science and machine learning, high-level programming languages such as Python and R are the most popular ones. These languages offer an interface for intuitive high-level programming. In addition, support can easily be found at online platforms such as Stack Overflow (www.stackoverflow.com) to solve specific programming problems. An alternative is to use algorithms available under a GUI environment,*e.g.* WEKA which also facilitates a command-line interface and Java API. These are helpful for machine learning, however, the use of these techniques to train hypothetical-driven SDMs can be tedious. In addition, these GUI applications are often difficult to use for repeated analysis (*i.e.* for uncertainty analysis) on high-performance computing infrastructure.

ii **Open science**: Developing a tailor-made package for a specific application can be a time-consuming practice. Therefore we recommend the development of an application based on existing Python or R packages. General or specific packages can be downloaded from the language developers websites and Github (https://github.com/). The latter is an open source code hosting platform for version control and (scientific) collaboration. Programmers in environmental and ecological science are increasingly aware of the importance of open source (science), code collaboration, reproducibility (not only in results but also in code) and modular scripting. A good example of this philosophy applied to species distribution modelling is published by Golding et al. (2017). In this approach, the authors provide a modular framework operating on snippets of R code that are interchangeable among each other. Finally, benchmarking algorithms and codes can be done by using datasets from GBIF, a free and open access to biodiversity data. In this case, an ecological data set is used, and different algorithms are applied to entangle the specific strengths and weaknesses of the used algorithms. As an alternative, an open, organized, online ecosystem for machine learning such as OpenML (https://www.openml.org/home) and Kaggle (https://www.kaggle.com) can be used.

insight is presented on how to use EAs as a technique to solve specific problems in ecology rather than using it as a ready-to-use technology to map species distributions.

Author contribution

S.G., A.M.M., A.F.C., M.V. and P.L.M.G. designed the research. S.G. conducted the literature review. S.G. wrote the manuscript, and A.M.M., A.F.C., A.K., M.V. and P.L.M.G. provided edits to the manuscript.

Acknowledgments

The authors would like to thank two anonymous reviewers for their thorough review and valuable comments which have considerably improved the manuscript. This study was carried out within the CROSSLINK project, funded through BiodivERsA, under the Horizon 2020 ERA-NET COFUND scheme. This research was partly funded through the 2013/2014 BiodivERsA/FACCE-JPI joint call, with the national funder BMBF – German Federal Ministry of Education and Research (Project “TALE - Towards multifunctional agricultural landscapes in Europe: Assessing and governing synergies between food production, biodiversity, and ecosystem services”, grant 01 LC 1404 A).

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.ecolmodel.2018.11.013.

References

Adriaenssens, V., 2004. Knowledge-based macroinvertebrate habitat suitability models for use in ecological river management, Ph.D. Thesis. Ghent University.

Adriaenssens, V., De Baets, B., Goethals, P.L.M., De Pauw, N., 2004. Fuzzy rule-based models for decision support in ecosystem management. Sci. Total Environ. 319, 1–12.

Allouche, O., Tsoar, A., Kadmon, R., 2006. Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic (TSS). J. Appl. Ecol. 43, 1223–1232.

Araújo, M.B., Guisan, A., 2006. Five (or so) challenges for species distribution modelling. J. Biogeogr. 33, 1677–1688.

Araújo, M.B., New, M., 2007. Ensemble forecasting of species distributions. Trends Ecol. Evol. 22, 42–47.

Austin, M.P., 2007. Species distribution models and ecological theory: a critical assessment and some possible new approaches. Ecol. Model. 200, 1–19.

Baert, J.M., Janssen, C.R., Sabbe, K., De Laender, F., 2016. Per capita interactions and stress tolerance drive stress-induced changes in biodiversity effects on ecosystem functions. Nat. Commun. 7, 1–8.

Bastille-Rousseau, G., Murray, D.L., Schaefer, J.A., Lewis, M.A., Mahoney, S., Potts, J.R., 2017. Spatial scales of habitat selection decisions: implications for telemetry-based movement modelling. Ecography 40, 1–7.

Bennetsen, E., Gobeyn, S., Goethals, P.L.M., 2016. Species distribution models grounded in ecological theory for decision support in river management. Ecol. Model. 325, 1–12.

Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. J. Mach. Learn. Res. 13, 281–305.

Boets, P., Holguin, G., Lock, K., Goethals, P.L.M., 2013. Data-driven habitat analysis of the Ponto-Caspian amphipod Dikerogammarus villosus in two invaded regions in Europe. Ecol. Inform. 17, 36–45.

Booth, T.H., 1985. A new method to assist species selection. Commonwealth Forest. Rev. 64, 241–250.

Booth, T.H., Nix, H.A., Busby, J.R., Hutchinson, M.F., 2014. Bioclim: The first species distribution modelling package, its early applications and relevance to most current MaxEnt studies. Divers. Distrib. 20, 1–9.

Boussaïd, I., Lepagnot, J., Siarry, P., 2013. A survey on optimization metaheuristics. Inform. Sci. 237, 82–117.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees, 1st ed. Taylor & Francis.

Breiner, F.T., Guisan, A., Bergamini, A., Nobis, M.P., 2015. Overcoming limitations of modelling rare species by using ensembles of small models. Methods Ecol. Evol. 6, 1210–1218.

Carpenter, G., Gillison, A.N., Winter, J., 1993. DOMAIN: a flexible modelling procedure for mapping potential distributions of plants and animals. Biodivers. Conserv. 2, 667–680.

Chatfield, C., 1993. Neural networks: forecasting breakthrough or passing fad? Int. J. Forecast. 9, 1–3.

Chen, W.C., Chang, N.B., Chen, J.C., 2003. Rough set-based hybrid fuzzy-neural controller design for industrial wastewater treatment. Water Res. 37, 95–107.

Cord, A.F., Klein, D., Gernandt, D.S., de la Rosa, J.A.P., Dech, S., 2014. Remote sensing data can improve predictions of species richness by stacked species distribution models: a case study for Mexican pines. J. Biogeogr. 41, 736–748.

Côté, P., Parrott, L., Sabourin, R., 2007. Multi-objective optimization of an ecological assembly model. Ecol. Inform. 2, 23–31.

Cutler, D., Edwards, T., Beard, K.H., Cutler, A., Hess, K., Gibson, J., 2007. Random Forests for Classification in Ecology. Ecology 88, 2783–2792.

D’Angelo, D.J., Howard, L.M., Meyer, J.L., Gregory, S.V., Ashkenas, L.R., 1995. Ecological uses for genetic algorithms: predicting fish distributions in complex physical habitats. Can. J. Fish. Aquat. Sci. 52, 1893–1908.

DeAngelis, D.L., Yurek, S., 2017. Spatially explicit modeling in ecology. A review. Ecosystems 20, 284–300.

Deb, K., Agrawal, S., Pratap, A., Meyarivan, T., 2000. A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 849–858.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6, 182–197.

D’heygere, T., Goethals, P.L.M., De Pauw, N., 2003. Use of genetic algorithms to select input variables in decision tree models for the prediction of benthic macroinvertebrates. Ecol. Model. 160, 291–300.

D’heygere, T., Goethals, P.L.M., De Pauw, N., 2006. Genetic algorithms for optimisation of predictive ecosystems models based on decision trees and neural networks. Ecol. Model. 195, 20–29.

Ding, S., Li, H., Su, C., Yu, J., Jin, F., 2013. Evolutionary artificial neural networks: a review. Artif. Intell. Rev. 39, 251–260.

Dormann, C.F., Schymanski, S.J., Cabral, J., Chuine, I., Graham, C., Hartig, F., Kearney, M., Morin, X., Römermann, C., Schröder, B., Singer, A., 2012. Correlation and process in species distribution models: bridging a dichotomy. J. Biogeogr. 39, 2119–2131.

Dunning, J.B., Stewart, D.J., Danielson, B.J., Noon, B.R., Root, T.L., Lamberson, R.H., Stevens, E.E., Danielson, B.J., Noon, B.R., Root, T.L., Lamberson, R.H., Stevens, E.E., 1995. Spatially explicit population models: current forms and future uses. Ecol. Appl. 5, 3–11.

Eiben, A.E., Smith, J., 2015. From evolutionary computation to the evolution of things. Nature 521, 476–482.

Elith, J., Graham, C., Anderson, R., Dudik, M., Ferrier, S., Guisan, A., Hijmans, R., Huettmann, F., Leathwick, J., Lehmann, A., Li, J., Lohmann, L., Loiselle, B., Manion, G., Moritz, C., Nakamura, M., Nakazawa, Y., Overton, J., Peterson, A., Phillips, S., Richardson, K., Scachetti-Pereira, R., Schapire, R., Soberon, J., Williams, S., Wisz, M., Zimmermann, N., 2006. Novel methods improve prediction of species’ distributions from occurrence data. Ecography 29, 129–151.

Elith, J., Leathwick, J.R., 2009. Species distribution models: ecological explanation and prediction across space and time. Annu. Rev. Ecol. Evol. Syst. 40, 677–697.

Elith, J., Phillips, S.J., Hastie, T., Dudík, M., Chee, Y.E., Yates, C.J., 2011. A statistical explanation of MaxEnt for ecologists. Divers. Distrib. 17, 43–57.

Everaert, G., Pauwels, I., Bennetsen, E., Goethals, P.L.M., 2016. Development and selection of decision trees for water management: impact of data preprocessing, algorithms and settings. AI Commun. 29, 711–723.

Favaro, L., Tirelli, T., Pessani, D., 2011. Modelling habitat requirements of white-clawed crayfish (*Austropotamobius pallipes*) using support vector machines. Knowl. Manage. Aquat. Ecosyst. 401, 21.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., Hutter, F., 2015. Efficient and Robust Automated Machine Learning. Adv. Neural Inform. Process. Syst. 28, 2944–2952.

Fukuda, S., 2009. Consideration of fuzziness: is it necessary in modelling fish habitat preference of Japanese medaka (*Oryzias latipes*)? Ecol. Model. 220, 2877–2884.

Fukuda, S., De Baets, B., Mouton, A.M., Waegeman, W., Nakajima, J., Mukai, T., Hiramatsu, K., Onikura, N., 2011. Effect of model formulation on the optimization of a genetic Takagi-Sugeno fuzzy system for fish habitat suitability evaluation. Ecol. Model. 222, 1401–1413.

Fukuda, S., De Baets, B., Waegeman, W., Verwaeren, J., Mouton, A.M., 2013. Habitat prediction and knowledge extraction for spawning European grayling (*Thymallus thymallus* L.) using a broad range of species distribution models. Environ. Model. Softw. 47, 1–6.

Fukuda, S., Hiramatsu, K., 2008. Prediction ability and sensitivity of artificial intelligence-based habitat preference models for predicting spatial distribution of Japanese medaka (*Oryzias latipes*). Ecol. Model. 215, 301–313.

Fukuda, S., Mouton, A.M., De Baets, B., 2012. Abundance versus presence/absence data for modelling fish habitat preference with a genetic Takagi-Sugeno fuzzy system. Environ. Monit. Assess. 184, 6159–6171.

Gendreau, M., Potvin, J.Y., 2010. Handbook of Metaheuristics, 2nd ed. Springer International Publishing, New York.

Gibbs, M.S., Dandy, G.C., Maier, H.R., 2008. A genetic algorithm calibration method based on convergence due to genetic drift. Inform. Sci. 178, 2857–2869.

Gibbs, M.S., Maier, H.R., Dandy, G.C., 2010. Comparison of genetic algorithm parameter setting methods for chlorine injection optimization. J. Water Resour. Plan. Manage. 136, 288–291.

Gibbs, M.S., Maier, H.R., Dandy, G.C., 2015. Using characteristics of the optimisation problem to determine the genetic algorithm population size when the number of evaluations is limited. Environ. Model. Softw. 69, 226–239.

Gobeyn, S., 2018. Species Distribution Model Identification Tool. URL: https://github.com/Sachagobeyn/SDMIT.

Gobeyn, S., Goethals, P.L., 2017. A variable length chromosome genetic algorithm approach to identify species distribution models useful for freshwater ecosystem management. In: Denzer, R., Schimak, G., H?ebíček, J. (Eds.), Environmental Software

Systems. Infrastructures, Services and Applications. Springer International Publishing, Cham, pp. 196–208.

Gobeyn, S., Volk, M., Dominguez-Granda, L., Goethals, P.L.M., 2017. Input variable selection with a simple genetic algorithm for conceptual species distribution models: a case study of river pollution in Ecuador. Environ. Model. Softw. 92, 269–316.

Goethals, P.L.M., Dedecker, A.P., Gabriels, W., Lek, S., De Pauw, N., 2007. Applications of artificial neural networks predicting macroinvertebrates in freshwaters. Aquat. Ecol. 41, 491–508.

Goldberg, D.E., Deb, K., 1991. A comparative analysis of selection schemes used in genetic algorithms. Found. Genet. Algorithms 1, 69–93.

Golding, N., August, T.A., Lucas, T.C., Gavaghan, D.J., van Loon, E.E., Mcinerny, G., 2017. The zoon R package for reproducible and shareable species distribution modelling. Methods Ecol. Evol. 9, 1–9.

Grefenstette, J.J., 1986. Optimization of control parameters for genetic algorithms. IEEE Trans. Syst. Man Cybern. 16, 122–128.

Grimm, V., Berger, U., DeAngelis, D.L., Polhill, J.G., Giske, J., Railsback, S.F., 2010. The ODD protocol: a review and first update. Ecol. Model. 221, 2760–2768.

Guisan, A., Rahbek, C., 2011. SESAM a new framework integrating macroecological and species distribution models for predicting spatio-temporal patterns of species assemblages. J. Biogeogr. 38, 1433–1444.

Guisan, A., Thuiller, W., 2005. Predicting species distribution: offering more than simple habitat models. Ecol. Lett. 8, 993–1009.

Guisan, A., Tingley, R., Baumgartner, J.B., Naujokaitis-Lewis, I., Sutcliffe, P.R., Tulloch, A.I.T., Regan, T.J., Brotons, L., McDonald-Madden, E., Mantyka-Pringle, C., Martin, T.G., Rhodes, J.R., Maggini, R., Setterfield, S.A., Elith, J., Schwartz, M.W., Wintle, B.A., Broennimann, O., Austin, M.P., Ferrier, S., Kearney, M.R., Possingham, H.P., Buckley, Y.M., 2013. Predicting species distributions for conservation decisions. Ecol. Lett. 16, 1424–1435.

Guisan, A., Zimmermann, N.E., 2000. Predictive habitat distribution models in ecology. Ecol. Model. 135, 147–186.

Hampton, S.E., Strasser, C.A., Tewksbury, J.J., Gram, W.K., Budden, A.E., Batcheller, A.L., Duke, C.S., Porter, J.H., 2013. Big data and the future of ecology. Front. Ecol. Environ. 11, 156–162.

Haupt, R.L., Haupt, S.E., 2004. Algorithms Practical Genetic Algorithms, 2nd ed. John Wiley & Sons, Inc., Hoboken.

He, K.S., Bradley, B.A., Cord, A.F., Rocchini, D., Tuanmu, M.N., Schmidtlein, S., Turner, W., Wegmann, M., Pettorelli, N., 2015. Will remote sensing shape the next generation of species distribution models? Remote Sens. Ecol. Conserv. 1, 4–18.

Hirzel, A.H., Le Lay, G., 2008. Habitat suitability modelling and niche theory. J. Appl. Ecol. 45, 1372–1381.

Hoang, T.H., Lock, K., Mouton, A.M., Goethals, P.L.M., 2010. Application of classification trees and support vector machines to model the presence of macroinvertebrates in rivers in Vietnam. Ecol. Inform. 5, 140–146.

Holland, J.H., 2000. Building blocks, cohort genetic algorithms, and hyperplane-defined functions. Evol. Comput. 8, 373–391.

Hutchinson, E.G., 1957. Concluding remarks. Cold Spring Harbor Symp. Quantit. Biol. 159, 415–427.

Hutter, F., Hoos, H.H., Leyton-Brown, K., 2011. Sequential model-based optimization for general algorithm configuration. In: Proceedings of the 5th International Conference on Learning and Intelligent Optimization. Springer-Verlag, Berlin, Heidelberg. pp. 507–523.

Iverson, L.R., Prasad, A.M., 1998. Predicting abundance of 80 tree species following climate change in the eastern United States. Ecol. Monogr. 68, 465–485.

Jakeman, A.J., Letcher, R.A., Norton, J.P., 2006. Ten iterative steps in development and evaluation of environmental models. Ecol. Model. Softw. 21, 602–614.

Jeong, K.S., Jang, J.D., Kim, D.K., Joo, G.J., 2011. Waterfowls habitat modeling: simulation of nest site selection for the migratory Little Tern (*Sterna albifrons*) in the Nakdong estuary. Ecol. Model. 222, 3149–3156.

Kacprzyk, J., Pedrycz, W., 2015. Springer Handbook of Computational Intelligence, 1st ed. Springer-Verlag, Berlin, Heidelberg.

Kennedy, J., Eberhart, R.C., 1997. A discrete binary version of the particle swarm algorithm. 1997 IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation 4104–4108.

Kissling, W.D., Dormann, C.F., Groeneveld, J., Hickler, T., Kühn, I., McInerny, G.J., Montoya, J.M., Römermann, C., Schiffers, K., Schurr, F.M., Singer, A., Svenning, J.C., Zimmermann, N.E., O'Hara, R.B., 2012. Towards novel approaches to modelling biotic interactions in multispecies assemblages at large spatial extents. J. Biogeogr. 39, 2163–2178.

Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International Joint Conference on Artificial Intelligence (IJCAI), vol. 2. Montreal. pp. 1137–1143.

Kotthoff, L., Thornton, C., Hoos, H.H., Hutter, F., Leyton-Brown, K., 2016. Auto-WEKA 2.0: automatic model selection and hyperparameter optimization in WEKA. J. Mach. Learn. Res. 17, 1–5.

Leibold, M.A., Holyoak, M., Mouquet, N., Amarasekare, P., Chase, J.M., Hoopes, M.F., Holt, R.D., Shurin, J.B., Law, R., Tilman, D., Loreau, M., Gonzalez, A., 2004. The metacommunity concept: a framework for multi-scale community ecology. Ecol. Lett. 7, 601–613.

López-Ibá nez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., Stützle, T., 2016. The irace package: iterated racing for automatic algorithm configuration. Oper. Res. Perspect. 3, 43–58.

Maier, H., Simpson, A., Zecchin, A., Foong, W., Phang, K., Seah, H., Tan, C., 2003. Ant colony optimization for design of water distribution systems. J. Water Resour. Plan. Manage. 129, 200–209.

Maier, H.R., Kapelan, Z., Kasprzyk, J., Kollat, J., Matott, L.S., Cunha, M.C., Dandy, G.C., Gibbs, M.S., Keedwell, E., Marchi, A., Ostfeld, A., Savic, D., Solomatine, D.P., Vrugt, J.A., Zecchin, A.C., Minsker, B.S., Barbour, E.J., Kuczera, G., Pasha, F., Castelletti, A., Giuliani, M., Reed, P.M., 2014. Evolutionary algorithms and other metaheuristics in water resources: current status, research challenges and future directions. Ecol. Model. Softw. 62, 271–299.

Manel, S., Ceri Williams, H., Ormerod, S.J., 2001. Evaluating presence-absence models in ecology: the need to account for prevalence. J. Appl. Ecol. 38, 921–931.

May, R.J., Dandy, G.C., Maier, H.R., Nixon, J.B., 2008. Application of partial mutual information variable selection to ANN forecasting of water quality in water distribution systems. Ecol. Model. Softw. 23, 1289–1299.

McClean, C.J., Lovett, J.C., Küper, W., Hannah, L., Sommer, H.J., Wilhelm, B., Termansen, M., Smith, G.F., Tokumine, S., Taplin, J.R.D., 2005. African plant diversity and climate change. Ann. MI Bot. Garden 92, 139–152.

Mcdermott, J., O'Reilly, U.m., 2015. Genetic programming. In: Kacprzyk, J., Pedrycz, W. (Eds.), Springer Handbook of Computational Intelligence, 1st ed. Springer-Verlag, Berlin, Heidelberg, pp. 845–869.

McKay, R.I., 2001. Variants of genetic programming for species distribution modelling – fitness sharing, partial functions, population evaluation. Ecol. Model. 146, 231–241.

Merow, C., Smith, M.J., Edwards, T.C., Guisan, A., Mcmahon, S.M., Normand, S., Thuiller, W., Wüest, R.O., Zimmermann, N.E., Elith, J., 2014. What do we gain from simplicity versus complexity in species distribution models? Ecography 37, 1267–1281.

Meyers, G., Kapelan, Z., Keedwell, E., 2017. Short-term forecasting of turbidity in trunk main networks. Water Res. 124, 67–76.

Mount, N., Maier, H., Toth, E., Elshorbagy, A., Solomatine, D., Chang, F.J., Abrahart, R., 2016. Data-driven modelling approaches for socio-hydrology: opportunities and challenges within the Panta Rhei Science Plan. Hydrol.Sci. J. 61, 1192–1208.

Mouton, A.M., Alcaraz-Hernández, J.D., De Baets, B., Goethals, P.L.M., Martínez-Capel, F., 2011. Data-driven fuzzy habitat suitability models for brown trout in Spanish Mediterranean rivers. Ecol. Model. Softw. 26, 615–622.

Mouton, A.M., De Baets, B., Goethals, P.L.M., 2010. Ecological relevance of performance criteria for species distribution models. Ecol. Model. 221, 1995–2002.

Mouton, A.M., De Baets, B., Van Broekhoven, E., Goethals, P.L.M., 2009. Prevalence-adjusted optimisation of fuzzy models for species distribution. Ecol. Model. 220, 1776–1786.

Muñoz-Mas, R., Fukuda, S., Vezza, P., Martínez-Capel, F., 2016a. Comparing four methods for decision-tree induction: a case study on the invasive Iberian gudgeon (*Gobio lozanoi*; Doadrio and Madeira, 2004). Ecol. Inform. 34, 22–34.

Muñoz-Mas, R., Martínez-Capel, F., Alcaraz-Hernández, J.D., Mouton, A.M., 2017. On species distribution modelling, spatial scales and environmental flow assessment with multi-layer perceptron ensembles: a case study on the redfin barbel (*Barbus haasi*; Mertens, 1925). Limnologica 62, 161–172.

Muñoz-Mas, R., Vezza, P., Alcaraz-Hernández, J.D., Martínez-Capel, F., 2016. Risk of invasion predicted with support vector machines: a case study on northern pike (*Esox lucius*, L.) and bleak (*Alburnus alburnus*, L.). Ecol. Model. 342, 123–134.

Nelder, J.A., Wedderburn, R.W.M., 1972. Generalized linear models. J. R. Stat. Soc. Ser. A: Gen. 135, 370–384.

Nix, H.A., 1986. A biogeographic analysis of Australian elapid snakes. In: In: Longmore, R. (Ed.), Australian Flora and Fauna, vol. 7. Australian Government Publishing Service, Canberra, pp. 4–15.

Nix, H.A., McMahon, J., Mackenzie, D., 1977. No Potential areas of production and the future of pigeon pea and other grain legumes in Australia. In: Wallis, E., Whiteman, P. (Eds.), The Potential for Pigeon Pea in Australia: Proceedings of Pigeon Pea (*Cajanus cajan* (L.) Millsp.) Field Day. University of Queensland, Queensland. pp. 1–12 (Chapter 5).

Olden, J.D., Lawler, J.J.L., Poff, N.L., 2008. Machine learning ethods without tears: a primer for ecologists. Q. Rev. Biol. 83, 171–193.

Pal, S.K., Bandyopadhyay, S., Ray, S.S., 2006. Evolutionary computation in bioinformatics: a review. IEEE Trans. Syst. Man Cybern. A: Syst. Hum. 36, 601–615.

Pauwels, I.S., Goethals, P.L.M., Coeck, J., Mouton, A.M., 2014. Movement patterns of adult pike (*Esox lucius* L.) in a Belgian lowland river. Ecol. Freshw. Fish 23, 373–382.

Pelikan, M., Hauschild, M.W., Lobo, F.G., 2015. Estimation of distribution algorithms. In: Kacprzyk, J., Pedrycz, W. (Eds.), Springer Handbook of Computational Intelligence, 1st ed. Springer-Verlag, Berlin, Heidelberg, pp. 899–928.

Penn, R., Friedler, E., Ostfeld, A., 2013. Multi-objective evolutionary optimization for greywater reuse in municipal sewer systems. Water Res. 47, 5911–5920.

Peterson, A.T., Ortega-Huerta, M.A., Bartley, J., Sanchez-Cordero, V., Soberon, J., Buddenmeier, R.H., Stockwell, D.R.B., Sánchez-Cordero, V., Soberón, J., Buddemeier, R.H., Stockwell, D.R.B., 2002. Future projections for Mexican faunas under global climate change scenarios. Nature 416, 626–629.

Peterson, A.T., Papes, M., Eaton, M., 2007. Transferability and model evaluation in ecological niche modeling: a comparison of GARP and Maxent. Ecography 30, 550–560.

Phillips, S.J., Anderson, R.P., Dudík, M., Schapire, R.E., Blair, M.E., 2017. Opening the black box: an open-source release of Maxent. Ecography 40, 887–893.

Phillips, S.J., Anderson, R.P., Schapire, R.E., 2006. Maximum entroy modeling of species geographic distributions. Ecol. Model. 190, 231–252.

Prasad, A.M., Iverson, L.R., Liaw, A., 2006. Newer classification and regression tree techniques: bagging and random forests for ecological prediction. Ecosystems 9, 181–199.

Quinlan Ross, J., 1993. C4. 5: Programs For Machine Learning. https://doi.org/10.1016/S0019-9958(62)90649-6.

Rauch, W., Harremoës, P., 1999. Genetic algorithms in real time control applied to minimize transient pollution from urban wastewater systems. Water Res. 33, 1265–1277.

Rokach, L., Maimon, O., 2015. Data Mining with Decision Trees: Theory and Applications, 2nd ed. World Scientific Publishing Co. Pte. Ltd., Singapore.

Rowe, J.E., 2015. Genetic algorithms. In: Kacprzyk, J., Pedrycz, W. (Eds.), Springer

Handbook of Computational Intelligence, 1st ed. Springer-Verlag, Berlin, Heidelberg, pp. 825–844.

Sadeghi, R., Zarkami, R., Van Damme, P., 2014. Modelling habitat preference of an alien aquatic fern, *Azolla filiculoides* (Lam.), in Anzali wetland (Iran) using data-driven methods. Ecol. Model. 284, 1–9.

Sadeghia, R., Zarkami, R., Sabetraftar, K., Van Damme, P., 2013. Application of genetic algorithm and greedy stepwise to select input variables in classification tree models for the prediction of habitat requirements of *Azolla filiculoides* (Lam.) in Anzali wetland, Iran. Ecol. Model. 251, 44–53.

Saeys, Y., Inza, I., Larra naga, P., 2007. A review of feature selection techniques in bioinformatics. Bioinformatics 23, 2507–2517.

Scheerlinck, K., Pauwels, V.R.N., Vernieuwe, H., De Baets, B., 2009. Calibration of a water and energy balance model: recursive parameter estimation versus particle swarm optimization. Water Resour. Res. 45, W10422.

Sirbu, A., Ruskin, H., Crane, M., 2010. Comparison of evolutionary algorithms in gene regulatory network model inference. BMC Bioinform. 11, 59.

Srikanth, R., George, R., Warsi, N., Prabhu, D., Petry, F.E., Buckles, B.P., 1995. A variable-length genetic algorithm for clustering and classification. Pattern Recogn. Lett. 16, 789–800.

Stockwell, D.R.B., Noble, I.R., 1992. Induction of sets of rules from animal distribution data: A robust and informative method of data analysis. Math. Comput. Simul. 33, 385–390.

Sutherst, R.W., Maywald, G.F., 1985. A computerised system for matching climates in ecology. Agric. Ecosyst. Environ. 13, 281–299.

Sweetapple, C., Fu, G., Butler, D., 2014. Multi-objective optimisation of wastewater treatment plant control to reduce greenhouse gas emissions. Water Res. 55, 52–62.

Szemis, J.M., Maier, H.R., Dandy, G.C., 2012. A framework for using ant colony optimization to schedule environmental flow management alternatives for rivers, etlands, and floodplains. Water Resour. Res. 48, 1–21.

Termansen, M., McClean, C.J., Preston, C.D., 2006. The use of genetic algorithms and Bayesian classification to model species distributions. Ecol. Model. 192, 410–424.

Thuiller, W., 2003. BIOMOD – optimizing predictions of species distributions and projecting potential future shifts under global change. Global Change Biol. 9, 1353–1362.

Thuiller, W., Lafourcade, B., Engler, R., Araújo, M.B., 2009. BIOMOD – a platform for ensemble forecasting of species distributions. Ecography 32, 369–373.

Tirelli, T., Pessani, D., 2009. Use of decision tree and artificial neural network approaches to model presence/absence of Telestes muticellus in piedmont (North-Western Italy). River Res. Appl. 25, 1001–1012.

Van Broekhoven, E., Adriaenssens, V., De Baets, B., 2007. Interpretability-preserving genetic optimization of linguistic terms in fuzzy models for fuzzy ordered classification: an ecological case study. Int. J. Approximate Reason. 44, 65–90.

Van Broekhoven, E., Adriaenssens, V., De Baets, B., Verdonschot, P.F., 2006. Fuzzy rule-based macroinvertebrate habitat suitability models for running waters. Ecol. Model. 198, 71–84.

Vayghan, A.H., Zarkami, R., Sadeghi, R., Fazli, H., 2016. Modeling habitat preferences of *Caspian kutum*, *Rutilus frisii kutum* (Kamensky, 1901) (Actinopterygii, Cypriniformes) in the Caspian Sea. Hydrobiologia 766, 103–119.

Verberk, W., Verdonschot, P., van Haaren, T., van Maanen, B., 2012. Milieu-en habitat-preferenties van Nederlandse zoetwatermacrofauna. Technical Report. STOWA, Eindhoven.

Vezza, P., Muñoz-Mas, R., Martinez-Capel, F., Mouton, A.M., 2015. Random forests to evaluate biotic interactions in fish distribution modelse. Ecol. Model. Softw. 67, 173–183.

VMM, 2018. Flemish Environment Agency. URL: https://www.vmm.be, accessed on 20.09.2018.

Weise, T., 2009. Global Optimization Algorithms: Theory and Application, vol. 1.

Whigham, P.A., 2000. Induction of a marsupial density model using genetic programming and spatial relationships. Ecol. Model. 131, 299–317.

Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1, 67–82.

Zarkami, R., Sadeghi, R., Goethals, P.L.M., 2012. Use of fish distribution modelling for river management. Ecol. Model. 230, 44–49.

Zarkami, R., Sadeghi, R., Goethals, P.L.M., 2014. Modelling occurrence of roach “Rutilus rutilus” in streams. Aquat. Ecol. 48, 161–177.

Zecchin, A.C., Simpson, A.R., Maier, H.R., Leonard, M., Roberts, A.J., Berrisford, M.J., 2006. Application of two ant colony optimisation algorithms to water distribution system optimisation. Math. Comput. Model. 44, 451–468.

Zuur, A.F., Ieno, E.N., Elphick, C.S., 2010. A protocol for data exploration to avoid common statistical problems. Methods Ecol. Evol. 1, 3–14.

Zuur, A.F., Ieno, E.N., Walker, N.J., Saveliev, A.A., Smith, G.M., 2009. Mixed Effects Models and Extensions in Ecology with R, 1st ed. Springer Science + Business Media, New York.