Environmental Modelling and Software 114 (2019) 195–213

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

Introductory overview: Optimization using evolutionary algorithms and other metaheuristics

H.R. Maier[a,*], S. Razavi[b,c], Z. Kapelan[d,h], L.S. Matott[e], J. Kasprzyk[f], B.A. Tolson[g]

[a] School of Civil, Environmental and Mining Engineering, The University of Adelaide, Adelaide, SA, 5005, Australia
[b] Global Institute for Water Security & School of Environment and Sustainability, University of Saskatchewan, Saskatoon, Saskatchewan, Canada
[c] Department of Civil and Geological Engineering, University of Saskatchewan, Saskatoon, Saskatchewan, Canada
[d] College of Engineering, Mathematics and Physical Sciences, University of Exeter, Harrison Building, North Park Road, Exeter, EX4 4QF, United Kingdom
[e] Center for Computational Research, University at Buffalo, Buffalo, NY, 14203, USA
[f] Civil Environmental and Architectural Engineering Department, University of Colorado Boulder, UCB 428, Boulder, CO, 80309, USA
[g] Department of Civil and Environmental Engineering, University of Waterloo, Waterloo, ON, N2L 3G1, Canada
[h] Department of Water Management, Faculty of Civil Engineering and Geosciences, Delft University of Technology, Stevinweg 1, Delft, CN 2628, Netherlands

A B S T R A C T

Environmental models are used extensively to evaluate the effectiveness of a range of design, planning, operational, management and policy options. However, the number of options that can be evaluated manually is generally limited, making it difficult to identify the most suitable options to consider in decision-making processes. By linking environmental models with evolutionary and other metaheuristic optimization algorithms, the decision options that make best use of scarce resources, achieve the best environmental outcomes for a given budget or provide the best trade-offs between competing objectives can be identified. This Introductory Overview presents reasons for embedding formal optimization approaches in environmental decision-making processes, details how environmental problems are formulated as optimization problems and outlines how single- and multi-objective optimization approaches find good solutions to environmental problems. Practical guidance and potential challenges are also provided.

Learning Objectives

As Introductory Overviews are designed to provide a concise topic overview so as to break down barriers to shared understanding and dialogue within multidisciplinary teams, the learning objectives of this paper are:

- To gain an appreciation of the suitability and benefits of using formal optimization approaches, and evolutionary algorithms (EAs) in particular, to support decision-making when using environmental models
- To gain an understanding of how to formulate and solve environmental problems as optimization problems
- To gain an understanding of the way single- and multi-objective EAs work
- To gain an understanding of how to implement EAs in practice
- To gain an understanding of potential challenges associated with the implementation of EAs

Assumed Background Knowledge

- Familiarity with mathematics and environmental modelling
- Familiarity with the use of environmental models to support decision-making
- Familiarity with the calibration of environmental models

1. Introduction

Many environmental decision-making problems can be cast as trying to find a preferred option among different alternatives. The use of environmental models can play a central role in this task (Merritt et al., 2017), as they can be used to assess the utility of different decision alternatives (e.g. infrastructure interventions, management strategies, policy options). However, identifying the most suitable option can be difficult, as environmental problems are generally complex and the number of available alternatives is often large. Consequently, significant benefits can be achieved by linking simulation models with a modern family of optimization techniques referred to as "metaheuristics" in order to identify the options that make best use of scarce resources, achieve the best environmental outcomes for a given budget, or provide the best trade-offs between competing objectives for further scrutiny. This enables decision-making processes to focus on options that provide the "best bang for buck", rather than options that are potentially less reliable, more expensive and achieve worse environmental outcomes.

The term "metaheuristic" (coined by Glover, 1986) is composed of two Greek words, the prefix "meta" meaning "beyond" and "heuristic"

* Corresponding author.
E-mail address: holger.maier@adelaide.edu.au (H.R. Maier).

https://doi.org/10.1016/j.envsoft.2018.11.018
Received 12 June 2018; Received in revised form 27 November 2018; Accepted 27 November 2018
Available online 01 December 2018

meaning "to discover". From an optimization perspective, metaheuristics are high level procedures designed to intelligently utilize heuristics to efficiently find near-optimal solutions to complex optimization problems (for definitions, see Blum and Roli, 2003). From an environmental management perspectives, they provide a means to make use of (existing) environmental models to identify a set of solutions (decision alternatives) that provide (near) optimal trade-offs between desired outcomes. Many metaheuristics are "nature-inspired" (also alternatively called "metaphor-based"), as they involve heuristics that imitate processes in natural systems (Blum et al., 2011; Boussaïd et al., 2013; Dréo et al., 2006; Zavala et al., 2014). Metaheuristics can be divided into two classes, including those that work with populations of solutions and those that do not. Evolutionary algorithms (EAs), which are population based, are a major, and arguably the most popular, class of metaheuristics. Consequently, as was done in Maier et al. (2014), the focus of this paper is on EAs, although many of the concepts discussed also broadly apply to other metaheuristics.

EAs have proven to be highly effective in solving a wide range of environmental problems (Maier et al., 2014), as they can be used in realistic decision-contexts. This is because they (i) can be used as one element in broader, participatory environmental decision-making processes (Di Matteo et al., 2017a; Kaim et al., 2018; Piscopo et al., 2015; Wu et al., 2016), (ii) can be linked with existing simulation models to assist with the exploration of large solution spaces (Maier et al., 2015), (iii) can cater to multiple competing objectives (Newland et al., 2018; Verstegen et al., 2017), (iv) can take into account uncertainties (Beh et al., 2017; Eker and Kwakkel, 2018) and (v) generally provide a number of "good" solutions that can be explored further, rather than being prescriptive (Di Matteo et al., 2017a; Kaim et al., 2018). Their main disadvantages are (i) that they are potentially computationally expensive, although this primarily depends on the computational efficiency of the simulation model with which they are linked, (ii) that they are not guaranteed to identify the globally optimal solution (from a mathematical perspective) and (iii) that they generally need to be tuned to the problem under consideration (Maier et al., 2014; Mala-Jetmarova et al., 2017).

The purpose of this Introductory Overview is to introduce EAs to the environmental modelling community, where environmental models are commonly used to assist with identifying the course of action that should be undertaken (e.g. pollution control, operation of infrastructure system, implementation of policy option) to produce desired environmental outcomes, but without formal strategies for achieving this. Consequently, before details of EAs are given, a case is made for why formal optimization methods can be useful where environmental models are used to support decisions. In addition, details are given of how to structure environmental decision problems to enable such optimization methods to be used. Then, information is provided on how EAs work, how EAs can be implemented and what some of the challenges are when applying EAs to real-world problems. For ease of understanding, key concepts related to the way decision problems are structured are introduced in the context of five classes of environmental problems, which are illustrated via five specific examples:

- **The mitigation of global climate change impacts:** This problem type is illustrated via the example of designing water distribution systems so as to minimize their impact on greenhouse gas emissions. As climate change is caused by anthropogenic systems, mitigation options typically require modifications to be made to engineered systems. In the specific example presented here, this corresponds to the selection of the components of the systems of pipes, tanks, pumps and valves that deliver water from sources (e.g. rivers, lakes, groundwater) to domestic, industrial and agricultural consumers. Such water distribution systems contribute to greenhouse gas emissions through both the embodied energy of their components (e.g. pipes, tanks) and the energy required for operating pumps to maintain the required system pressures. By optimizing the design of these systems, the best possible trade-offs between cost and climate change impacts (via greenhouse gas emissions) can be identified, while ensuring such systems perform reliably.
- **Regional resource consumption in urban systems:** This problem type is illustrated via the example of the long-term planning of water resources systems, where optimization can be used to ensure that the best possible balance between the benefits (i.e. water supply security) and impacts (e.g. social and environmental impacts of water extraction) of natural resource consumption can be identified.
- **Natural resource management:** This problem type is illustrated via the example of the operation of reservoir systems, where optimization can be used to determine how to operate an infrastructure system so as to achieve the best possible trade-offs between various economic, social and environmental outcomes in catchments (watersheds).
- **Pollution management in an environmental system:** This problem type is illustrated via the example of the determination of wastewater treatment levels, where optimization can be used to determine the lowest-cost treatment strategies that maintain environmental health in a river system.
- **Environmental model development:** This problem type is illustrated via the example of model calibration, where optimization can be used to identify the model parameters that provide the best match between model outputs and the corresponding measured data.

It should be noted that the above example applications are illustrative only and were selected to ensure that the principles presented are accessible to as wide an audience as possible, which is one of the objectives of Introductory Overviews. This is why examples belonging to five different classes of problems are presented, so that the underlying principles are likely to resonate with as many readers as possible. However, the basic principles on optimization and evolutionary algorithms presented in this paper are applicable to all problems where environmental models are calibrated or used to support decisions (see Section 2). It should also be noted that as this paper belongs to the Introductory Overview class of papers, it presents fundamental principles in a simple, easy-to-understand fashion. More detailed descriptions can be found elsewhere (e.g. Maier et al., 2014).

2. Why do we need optimization?

As mentioned in the Introduction, environmental simulation models are used extensively to support decision-making processes in a variety of application areas, such as: the development and evaluation of national and international environmental regulations (Giupponi, 2007; Laniak et al., 2013); land use management (Amato et al., 2018); natural hazard management (Newman et al., 2017); the operation and management of reservoir systems (Razavi et al., 2013); the assessment of environmental and human health (Morley and Gulliver, 2018; Reis et al., 2015); the management of river systems (He, 2003; Humphrey et al., 2016; Hunter et al., 2018; Ravalico et al., 2010); the management of drains (Humphrey et al., 2016); the management of air pollution (Baró et al., 2014; Borge et al., 2014); flood inundation assessment (Teng et al., 2017); groundwater management and remediation (Jakeman et al., 2016; Piscopo et al., 2015; Singh, 2014); the design of water distribution networks so as to minimize global climate impacts (Stokes et al., 2015a; Stokes et al., 2014b; Wu et al., 2010a); the prediction of and adaption to natural hazards such as floods or droughts (Basher, 2006); crop and livestock management (Moore et al., 2014; van Keulen and Asseng, 2018); the design of green infrastructure for stormwater management and urban renewal (Liu et al., 2014; Yigitcanlar and Teriman, 2015); and evaluating the effects of resource extraction by the petroleum (Fiori and Zalba, 2003), natural gas (McJeon et al., 2014), mining (Côte et al., 2010) and timber (Alavalapati and Adamowicz, 2000) industries. Environmental models are in such widespread use because they can be designed to effectively

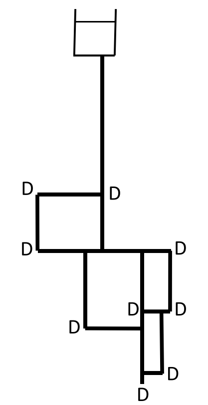
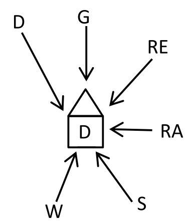
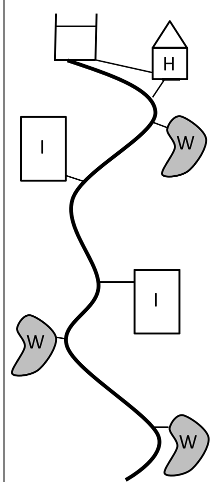
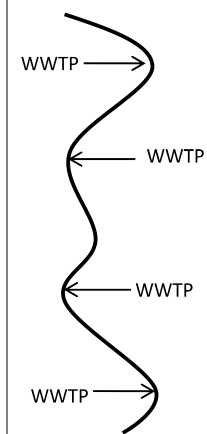
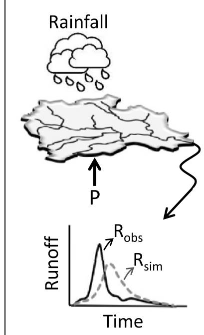
Problem	Design of Water Distribution Systems	Planning of Urban Water Supply Systems	Operation of Reservoir Systems	Wasteload Allocation	Model Calibration
	 Reservoir Pipe D Demand	 D Desalination G Groundwater RE Reservoir RA Rainwater S Stormwater W Wastewater D Demand	 Reservoir I Irrigation W Wetland H Hydropower River	 River WWTP Wastewater treatment plant	 River P Model Parameters R_{obs} Observed Runoff R_{sim} Simulated Runoff
Decision Variables	Pipe diameters	Supply source capacities, supply has to be greater than or equal to demand	Reservoir release amounts, duration and timing	Treatment levels	Model Parameters
Constraints	Minimum pressure in pipes	Supply capacity from various sources	Available water	Allowable water quality in river	Physically feasible parameter ranges
Objectives	Minimize: Cost, greenhouse gas emissions	Minimize: Cost of supply, energy usage, greenhouse gas emissions	Maximize: hydropower production, environmental health, crop production Minimize: Downstream flooding	Minimize: Total treatment costs	Minimize the model errors, e.g., difference between simulated and observed runoff

Fig. 1. Details of formulation of five example environmental problems, including problem representation as well as example decision variables, constraints and objectives. The five problems include the Design of Water Distribution Systems (e.g. Basupi et al., 2013; Herstein et al., 2010; Stokes et al., 2014a; Stokes et al., 2015b), the Planning of Urban Water Supply Systems (e.g. Kasprzyk et al., 2013; Paton et al., 2014; Wu et al., 2016, 2017), the Operation of Reservoir Systems (e.g. Razavi et al., 2013; Reddy and Kumar, 2006; Szemis et al., 2014; Teegavarapu and Simonovic, 2002), Wasteload Allocation (e.g. Burn and Yulianti, 2001; Cho et al., 2004; Vasquez et al., 2000; Yandamuri et al., 2006) and Model Calibration (e.g. Gupta et al., 1999; Razavi and Tolson, 2013; Vrugt et al., 2003; Yapo et al., 1998).

reproduce the dynamics of real-world systems under traditional management situations as well as *alternative* virtual realities, including different environmental conditions and management alternatives, enabling optimal designs, strategies and policies to be developed under a range of scenarios (Maier et al., 2016).

In order to use environmental simulation models to support decisions, the model inputs have to correspond to the proposed decision alternatives (e.g. infrastructure interventions, management strategies, policy options) and the model outputs have to correspond to the environmental (and other) outcomes of interest. In a decision-support context, the outcomes of interest are generally referred to as *objectives*, but can also correspond to system *constraints* (e.g. allowable pollution levels), and the available decision alternatives are referred to as *decision variables.* The environmental model is then used to find the values of the *decision variables* (i.e. model inputs) that optimize (maximize or minimize) the environmental (and other) *objectives* (i.e. corresponding values of model outputs) and ensure that any *constraints* are satisfied. The selected values for all *decision variable* are referred to as a *solution* to the problem under consideration. These terms are illustrated below for the five example environmental problems introduced in Section 1 (see also Fig. 1):

- **The design of water distribution systems to mitigate global climate change impacts:** For this problem, the *objectives* (i.e. outcomes of interest) are to minimize greenhouse gas emissions and system cost by selecting appropriate sizes of pipes, valves, pumps, tanks etc., which are the *decision variables* (i.e. interventions). The *constraints* are that water of sufficient pressure and quality has to be delivered to consumers. A *solution* consists of the selected sizes for the pipes, valves, pumps, tanks etc.
- **The long-term planning of water resources systems for managing regional resource consumption:** For this problem, the *objectives* (i.e. outcomes of interest) are to minimize the impact on available water resources, energy usage and system cost while maximizing water supply security by selecting an appropriate portfolio of supply and demand interventions, which are the *decision variables.* The *constraints* are the available water supply and demand management options. A *solution* consists of the selected portfolio of supply and demand interventions.
- **The operation of reservoir systems for natural resources management:** For this problem, the *objectives* (i.e. outcomes of interest) are to maximize environmental, social and economic benefits and to

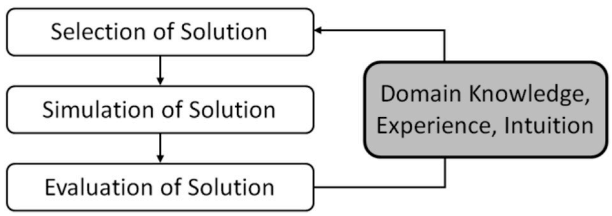


Fig. 2. Steps in a traditional informal "optimization" process for finding optimal solutions to environmental problems with the aid of environmental simulation models via trial-and-error.

minimize environmental, social and economic costs by selecting appropriate reservoir release magnitudes and durations over time, which are the decision *variables* (i.e. interventions). The *constraints* are the available water resources and limits on reservoir outflows. A *solution* consists of the selected reservoir release schedule.

- **The determination of wastewater treatment levels for pollution control in environmental systems:** For this problem, the outcomes of interest correspond to both *constraints* (i.e. the maintenance of acceptable water quality levels in the river system under consideration) and *objectives* (i.e. minimizing the cost of the interventions required to achieve the desired water quality outcomes). The *decision variables* (i.e. interventions) are the required wastewater treatment levels and a *solution* consists of the selected treatment levels at each wastewater treatment plant.
- **The calibration of simulation models:** For this problem, the *objectives* (i.e. outcomes of interest) are to minimize an error measure between simulated flux/state variables and corresponding measured flux/state variables by selecting appropriate values of the parameters of a simulation model, which are the decision *variables.* The *constraints* are the allowable ranges of the parameters. A *solution* consists of a selected value for each of the model parameters to be obtained by calibration.

Once an environmental problem has been formulated in terms of *objectives* (i.e. the values to be maximized or minimized), *decision variables* (i.e. the values that can be changed to maximize or minimize the objective(s)) and *constraints* (i.e. the allowable values of decision variables and/or objectives/constraints), the best *solutions* are generally identified using an informal "optimization" process, where the "best" solutions are identified via trial-and-error (Simon, 1996), as illustrated in Fig. 2. The first step of this process involves the selection of a plausible solution to the problem under consideration from the available alternatives. This selection is generally based on domain knowledge, experience and intuition. For example:

- For the water distribution system design problem, this would involve the selection of the diameters of all of the pipes in the network, the sizes and locations of all of the tanks and the types, locations and sizes of the valves and pumps.
- For the long-term water resources planning problem, this would involve the selection of the capacities of the various potential supply sources, as well as the various demand management strategies.
- For the reservoir operation problem, this would involve the selection of when to release water, how much to release and how long for.
- For the wastewater treatment problem, this would involve the selection of the treatment level at each wastewater treatment plant.
- For the model calibration problem, this would involve the selection of values for each model parameter to be obtained by calibration.

The performance of the selected solution in terms of objective function values and constraint violation (i.e. the outcomes of interest) is then evaluated, typically with the aid of one or more environmental simulation models. For example:

- For the water distribution system design problem, a hydraulic simulation model (e.g. Rossman, 2000) would typically be used to check whether minimum pressure constraints are violated for the selected pipe diameters, in addition to the calculation of the values of the cost and greenhouse gas emission objectives.
- For the long-term water resources planning problem, a river basin planning model would generally be used to simulate the interactions between various sources and demands throughout the year in order to enable the impact on water resources, supply cost, energy and greenhouse gas emissions, as well as reliability of supply, to be calculated for different combinations of water sources and extraction amounts/capacities and demand management strategies.
- For the reservoir operation problem, an integrated model would be required, linking reservoir releases to the hydropower, environmental health, crop production and flood protection objectives.
- For the wastewater treatment problem, a water quality model (e.g. Ambrose et al., 2017) would generally be used to check whether the water quality constraints in the river are violated for the selected treatment levels, in addition to the calculation of the corresponding treatment costs.
- For the model calibration problem, the simulation model to be calibrated (e.g. Arnold et al., 1998) would be run with the selected parameter values, and the error metric (see Bennett et al., 2013, for a comprehensive range of metrics) would be calculated between the simulated model outputs for a set of inputs and the corresponding measured outputs.

The next step in the informal "optimization" process involves selection of an alternative, hopefully improved, solution. This selection takes into account the performance of the solution that was selected initially, as well as experience, domain knowledge, and intuition. For example:

- For the water distribution system design problem, if the previously selected solution resulted in a violation of the pressure constraint in a particular part of the system, some of the pipe diameters in this region of the network might be increased, reducing pressure losses, but increasing cost and greenhouse gas emissions. Conversely, if the previously selected solution resulted in a large pressure excess in a particular part of the system, some of the pipe diameters in this region of the network might be decreased, increasing pressure losses, but decreasing cost and greenhouse gas emissions.
- For the long-term water resources planning problem, if the previously selected solution resulted in a demand shortfall, the capacities of one or more of the sources might be increased, resulting in increased cost and water resources impact. In contrast, if the previously selected solution resulted in a large supply excess, the capacities of one or more of the sources might be decreased, reducing cost and water resources impact.
- For the reservoir operation problem, if the previously selected solution resulted in unacceptably high levels of spillage (water flowing over the spillway and not generating power), less water might be held back in the reservoir prior to the spillage event, increasing storage capacity at the cost of reducing the amount of water available for irrigation and environmental flows. Conversely, if the previously selected solution resulted in very low levels of flood risk, more water might be stored in the reservoir for hydropower and environmental and irrigation usage.
- For the wastewater treatment problem, if the previously selected solution resulted in a violation of the water quality standard in the river, some of the upstream treatment levels might be increased, improving water quality, but increasing cost. Conversely, if the previously selected solution resulted in water quality that is above the required level, some of the treatment levels might be decreased, reducing water quality, but decreasing cost.
- For the model calibration problem, in the case of a rainfall-runoff

Table 1
Example search space sizes for the five environmental problems considered for illustration purposes in Fig. 1. It should be noted that for problems for which potential decision variable values are continuous, as is generally the case for model calibration, the size of the search space is theoretically infinite.

Problem	Number of decision variables	Number of decision variable options (levels)	Number of potential solutions i.e. size of search space	0.01% of number of potential solutions
Water distribution system design	100 pipes	6 diameters	$6^{100} = 6.5 \times 10^{77}$	6.5×10^{73}
Long-term water resources planning	25 potential sources	10 potential capacities	$10^{25} = 1.0 \times 10^{25}$	1.0×10^{21}
Reservoir operation	52 weeks	5 different releases	$5^{52} = 2.2 \times 10^{36}$	2.2×10^{32}
Wastewater treatment	20 WWTPs	4 treatment levels	$4^{20} = 1.1 \times 10^{12}$	1.1×10^{8}
Model Calibration	15 parameters	100 values	$100^{15} = 1.0 \times 10^{30}$	1.0×10^{34}

model, if the previously selected solution resulted in an underestimation of peak flows, the value of the runoff curve number parameter might be increased to intensify the runoff response to rainfall, and vice versa. Conversely, if that solution resulted in a longer time-to-peak than reality, the value of the roughness coefficient might be decreased to allow faster movement of water through channels.

The steps of selecting a trial solution, evaluating its performance with the aid of one or more environmental models and selecting a new (hopefully improved) solution based on experience and domain knowledge, as well as the performance of the previously selected solution(s), are repeated until no further improvement in objective function values can be achieved or the analyst is satisfied with the selected solution.

While the informal optimization approach described above works reasonably well when the problem under consideration and simulation model used are well understood by the analyst, and the number of decision variables and their feasible ranges (i.e., number of alternative solutions) are relatively small, it is unlikely to yield the best possible outcome for the majority of environmental problems. This is because the size of the solution space of typical environmental problems is extremely large, as illustrated in Table 1 for particular instances of the five example problems considered previously. While a vast majority of these solutions might not be considered reasonable from a practical perspective, and would thus be excluded from consideration as part of the informal optimization process outlined above, the resulting search spaces would still be extremely large, as shown in the last column of Table 1, which shows the size of the solution spaces after 99.99% of possible solutions have been discarded. As a result, it is extremely unlikely that optimal, or even near-optimal, solutions can be identified by using the informal optimization process outlined above, as only a very small fraction of the search space is explored. This problem is exacerbated by the fact that many environmental problems are complex and poorly understood, making it more difficult to bring domain knowledge to bear, especially for inexperienced analysts.

At the other end of the spectrum of possible solution approaches, every possible value in the solution space could be evaluated (i.e. complete enumeration of the search space), such that the best possible solution is guaranteed to be identified. However, this is generally not possible from a practical perspective, as the simulation times associated with the search space sizes in Table 1 are likely to be prohibitively long. Assuming each run of the simulation model(s) used to evaluate the utility of the solutions takes only one second, the total simulation times for the examples in Table 1 will range from ~35,000 to $\sim 2 \times 10^{70}$ years (for example, in the water distribution system problem, the total simulation time is calculated as $(6.5 \times 10^{77} \times 1\,\text{s})/(60\,\text{s} \times 60\,\text{min} \times 24\,\text{h} \times 365\,\text{day}) \sim 2 \times 10^{70}$ years). In real-world problems, simulation models can be computationally expensive, taking minutes or longer for a single run, making the full enumeration of all possible solutions even less feasible.

The fact that informal optimization approaches are unlikely to enable near-optimal solutions to be identified and that the evaluation of every possible solution is computationally intractable provides a compelling case for the use of *formal* optimization methods in order to identify optimal or near-optimal solutions to complex environmental problems in a reasonably computationally efficient manner. Thus, formal optimization approaches enable innovative solutions to complex problems to be identified and result in the most efficient use of increasingly constrained economic and natural resources. In fact, the use of formal optimization methods can result in improvements on the order of 10%–50% compared with the use of the informal trial-and-error process outlined in Fig. 2 (Liner and Maier, 2015). In addition, the use of formal optimization methods provides a more consistent approach to solving environmental problems with conflicting objectives (Liebman, 1976), especially for complex problems that are poorly understood (Di Matteo et al., 2017b).

3. How do we formulate and solve optimization problems?

3.1. Problem formulation

The formulation of formal optimization problems is very similar to that of problems that are solved using the informal optimization process outlined in Section 2, requiring the specification of the following three components:

- Objective functions, which represent the values that should be maximized (e.g. system performance) or minimized (e.g. environmental impact) (see Fig. 1).
- Decision variables, which are the values that can be manipulated in order to maximize or minimize the objective functions (see Fig. 1). These can take on discrete (e.g. integer) or continuous (real) values, depending on the problem under consideration.
- Constraints, which can generally be placed on the values that decision variables can take, or used to avoid undesirable/infeasible system responses (see Fig. 1).

A solution is defined as a set of selected values of the decision variables, and a feasible solution is one that satisfies all problem constraints. The quality of different solutions is evaluated using the objective function. As mentioned previously, the (environmental) outcomes of interest are represented by the objectives and constraints, which generally correspond to the outputs of environmental models, and the potential interventions/choices are represented by the decision variables, which generally correspond to the inputs of environmental models. Generally, decision variables correspond to anthropogenic changes to natural systems, either directly (e.g. revegetation) or via an engineered system (e.g. pollution control).

Formally, optimization problems are represented (Cohon and Marks, 1975) as a maximization problem of an objective function, subject to inequality (and sometimes equality) constraints, as follows:

$$\text{maximize } f(x) \tag{1}$$

subject to

$$g_i(x) \le 0,\ i = 1, 2, \ldots, m \tag{2}$$

$$x_{jl} \le x_j \le x_{ju}\ j = 1, 2, \ldots, n \tag{3}$$

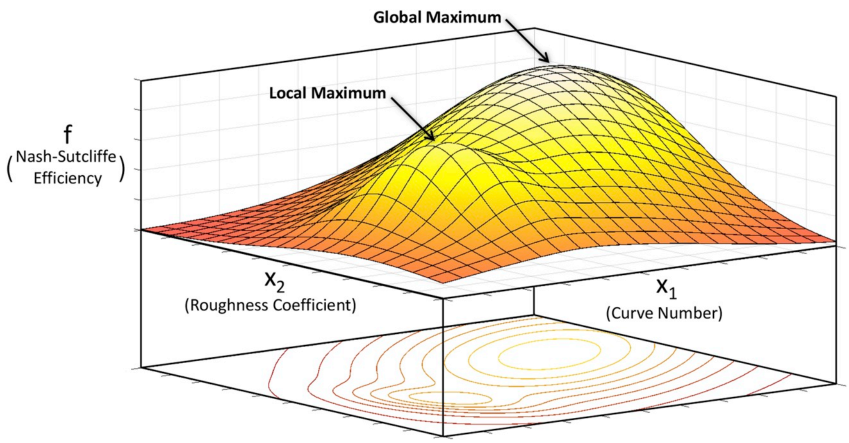


Fig. 3. A typical fitness landscape (response surface) and its contour plot for a model calibration problem, where the objective function f is a goodness-of-fit metric to be maximized and the decision variables x_1 and x_2 are model parameters. This optimization problem has two regions of attraction (one local maximum in addition to the global maximum).

where x is the vector of decision variables, x_j indicates the jth value in this vector, and x_{jl} and x_{ju} are lower and upper bounds on the decision variables, respectively; f is the objective function and g_i is the ith constraint function of vector x. These functions can be highly non-linear and complex, and their collective evaluation for any given x requires the running of environmental simulation models.

For minimization problems, the maximization formulation above can be easily transformed into a minimization formulation by multiplying the objective function by minus one (i.e., maximize -f(x) is equivalent to minimize f(x)). Also, the "single-objective" optimization formulation above can be extended to a "multi-objective" optimization formulation by introducing more objective functions to be maximized simultaneously in Eq. (1) above; that is, to maximize ($f_1(x)$, …, $f_k(x)$), where k is the number of objective functions.

It should be noted that the translation of environmental and water resources management problems into the somewhat rigid formulation outlined above can be challenging, especially for real-world problems, which are often complex, high-dimensional, ill-defined and not well structured. Consequently, where possible, stakeholders should be directly involved in the formulation of optimization problems, informing the objectives, decisions and constraints as part of an interactive process (Di Matteo et al., 2017a; Smith et al., 2017; Wu et al., 2016).

3.2. Problem representation

Optimization problems can be represented geometrically by considering a "fitness landscape". A fitness landscape, or synonymously called a "response surface", depicts the shape of the fitness (objective) function for a particular objective with respect to the decision variables (e.g. model error as a function of different values of model parameters for model calibration problems). For an optimization problem with only one decision variable, the resulting fitness landscape is a (typically nonlinear) line, and for a problem with two decision variables, the fitness landscape is a plane (see Fig. 3). For higher-dimensional problems with three or more decision variables, the fitness landscape becomes a hyperplane that cannot be visualised easily.

When there is a single objective to be maximized, there is a single fitness landscape and the purpose of optimization is to identify the highest peak (or the lowest trough for minimization problems) in the fitness landscape for the problem under consideration. In other words, the purpose of optimization is to identify the feasible combination of decision variable values that results in the largest value of the objective function (fitness). Consequently, the optimization process is akin to identifying the "highest hill" in the fitness landscape, which is referred to as the "global optimum", as it is the best performing among all possible solutions. It should be noted that there may exist one or multiple other hills in the fitness landscape with lower peaks, which are referred to as "local optima", as their respective solutions are optimal only within their *neighbourhood* in the decision variable space.

When there is more than one objective, each objective has its own fitness landscape, as variations in objective values with changes in decision variable values are likely to be different for different objectives. For many problems, objectives compete with each other, so that solutions that improve values of one objective might degrade values in another. For example, a reservoir release schedule that increases ecological health in the river is unlikely to also increase agricultural production. Consequently, when considering competing objectives, it is less clear which solutions are better than others, as the solution that results in the highest peak in the fitness landscape for one objective might result in the lowest peak in the fitness landscape for the other objective and vice versa. In such cases, the optimality of a solution is determined using the concept of dominance.

If a solution (x_i) performs better than another (x_j) in at least one objective and does not perform worse in any of the other objectives (e.g. $f_1(x_i) < f_1(x_j)$ and $f_l(x_i) <= f_l(x_j)$ for $l = 2 \ldots k$), then the first solution (x_i) is said to dominate the other solution (x_j), as it is clearly better with respect to all objectives. However, given a pair of solutions (x_p, x_q) where each solution performs better than the other in at least one objective (e.g. $f_1(x_p) < f_1(x_q)$ and $f_2(x_p) > f_2(x_q)$), it is not possible to determine which solution is best without some value judgement about the relative importance of different objectives. Consequently, for problems with more than one competing objective, it is not possible to determine a single best solution, as there are a number of solutions (e.g. x_p and x_q above) that are not dominated by any others. This set of non-dominated solutions forms the Pareto front (Pareto, 1896), which is unique and provides the optimal trade-off between objectives.

The above concepts are illustrated in Figs. 4 and 5 for a hypothetical problem with two decision variables (2-dimensional decision space) and two objective functions (2-dimensional objective space), both of which are to be minimized. Fig. 4 illustrates the mapping from the solution to the objective space, which is generally done with the aid of one or more simulation models. As can be seen, the solutions that are non-dominated in the objective space lie on the Pareto front, whereas dominated solutions do not. The relationship between the different fitness landscapes for the two objectives and the Pareto front is illustrated in Fig. 5.

3.3. Problem solution

As mentioned previously, the solution of single objective optimization problems corresponds to the process of searching the fitness landscape for the highest peak or lowest trough, depending on whether the objective is to maximize or minimize the objective function. For problems with more than one objective, the optimization process corresponds to searching the fitness landscapes for each of the objectives so as to identify non-dominated solutions. As the globally optimal

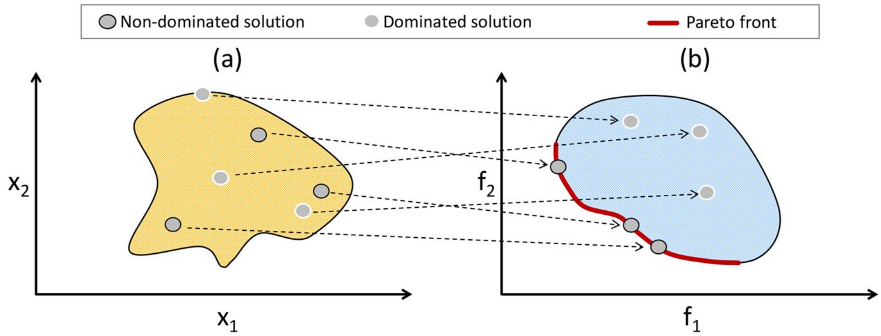


Fig. 4. Mapping of (a) a decision space onto (b) an objective space, where both objectives are to be minimized.

objective function value(s), and the corresponding decision variable values (i.e. the solution), are unknown for real-life optimization problems (i.e. the whole purpose of the optimization process is to identify these), it is not possible to assess how well an optimization algorithm has performed. The exception to this is model calibration, where the "best" value that can possibly be achieved (i.e. a calibration error of zero) is known. Consequently, split sample testing (i.e. using a subset of the available data for calibration and a subset for validation) can be used for these types of optimization problems to obtain a measure of the "robustness" of the calibration process. This, however, is generally affected by a number of factors in addition to the optimization method used, including the statistical properties of the available data and how they are divided in calibration and validation subsets (Wu et al., 2013; Zheng et al., 2018).

How difficult it is to search the fitness landscape(s) and find the globally optimal solution(s) is dependent on the following factors:

1. The size of the area to be searched. The size of the search space is a function of the number of decision variables (e.g. how many pipes have to be sized in a water distribution system), the ranges of the decision variables (e.g. what is the smallest and what is the largest diameter to be considered), and the level of discretization (resolution) in cases where discrete variables are considered (e.g. how many commercially available diameters are considered within the selected range of diameters). The larger the number of decision variables, the larger the ranges of the decision variables and the finer the resolution, the larger the search space and the more difficult it is likely to be to identify the optimal solution(s).

However, it is not only the size of the total search space that matters, but also the size of the feasible portion of this space (i.e. the size of the space for which all constraints are met). For some problems, many of the decision variable combinations result in solutions that violate constraints related to target system performance (i.e. infeasible solutions). For example, for the wastewater treatment problem (Fig. 1), various combinations of treatment levels at the WWTPs might result in better objective function values (i.e. lower treatment costs), but might result in water quality levels in the receiving waters that do not satisfy minimum water quality constraints. If the size of the feasible portion of the decision variable space is small, it might be difficult to find a combination of decision variables that results in feasible solutions. Consequently, the ability to find optimal solutions can not only be made more difficult if the size of the search space is very large, but also if the feasible region of the total search space is very small.

2. The properties of the fitness landscape. If the fitness landscape is very smooth, like the "big bowl" in Fig. 6a, the search is relatively easy, as there is a single optimum (for a minimization problem). However, if the fitness landscape is very rugged, as shown in Fig. 6b, finding the lowest point in the landscape is very difficult. This is because there are many troughs, making it difficult to navigate the landscape and to know if a trough that has been found is the lowest trough (i.e. the global optimum), or just a local optimum. For real-world problems, fitness landscapes tend to be complex (e.g. Gibbs et al., 2011, 2015; Kingston et al., 2005), characterised by a range of features from small-scale features, such as roughness or noise, to large-scale features, such as multimodality (Razavi and Gupta, 2015).

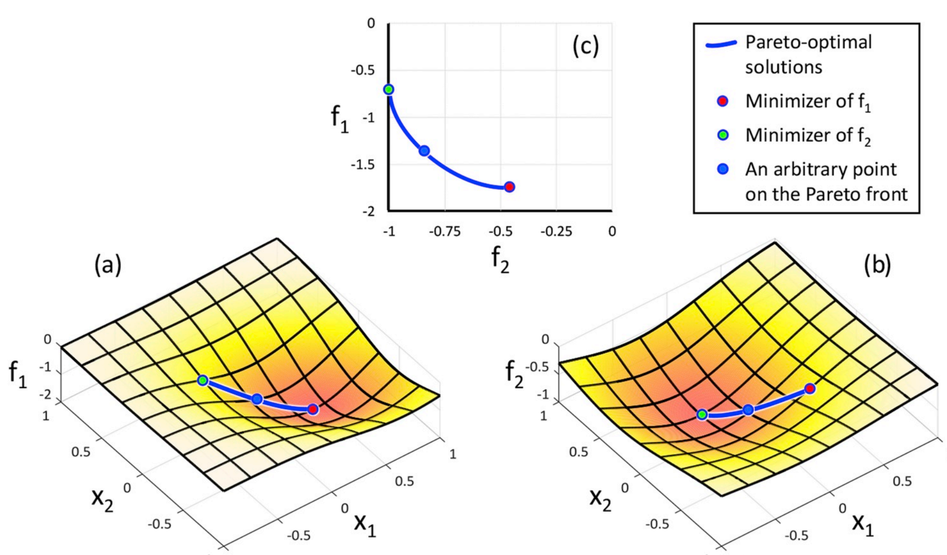


Fig. 5. Illustration of the relationship between (a) the fitness landscape of objective 1, (b) the fitness landscape of objective 2 and (c) the Pareto-front of the two-objective optimization problem.

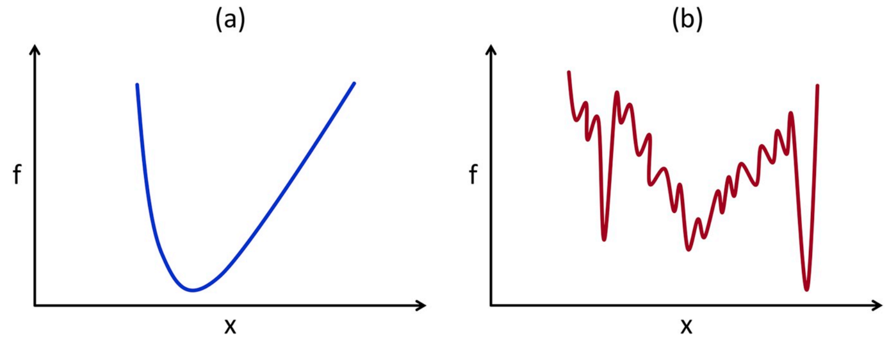


Fig. 6. Illustrations of smooth (a) and rugged (b) fitness landscapes for minimization problems, where f represents an objective function and x represents a decision variable.

Consequently, the fitness landscapes for many environmental problems are likely to be rugged and characterised by many local optima, a large degree of non-linearity and possibly discontinuities, especially if objective function values are calculated with the aid of complex simulation or integrated assessment models (Hamilton et al., 2015).

Although increasing the difficulty of the optimization problem, the presence of multiple local optima with similar objective function values in the fitness landscape can be either advantageous or disadvantageous from a practical perspective. A potential advantage is the provision of choice (diversity) to decision-makers in relation to which option to implement based on factors not formally considered during the optimization. For example, if there are a number of equally "good" management options, these can be fed into a participatory decision-making process to determine which should be implemented. However, the existence of multiple local optima with similar objective function values in the fitness landscape creates problems with model identifiability during model calibration (Shin et al., 2015), as it is difficult to determine which set of model parameters is the "correct" one. Another potential complicating factor related to model calibration is that the global optimum in the fitness landscape might not necessarily correspond to the model parameterization that results in the model that best represents the underlying physical processes. This is because for model calibration problems, the fitness landscape is a function of which error measure is used, how much data are available for calibration and which subset of the available data are used for model calibration and which subset for model validation (e.g. see Zheng et al., 2018), all of which can have a significant impact on the properties of the fitness landscape.

3. **The behaviour of the search method**. How well a search space of a particular size and ruggedness can be explored, both in terms of the ability to find globally optimal or near-globally optimal solutions and the computational efficiency with which this is achieved, is a function of the way a particular optimization algorithm navigates the fitness landscape(s). All optimization algorithms generally search the fitness landscape(s) in a stepwise fashion, starting from a random point and making incremental improvements over a number of iterations, as formalized in Equation (4), with the only difference between optimization algorithms being the mechanism that is used decide how to make adjustments to the decision variables from one iteration to the next (i.e. Δx_{t-1}). If we write

$$x_t = x_{t-1} + \Delta x_{t-1} \tag{4}$$

where t denotes the iteration number, then x_{t-1} is the vector of decision variables (solutions) in the previous iteration, x_t is the updated vector of decision variables, and Δx_{t-1} denotes the change in the vector of decision variables from one iteration to the next.

The adjustment mechanisms used by different optimization algorithms are generally characterised by two competing processes: diversification (exploration) and intensification (exploitation). The goal of diversification is to explore the search space as widely as possible. This is an advantage if the search space is rugged and characterised by a large number of local optima, as illustrated in Fig. 6b, so that the region containing the global optimum can be found. This is particularly important for large search spaces. However, an algorithm that explores more than is needed is less computationally efficient and, while it might be able to find the region of the search space that contains the global optimum, it might not be able to converge on this solution.

The goal of intensification is to exploit any information about the fitness landscape in order to converge to good solutions as quickly as possible. This works well for relatively smooth fitness landscapes, such as the one illustrated in Fig. 6a, as globally optimal solutions can be identified in a computationally efficient manner. However, for rugged fitness landscapes (Fig. 6b), exploitative searching behaviour generally results in convergence to local optima that are in the vicinity of the starting position of the search, as there is insufficient exploration of other regions of the search space. Consequently, for rugged fitness landscapes, the performance of algorithms that exhibit exploitative behaviour can be highly variable, depending on the starting position in the search space.

4. Why should we use evolutionary algorithms?

There are a number of reasons why EAs are a good choice for finding optimal solutions to environmental problems, as discussed below.

EAs are able to find (near-) globally optimal solutions: A major advantage of EAs is that they are population-based. This is analogous to having an entire search party exploring the fitness landscape(s) on the lookout for the globally optimal solution(s), rather than a single person. As a result, large and rugged search spaces can be explored more effectively and efficiently because the use of a search party, rather than a single person, enables a greater area of the search space to be covered. In addition, the members of the search party often exchange and share information with each other using a range of mechanisms (see Sections 5 and 6), enabling promising regions of the search space to be identified more effectively and subsequently enabling the search to be concentrated in these regions.

The searching behaviour of EAs can be customised to the problem under consideration: EAs have the ability to be tuned to determine Δx_{t-1} values (Equation (4)) that represent an appropriate balance between exploration (diversification) and exploitation (intensification), depending on the properties of the fitness landscape (i.e. the problem under consideration). This enables EAs to escape local optima in the fitness landscape, while still being able to converge relatively quickly (to global, or near-global, optima). In contrast, most conventional optimization algorithms do not have this ability.

For example, gradient methods, which are a classical optimization strategy, calculate Δx_{t-1} as a function of the local gradient of the current solution (x_{t-1}) in the fitness landscape and a step size (Fig. 7a). Consequently, gradient-based methods exhibit a high degree of exploitation (intensification) and therefore only search locally in certain regions of

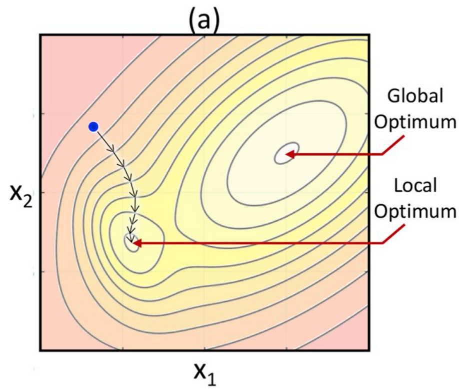


Fig. 7. Examples of (a) search using a gradient-based approach, and (b) random (exhaustive) search that "blindly" samples the search space. Here, the response surface of Fig. 1 is shown as contour plots.

the fitness landscape, but they do not have the ability to escape local optima. In contrast, as part of traditionally used random search strategies, Δx_{t-1} is chosen randomly by sampling the feasible space, as shown in Fig. 7b. Consequently, this strategy exhibits a high degree of exploration (diversification), enabling larger areas of the fitness landscape to be searched, but does not have the ability to converge to good solutions.

The fact that these more traditional optimization algorithms exhibit fixed searching behaviours that cannot be tailored to match the characteristics of the fitness landscape limits their range of applicability, as they are generally only suited to certain types of problems. For example, gradient-based methods are generally only suited to problems with relatively smooth fitness landscapes with a single basin of attraction (see Fig. 6a), whereas random search methods are generally only suited to problems with very rough and unstructured search spaces (see Fig. 6b). However, as mentioned previously, the searching behaviour of EAs can be adjusted to achieve the most appropriate balance between diversification and intensification for the problem under consideration.

EAs are easily linked with (existing) simulation models: One of the greatest advantages of using EAs is that they can be easily linked with existing simulation and integrated assessment models (see Section 8). This can be achieved in a straightforward manner, only requiring a two-way coupling between the optimization algorithm and an (existing) simulation model (Fig. 8). In the coupling, (i) the optimization algorithm determines decision variable values that are passed to the simulation model, (ii) the simulation model evaluates the corresponding objective function and constraint values and (iii) the objective function and constraint values are passed back to the optimization algorithm. Consequently, if the performance of a system can be simulated using an existing model, it can also be optimized using EAs. This facilitates the comparison of solutions created via informal optimization (using the simulation model as an engine) with the solutions resulting from the EA optimization process, which can aid in users' trust of the optimization results (Smith et al., 2015).

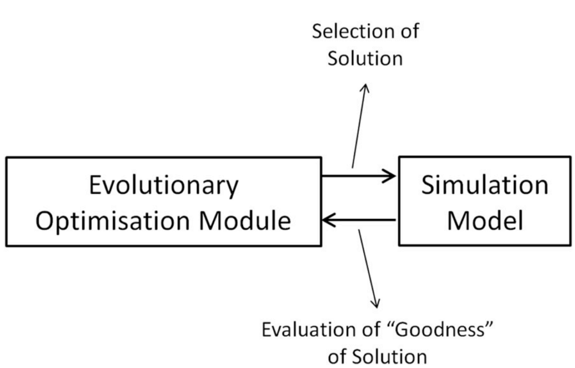


Fig. 8. Illustration of coupling between the evolutionary optimization module and the simulation model, where the optimization module can be "bolted onto" any simulation model. The optimization module identifies which solutions to try and passes these to the simulation model, which evaluates the utility of these solutions (i.e., objective function and constraints). This information is passed back to the optimization module, where it is used to determine which solutions to try next.

An implication of this is that EAs have a wider range of applicability compared with many more traditional optimization approaches, such as linear programming (Md. Azamathulla et al., 2008) and gradient-based methods (Kessler and Shamir, 1989). For example:

- Linear programming works when the objective function and constraints are linear, while many environmental processes are highly non-linear. Therefore, when linear programming is used, the outputs from simulation models can generally not be used directly, and instead, a simplified (linear approximation) version of the problem has to be solved. In this way, although linear programming is very computationally efficient and guaranteed to find the globally optimal solution, the solution that is found is generally not to the actual problem (unless the problem is linear), but a simplified version of the problem.
- When gradient methods are used, information about the gradient of the fitness function is used to decide in which direction the search should proceed (i.e., how $\Delta x_{t-}1$ is determined), as mentioned previously. However, the gradient (derivative) information of environmental models is not typically available, requiring the derivative of the objective function to be approximated numerically. This increases the computational demand, as many more objective function values (i.e., model runs) need to be evaluated. In addition, in fitness landscapes with significant roughness/noise (small-scale features that might be due to simulation model errors), the resulting derivative information might be deceptive and misguide the search. Related to this, the objective function might be non-differentiable (e.g. due to discontinuity) in some areas of the feasible space. The derivative-free nature of EAs, however, makes them insensitivet to possible discontinuity and ruggedness in fitness landscapes.

The way EAs are used is intuitive: The application of EAs is intuitive, as the optimization process that is used mirrors the informal optimization process adopted when using simulation models to assist with the identification of solutions to environmental problems (Fig. 2). The major difference between the informal optimization process in Fig. 2 and the formal EA-based optimization process is in how solutions are altered (i.e., how Δx_{t-1} is obtained). As part of informal optimization processes, values of Δx_{t-1} are obtained using personal domain knowledge, experience and intuition, but in the EA-based process, these

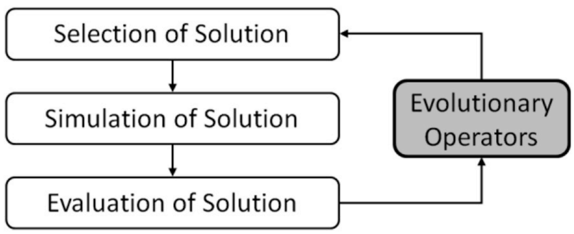


Fig. 9. Process of identifying optimal solutions when evolutionary algorithms are used.

changes in decision variable values from one iteration to the next are obtained automatically with the aid of evolutionary operators (Fig. 9). The operators are generally based on optimization strategies found in nature, such as survival of the fittest, although this is not always the case (see Sections 5 and 6 for details).

EAs are applicable to discrete and continuous decision variables: Another advantage of EAs is that they can generally work with both discrete and continuous decision variables, as opposed to most traditional optimization methods, which require continuous decision variables (exceptions include integer programming (Samani and Mottaghi, 2006) and combinatorial optimization methods (Da Conceição Cunha and Sousa, 1999)). This is of benefit for a large number of real-world applications where decision variables are either categorical (e.g. distinct crop choices, distinct infrastructure options) or where continuous variables are constrained by practical considerations (e.g. pipes are only manufactured in discrete diameters, wasteload reduction levels are aligned with particular treatment processes). However, while EAs can cater to both discrete and continuous variables, the mechanisms that aim to improve solution quality from one iteration to the next are different for discrete and continuous variables, requiring these mechanisms to be implemented in a single algorithm, as is done for example in the Exploratory Modelling Workbench (Kwakkel, 2017), the Water System Multi Objective Genetic Algorithm (WSMGA) (https://github.com/jeffrey-newman/WSMGA-with-Wrapper-and-Analytics; Wu et al., 2010b) and the mixed integer implementation of the Pareto Archived Dynamically Dimensioned Search algorithm applied in Tolson et al. (2012) (http://www.civil.uwaterloo.ca/btolson/software.aspx).

EAs can handle constraints in a straightforward manner: Constraint handling in EAs is easy and straightforward. The most common approach to handling constraints is to use penalty functions. The idea is to transform a constrained optimization problem into an unconstrained problem by adding a penalty value to the objective function based on the amount of constraint violation of a candidate solution (see Section 5.3). As reviewed by Coello (2002), there are other constraint handling techniques that work well, such as restricted tournament selection for multi-objective optimization, in which the feasibility of solutions is integrated into the selection procedure (see Section 5.3 for details, especially Fig. 14).

EAs have the ability to deal with multiple objectives simultaneously: Many traditional approaches to solving multi-objective optimization problems merge the different fitness landscapes into one, typically through their weighted summation based on their perceived importance, thereby turning a multi-objective optimization problem into a single objective problem. Such approaches, however, limit any insights that can be gained into the problem and its solutions, as trade-offs between objectives are not able to be explored. EAs, however, are able to overcome this limitation, as they typically have multi-objective analogues. Such analogues navigate multiple fitness landscapes simultaneously during the optimization process and are able to approximate Pareto fronts in a single algorithm run, as explained in Section 3.2.

EAs are easily parallelizable: EAs are naturally suited to being implemented in parallel computing environments. At each generation, individual solutions in the population can be evaluated in parallel on multiple processors to accelerate the search. This can result in a significant saving in (computational) time, compared with most traditional optimization methods, where candidate solutions would have to be evaluated serially during the search.

EAs are not prescriptive: As EAs work with populations of solutions, they produce a number of near-optimal solutions, which might be similar in objective function space, but quite different in decision variable space (either for single- or multi-objective problems). This enables consideration of factors other than those captured in the mathematical formulation of the optimization problem when selecting the final "optimal" solution. As a result, decision-makers have greater control in terms of using their judgement and intuition to select the final solution based on a number of good solutions "suggested" by the optimization algorithm (Di Matteo et al., 2017b). In this way, EAs are used to assist with "sifting through" the very large solution spaces that are a feature of environmental problems (see Table 1) in order to identify a set of near-optimal candidate solutions that can then be scrutinized by decision-makers to identify those that make most sense, either informally or with the aid of multi-criteria decision analysis methods (e.g. Hyde and Maier, 2006).

EAs result in increased trust in optimization results: The fact that the evaluation of objectives and constraints when EAs are used is based on the outputs of (existing) environmental simulation models, and that the iterative process of identifying improved solutions mimics that used by experienced environmental modellers, reduces the black-box nature of the optimization process and increases trust in the optimal solutions identified. This is in contrast to the use of more traditional optimization methods, such as linear programming, where simplified versions of existing models are likely to be used, the application of which is unlikely to evoke the same level of trust. The fact that solutions are not "prescriptive", as outlined above, is also likely to increase trust in the overall optimization process. However, the degree to which the solutions developed with the aid of simulation-optimization approaches are trusted is likely to be a function of the degree to which decision-makers are familiar with and trust the simulation model, especially if the optimal solutions do not correspond with expectations (Di Matteo et al., 2018).

5. How do single objective EAs and other metaheuristics work?

As mentioned previously, EAs are similar to many conventional optimization algorithms in that they update decision variable values in an iterative fashion (Equation (4)) in order to identify solutions that optimize the objective function. However, as shown in the high-level pseudo-code for single-objective EAs below, they work with populations of solutions, rather than a single solution, as discussed in Section 4, and update decision variable values using heuristic operators. These operators can vary significantly between different types of EAs, but are usually inspired by examples from nature. In general, there are two major philosophies for determining Δx_{t-1}, "evolutionary computation" and "swarm intelligence". The former, such as Genetic Algorithms (GAs) (Goldberg, 1989), was inspired by biological evolution, such as selection, recombination (crossover), and mutation, while the latter, such as Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995) and Ant Colony Optimization (ACO) (Dorigo et al., 1996), was inspired by the collective social behaviour of natural organisms such as birds, ants, and fish.

A range of "operators" has been developed over the past decades to mimic the functioning of these natural systems for optimization purposes. An EA utilizes a collection of such operators that work together to determine Δx_{t-1} intelligently based on feedback from the performance of the solutions generated in the previous iteration (Fig. 9), while keeping "some" balance between diversification and intensification. The relative degree of exploration and exploitation of the search is adjusted by changing the values of a number of parameters that control

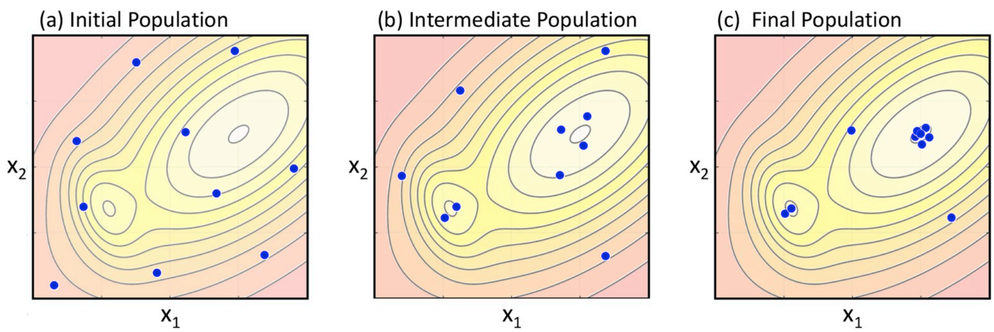


Fig. 10. Example performance of an evolutionary algorithm. Plot (a) shows an initial population of solutions randomly distributed in the decision variable space (population size = 10). Plot (b) shows the "evolving" population after a limited number of generations where both regions of attraction have been identified. Plot (c) shows the final population that has converged to the global optimum.

the function of theses operators, which can be done in a variety of ways (see e.g. Zheng et al., 2017).

A high-level pseudo-code for single-objective EAs

- Generate an initial population of solutions, $\mathbf{x}_0$, and evaluate their fitness
- REPEAT
  - Generate a new population, $\mathbf{x}_t$, by updating $\mathbf{x}_{t-1}$ using heuristic operators (e.g. evolution operators including selection, recombination, and mutation that focus on the best solutions in the set so far)
  - Evaluate the fitness of solutions in the new population by using the simulation model to evaluate the fitness function
- UNTIL stopping criteria are met

In general, as shown in Fig. 10 for an example with a 2-dimensional fitness landscape, EAs begin with a randomly distributed initial population (Fig. 10a) and an exploration-oriented search in the first iterations to locate the main regions of attraction (Fig. 10b). As the search continues, it becomes more exploitation-oriented in the regions of attraction, and identifies the best solution in the final iteration (Fig. 10c). Examples of how this is achieved using evolutionary computation- and swarm intelligence-based algorithms are given in Sections 5.1 and 5.2, respectively.

5.1. Genetic algorithms

Fig. 11 shows how a set of operators works consecutively to determine Δx_{t-1} in Genetic Algorithms (GAs). Individual solutions (sets of decision variables) of a population and their individual decision variables are analogous to chromosomes (or genotypes) and genes, respectively. To update the current population, the algorithm utilizes some "selection" operator to repeatedly select parent chromosomes (a pair of individual solutions) from the current population, typically based on their fitness function. The selected parent chromosomes are then used for breeding via some "crossover" and "mutation" operators, and the resulting pair of offspring are used to form the next-generation population.

A selection operator is generally based on the "survival of the fittest" concept from Darwinian evolution theory and picks probabilistically the fittest candidate solutions (best in terms of objective function) in the current population for breeding. An example selection operator is called "roulette wheel", named after a casino game for

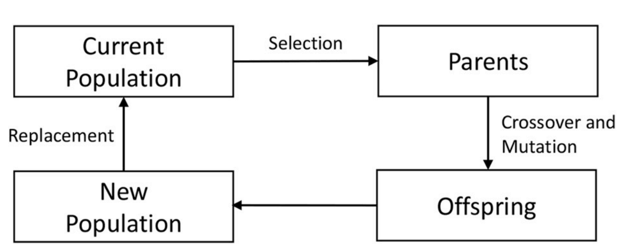


Fig. 11. Genetic heuristic operators for population updating in EAs.

gambling, where the fitness level is used to associate a probability of selection with each candidate solution of the population. Alternatively, fitter candidate solutions can be identified by comparing the fitness of a subset of solutions (usually pairs) in "tournaments", with the winner of the tournament selected for reproduction. Selection can also directly carry over the best solution(s) in a population to the next generation. This mechanism is called "elitism" and guarantees that solution quality will not degrade during optimization. Overall, selection promotes intensification during search.

When two parent solutions are selected, the crossover operator is applied to reproduce two offspring solutions. Fig. 12a shows two parent solutions (chromosomes) i and j of a 5-decision variable problem (5 genes) where a "single-point crossover" is applied. In this crossover, a single point on the chromosomes is selected randomly, and all genes (decision variable values) beyond that point are swapped between the two parents to reproduce the two offspring. Fig. 12c illustrates how a single-point crossover operator functions with 2 decision variables on the fitness landscape of Fig. 10. Crossover operators may work for both intensification and diversification during the search, depending on the type of crossover operator and the locations of parents with respect to each other.

A mutation operator may be applied to mutate a small number of offspring that resulted from the crossover. Fig. 12b shows an example mutation where one of the decision variables (genes) is randomly selected and replaced by a uniformly distributed random number in its feasible range. Fig. 12d illustrates how this mutation works in the 2-decision variable space where x_1 is mutated. Mutation works to preserve and introduce diversity during the search. It enables the EA to escape local optima.

The heuristic operators above can be easily applied to both continuous and discrete problems. In case of continuous problems, such as model calibration, the procedure explained above directly applies to model parameters that can be varied within their feasible ranges. In case of discrete problems, such as water distribution system design where the optimal size of pipes is of interest, the feasible range of each decision variable is reduced to a set of possible values (e.g. commercially available pipe sizes) that can be directly incorporated into the procedure. For example, for mutation of Fig. 12b, the pipe size can be directly replaced by another pipe size chosen randomly from the available set of pipe sizes. Further details on GAs can be found in Holland (1975), Goldberg (1989), Michalewicz (1992) Coley (1999), Nicklow et al. (2009) and Maier et al. (2014).

5.2. Ant colony optimization algorithms

In contrast to GAs, in ACO algorithms (ACOAs), the evolutionary operators are not applied to solutions directly. Instead, the solution space is modified to encourage the selection of decision variable values that have resulted in good solutions in previous iterations. As a result, completely new solutions are generated in each iteration, rather than modifying solutions from previous iterations.

This searching behaviour is inspired by the way colonies of ants look

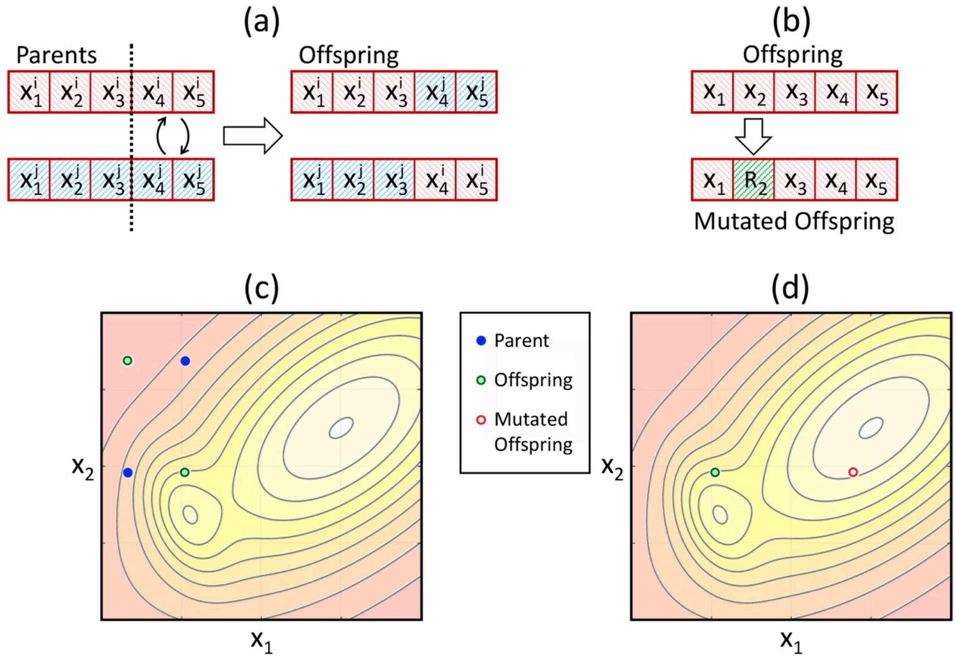


Fig. 12. Example crossover and mutation operators and their performance on a fitness landscape. (a) Single-point crossover that reproduces two offspring from two parents in a 5D decision variable space. (b) Uniform mutation that replaces the value of a decision variable by a uniformly distributed random number between the variable lower and upper bounds (R_2 = random number in [x_{2l}, x_{2u}]). Example outcome of (c) single-point crossover and (d) uniform mutation in a 2D decision variable space.

for food in nature. As part of this process, ants deposit pheromone, leaving trails, and other ants are more likely to follow trails with higher concentrations of pheromone. As shorter paths between a colony's nest and a food source can be traversed more quickly, shorter paths receive more pheromone per unit time, encouraging more ants to follow this path, further increasing its pheromone concentration. In this way, a positive feedback loop is created, reinforcing shorter paths (i.e. better solutions) from one iteration to the next. At the same time, the attractiveness of longer paths diminishes over time due to pheromone evaporation.

When using ACO to identify optimal solutions to environmental problems, the values particular decision variables can take correspond to the different paths ants can traverse (e.g. Golding et al., 2017; Nguyen et al., 2016; Szemis et al., 2012). As these paths are discrete, the decision variable options also have to be discrete, which requires continuous decision variables to be discretized using an appropriate resolution. In the example in Fig. 13, there are two decision variables, X_1 and X_2, each of which can take on nine discrete values (i.e. $X_{1,1}$, $X_{1,2}$, …$X_{1,9}$; $X_{2,1}$, $X_{2,2}$, …$X_{2,9}$). At the beginning of the optimization process, the same pheromone concentrations are assigned to all decision variable options, as indicated by the same thickness of blue shading for each of the nine paths for both decision variables in Fig. 13a. As the optimization process progresses, decision variable options (i.e. paths) that result in better solutions (i.e. better objective function values) receive more pheromone, increasing their chances of being selected in subsequent iterations, creating the positive reinforcement loop observed in nature. Consequently, at the end of the optimization process, the paths that result in the globally optimal solution have significantly larger pheromone concentrations than the other paths, as shown in Fig. 13b i.e. the paths corresponding to decision variable options $X_{1,7}$ and $X_{2,6}$.

At each iteration of the optimization process, decision variable options (i.e. paths) are selected as a function of the relative pheromone concentrations on different paths. A common way of achieving this is by using a biased roulette wheel, although the way the probabilities of selecting a particular path are calculated generally varies between algorithms (Dorigo and Gambardella, 1997; Dorigo et al., 1996). For example, in many algorithms, these probabilities are often also a function of a preference for locally optimal solutions (i.e. decision variable values that are known to increase objective function values (for a maximization problem)), in addition to pheromone levels, with the relative contribution of these two mechanisms based on user-selected weightings. Once paths have been selected for each decision variable, a trial solution to the problem under consideration has been obtained. In general, a number of solutions is obtained in each iteration, which is akin to the size of the ant colony in natural systems and is equivalent to the population size in GAs.

The way pheromone levels on each of the paths are updated from one iteration to the next is generally a function of the quality of the solutions obtained, where paths that contribute to solutions with better objective function values receive a greater amount of pheromone than paths that contribute to poorer quality solutions. The way this is achieved varies between algorithms, where some algorithms use information from all ants in the colony, while others only use information from ants that select high-performing solutions (Bullnheimer et al., 1999; Dorigo et al., 1996; Stützle and Hoos, 2000). Further details on ACO algorithms and their applications can be found in Dorigo et al. (1996), Stützle and Hoos (2000), Dorigo et al. (2006), García-Martínez et al. (2007), Zecchin et al. (2007) and Afshar et al. (2015).

5.3. Constraint handling

The design of EAs and other metaheuristics is typically such that they can search for optimal solutions in a hypercube decision space, where there are only bound constraints (see Equation (3)). In some cases, the design of an algorithm functions naturally to respect bound constraints (e.g. binary decision variable problem solved with a GA). However, in many algorithms, some special constraint handling procedures, often invisible to users, are required to respect bound constraints. If during the search a generated solution goes beyond a bound, it may be rejected (and a new solution would be generated), absorbed to the bound, or reflected back into the hypercube. Fig. 14 shows an example with these three simple "constraint handling" mechanisms for bound constraints.

The ability to handle other types of constraints (see Equation (2)) may not be as straightforward. In general, the "rejecting" approach is the simplest but most inefficient way of handling any type of constraint. With this approach, any candidate solution that violates a constraint will be rejected and removed from the search and the degree of constraint violation is not considered. An efficient, and perhaps the most

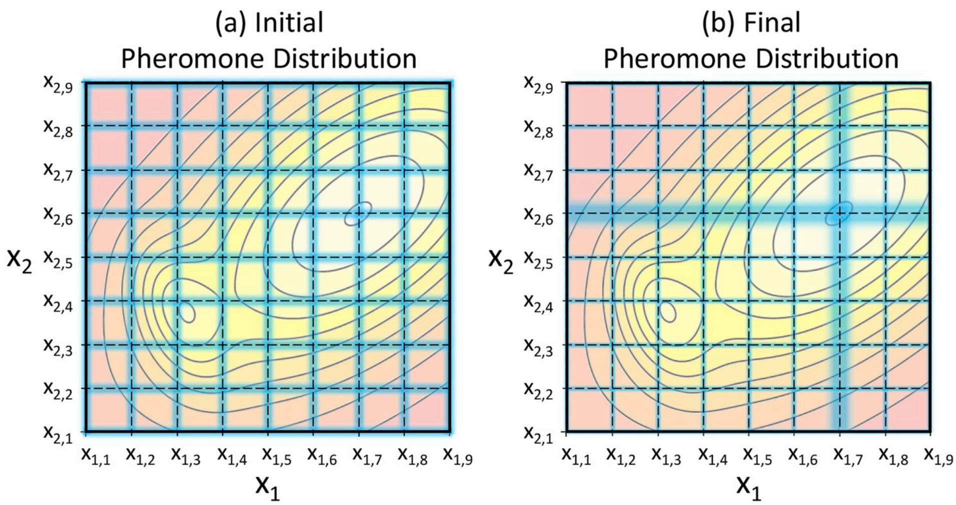


Fig. 13. Illustration of the evolution of pheromone concentrations over a large number of iterations, where the thickness of the blue shading of the paths corresponding to discrete decision variable choices corresponds to pheromone concentration. As can be seen, the discrete values of X_1nd X_2 that correspond to the globally optimal solution (i.e. $X_{1,7}$ and $X_{2,}6$) have higher pheromone concentrations during the final iteration, thereby increasing the probability that these decision variable values will be selected (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

popular, way of handling any type of constraint is the "penalizing" approach. In this general-purpose approach, which is also referred to as the "penalty function" approach, a penalty function is defined to quantify the degree of infeasibility (constraint violation) of an infeasible solution and penalizes it proportionally. In this way, the constraint is removed from the optimization formulation, and instead, a corresponding penalty function is integrated into to the objective function. This integration is typically achieved by weighted averaging with a carefully chosen weight for the penalty function. In practice, the user may need to identify the appropriate penalty function and corresponding weights via trial and error.

6. How do multi-objective EAs work?

The searching mechanisms of multi-objective EAs (MOEAs) are based on the same basic principles that underpin single objective EAs. However, as shown in the high-level pseudo-code below, a major difference between the two is in the way they evaluate the fitness of a solution. Unlike single-objective optimization, where fitness evaluation is straightforward and is performed simply by evaluating a single objective function, multi-objective optimization requires a more elaborate "fitness assignment" scheme that brings together and unifies the different objective functions. Fitness assignment is a major component of multi-objective EAs and other metaheuristics and basically refers to the process of translating a vector of objective function values into a scalar-valued fitness that can be used for ranking and selecting solutions of higher quality during the course of the search.

A high-level pseudo-code for multi-objective EAs

- Generate an initial population of solutions, $\mathbf{x}_0$, and assign their fitness using code *a*
- REPEAT
  - ○ Generate a new population, $\mathbf{x}_t$, by updating $\mathbf{x}_{t-1}$ using heuristic operators (e.g. evolution operators including selection, recombination, and mutation)
  - ○ Assign the fitness of solutions in the new population using code A below
- UNTIL stopping criteria are met
- Return the non-dominated solutions in the current population and respective fitness functions

A: Sub-pseudo-code for fitness assignment in multi-objective EAs
- ○ Evaluate the multiple fitness functions for each solution
- ○ Rank the solutions based on their dominance and density

There are different strategies for fitness assignment, but these are generally based on the dominance concept illustrated in Section 3. The idea is to rank candidate solutions based on their dominance strength in objective space and to use the rank numbers of the solutions as their respective fitness values. Fig. 15 shows three strategies for ranking the solutions in a population, namely "dominance depth", "dominance rank" and "dominance count". In the dominance depth strategy (e.g., used in NSGA-ll developed by Deb et al., 2000), the solutions in a population are grouped into multiple fronts. The non-dominated solutions form the first front receive rank 1, and the remaining solutions that are non-dominated by all solutions, excluding the solutions on the first front, form the second front and receive rank 2. This will continue to the third and higher fronts. In this strategy, candidate solutions lying on the same front are deemed equally fit. In the dominance rank strategy, the fitness of any given solution is the number of other solutions that dominate that solution (e.g., used in MOGA developed by Fonseca and Fleming, 1993); while in the dominance count strategy, fitness of a given solution is defined as the number of other solutions this solution dominates (e.g., used in SPEA2 developed by Ziztler et al., 2002). Readers may refer to Zitzler et al. (2004) for more details on the different fitness assignment strategies for multi-objective optimization.

In multi-objective EAs, diversification is not limited to the decision-variable space. It is also important to introduce diversity in the objective space (see Fig. 16) to diversify the optimal solutions and spread them uniformly on the Pareto front. This is because the purpose of the optimization process is not to identify a single, optimal solution, but to identify a set of optimal solutions that are distributed with a uniform "density" over the Pareto front. For this purpose, "crowding distance", which is a measure to estimate the density of solutions surrounding any given point in the objective space, is often used to encourage solutions to "spread out". Accurately stated, in a bi-objective space, the crowding distance of a solution is defined as one half of the circumference of the rectangle formed by the two neighbouring solutions of the same front on either side of that solution (Fig. 16). In addition, the three ranking strategies in Fig. 15 may result in multiple solutions receiving the same rank, while EAs may need to differentiate such solutions in the process of selection. For example, in the dominance depth strategy, all solutions lying on the same front would be seen equally. In this case, *density*, as quantified, for example, by crowding distance, can be used for a second-level fitness assignment of these solutions, indicating that solutions lying on the same front, but with larger crowding distance, would receive a higher fitness value (i.e., a higher chance to be selected).

Unlike the above strategies for MOEAs, some other algorithms take a more simplified approach and only search from dominance rank 1 (nondominated solutions). That is, the population or set of current solutions maintained during the search only contains those that are currently non-dominated. Two examples would be PAES and PADDS (Asadzadeh and Tolson, 2013; Asadzadeh and Tolson, 2009; Knowles and Corne, 2000). In both of these algorithms, a measure such as crowding distance is utilized to influence algorithm trajectory when preference between two or more non-dominated solutions is required.

It should be noted that the number of Pareto-optimal solutions

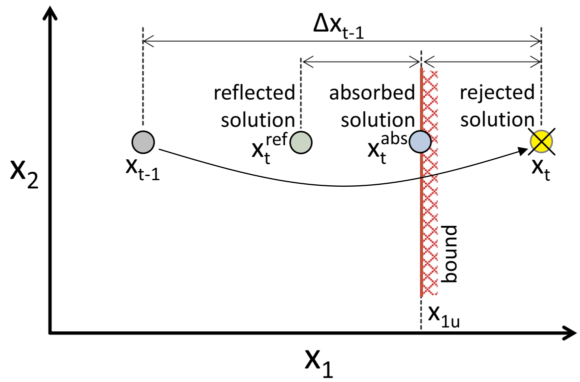


Fig. 14. Bound constraint handling in metaheuristic optimization. Three strategies are shown: 1- the new solution violating the constraint is rejected in order to generate another new solution, 2- the new solution is modified and absorbed to the bound, and 3- the new solution is modified and reflected by the bound.

increases with an increase in the number of objective functions. This is because all of the Pareto-optimal solutions of a *k*-objective optimization problem would remain Pareto optimal in the same problem with additional objective functions, while any additional objective function would introduce a new dimension in the objective space and extend the Pareto front. So as more objectives are added to a problem, the proportion of feasible solutions being Pareto-optimal also increases. This increase limits the ability of MOEAs (and any multi-objective algorithm) to accurately approximate high dimensional Pareto fronts and as such, the application of MOEAs is often limited to two or three objectives. Further details on MOEAs and their applications can be found in Coello et al. (2007), Deb (2001), Nicklow et al. (2009) and Reed et al. (2013).

7. What are the challenges associated with using EAs?

While EAs have a number of advantages over more conventional optimization methods, as discussed in Section 4, they also present a number of additional challenges, especially related to computational efficiency and the adjustment of their searching behaviour.

Challenges with computational efficiency: As EAs work with populations of solutions and generally evolve better solutions over tens or hundreds of iterations, the number of times objective function and constraint values have to be calculated is the product of the population size and the number of iterations. This can make their use computationally demanding, especially if objective function and/or constraint values are obtained with the aid of environmental models. The application of EAs to complex, real-world problems are of particular concern in this regard, as (i) model run-times can be in the order of minutes or even hours and (ii) search spaces can be extremely large and rugged, requiring larger population sizes and/or numbers of generations in order to find good solutions. These computational challenges are exacerbated when considering uncertainties, as performance metrics such as risk, resilience, reliability or robustness (Hashimoto et al., 1982; McPhail et al., 2018) generally require simulation models to be run hundreds or even thousands of times during each iteration of the optimization process (Beh et al., 2017).

These computational challenges can be addressed using a number of strategies. A commonly used approach is to replace computationally expensive simulation models with more efficient surrogate-, meta- or emulation-models (Razavi et al., 2012), which can increase the computational efficiency of EAs by one to two orders of magnitude (Beh et al., 2017; Broad et al., 2015b; Timani and Peralta, 2017). As EAs are easily parallelizable (see Section 4), the use of parallel computing can also be an effective means of reducing overall runtime (Newland et al., 2018; Tang et al., 2007). Depending on the type of problem and the EA algorithm used, "pre-emption" strategies can be used to opportunistically evade running computationally expensive models during optimization (Asadzadeh et al., 2014; Razavi et al., 2010). Alternatively, the size and complexity of the optimization problem can be decreased by reducing the size of the search space (Fu et al., 2012) or re-formulating the optimization problem (Garner and Keller, 2018; Quinn et al., 2017).

Challenges with adjustment of searching behaviour: As mentioned in Section 4, one of the advantages of EAs is that their searching behaviour (i.e. their degree of diversification (exploration) and intensification (exploitation)) can be adjusted to match the properties of the fitness landscape for the problem under consideration. However, in practice, this can be a challenge, as the properties of the fitness landscape are unknown. As a result, the parameters that control the searching behaviour of EAs are commonly determined with the aid of extensive sensitivity analysis, although guidelines for the selection of appropriate parameters have been developed for some problem types (e.g. Gibbs et al., 2015; Zecchin et al., 2005). Alternatively, algorithm searching behaviour can be adjusted dynamically throughout the search by automatically adjusting (i) parameter values to achieve pre-determined searching behaviours (Hadka and Reed, 2013; Vrugt et al., 2009; Wang et al., 2017; Zheng et al., 2017) or (ii) the utilisation rate of different operators in response to algorithm performance (Hadka and Reed, 2013; Vrugt et al., 2009; Wang et al., 2017; Zheng et al., 2017).

8. How do we implement EAs?

A typical implementation strategy for the workflow described by Fig. 8 is to view the search algorithm and simulation model as "black box" components that are coupled via an I/O (input/output) interface. This I/O interface manages run-time execution and data transfers of both the algorithm and the simulation model. As discussed in Section 4 (e.g. see Fig. 8), data transfer consists of preparing model input and processing model output and can be accomplished by reading and writing files or by in-memory or inter-process communication.

As summarized by Matott et al. (2009), a number of software packages have been built using the black box approach described above, and such packages typically support multiple algorithms and may be written in compiled (i.e., C/C+ + or FORTRAN) or interpreted (i.e., R, Python, or MATLAB) languages. Fig. 17 illustrates a black-box approach to implementing EAs. As indicated in Fig. 17, the primary operators of the optimizer (e.g. elitism, selection, crossover and mutation) are invariant with respect to the simulation model and linkage between the optimizer and simulation model is only required for evaluation of the fitness of the population of each generation.

In addition to standalone EA packages, there has been a variety of APIs (application programming interfaces) and libraries developed to support EA implementations (e.g. Hadka, 2016; Wilhelmstötter, 2018). These APIs and libraries generally prescribe a specific I/O interface that will ensure interchangeability and interoperability across algorithms implemented using a given API or library. In this way, libraries and APIs foster a community of practice with respect to EA implementation and their adoption is highly encouraged for advanced users.

The difficulty of comparing alternative algorithm implementations and of assessing incremental changes to a given algorithm is now being increasingly recognized (e.g. Hansen et al., 2016; Sörensen, 2013; Weyland, 2012, 2015). This has led some algorithm developers to formalize "standard" algorithm implementations that can be used to establish the expected baseline reference behaviour and performance of a given algorithm (Bratton and Kennedy, 2007; Swan et al., 2015).

The choice of programming language for implementing an EA algorithm is typically made pragmatically based on the historical preferences and experience of the code developer. In general, compiled languages are capable of delivering significantly better computational performance, relative to interpreted languages. However, these effects are not likely to be significant within a given algorithm implementation since runs of the simulation model usually dominate the overall

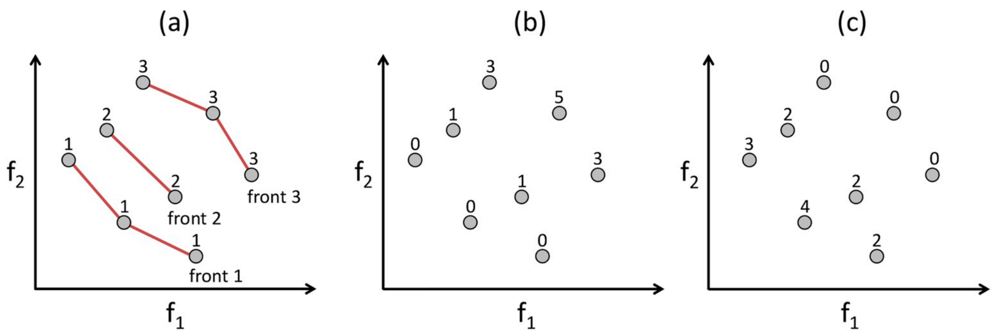


Fig. 15. Strategies for dominance-based ranking (fitness assignment) of candidate solutions in the objective space: (a) dominance depth, (b) dominance rank, and (c) dominance count. The number next to each candidate solution is the associated rank for that solution; in (a) and (b), a smaller number indicates a better solution, while in (c) a larger number indicates a better solution.

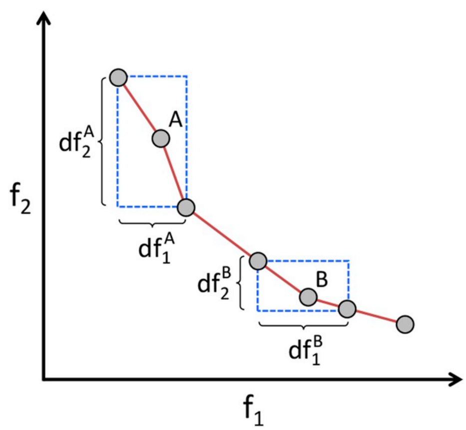


Fig. 16. Ranking of solutions lying on the same front by crowding distance. The crowding distances of solutions A and B are $df_1^A + df_2^A$ and $df_1^B + df_2^B$, respectively. As the crowing distance of A is larger than that of B, A will receive a better rank for selection.

compute requirements of a given simulation-based optimization exercise.

As mentioned previously, EAs that iteratively operate on a population of candidate solutions sets are readily parallelized using distributed computing, whereby population members are evaluated simultaneously by a team of processors. These processors may reside on the same motherboard (e.g. in a typical desktop system), within the same rack or cluster of a data centre, within multiple data centres that are geographically separated, or, most generally, across multiple interconnected computing devices (e.g. desktops, laptops, tablets, and smartphones). In addition to choosing an appropriate communications interface (e.g. the message passing interface (MPI)(Gibson and Meter, 2000; Gropp et al., 2014), the sockets protocol, network attached storage (NAS), etc.) for relaying information between these processors, implementers must decide whether the processors should operate synchronously or asynchronously. Fig. 18 illustrates the difference between the synchronous (left hand side of Fig. 18) and asynchronous (right hand side of Fig. 18) schemes in the context of evaluating the new generation of an EA using parallel computing.

As illustrated in Fig. 18, in the synchronous mode of operation, each processor is assigned a set of a priori compute tasks and these are typically keyed to the processor's unique id. For example, processor 0 might always be responsible for evaluating the first 10 population members, with processor 1 assigned the next 10 members, and so on. While this approach minimizes inter-processor communication overhead, it can result in significant bottlenecking if simulation run times are markedly heterogeneous across the population members. For example, such heterogeneity can occur if model pre-emption is employed (Razavi et al., 2010). Consequently, some processors may take longer than others to complete a given round of evaluations, thereby forcing those that finish early to wait.

As shown in the right hand side of Fig. 18, in the asynchronous mode of operation, one or more processors are designated as "managers" or "coordinators". These managers communicate asynchronously and on-demand with the remaining "worker" processors to assign compute tasks. This asynchronous approach entails more inter-processor communication overhead but has the benefit of minimizing the idle time of "worker" processors.

As the use of EAs enters into the state of practice, there has been increased interest in incorporating them into decision support systems and related integrated modelling frameworks (e.g. Babendreier and Castleton, 2005; Laniak et al., 2013). Such "meta-systems" involve complex modelling workflows that can include parameter estimation, uncertainty and sensitivity analysis, and design optimization. Ideally, these systems should be structured to accommodate varied programming languages, libraries and APIs. Furthermore, the execution management used within a given meta-system should be implemented in a manner that can exploit the inherent parallelism of EAs.

Looking to the future, various EA research directions and emerging technologies are already influencing their implementation in computer software. For example, the ubiquity of cloud computing has fostered the emergence of optimization as a service (OaaS) as a practical on-demand tool for decision makers to apply EAs to solve previously intractable problems (Broad et al., 2015a; Kurschl et al., 2014). Rapid and agile OaaS implementation is facilitated by portable and interpreted web-ready languages that support reflection and introspection (e.g. python, php, and javascript). Furthermore, researchers are in the early stages of exploring how EAs can best exploit exascale parallel computing technologies like accelerators and graphics processing units (GPUs) (Alonso, 2015; Cárdenas-Montes et al., 2016; Hadka and Reed, 2015). Finally, the development of hybridized EA algorithms is an ongoing line of research (e.g. Hadka and Reed, 2013; Lozano and García-Martínez, 2010) that is likely to further motivate the desire for increased interoperability among EA implementations.

9. Conclusion

Environmental simulation models are used extensively to support decisions. However, in many instances, the problems being addressed are complex, non-linear, relatively poorly understood and, probably most importantly, characterised by extremely large solution spaces. This makes it almost impossible to identify a set of solutions that provides the best trade-offs between competing objectives, such as minimizing cost and maximizing environmental outcomes, using an informal "optimization" approach, as part of which the best solution is identified with the aid of domain knowledge, experience and intuition, coupled with the outputs from one or more environmental simulation models.

Evolutionary algorithms and other metaheuristics provide a means of addressing the above shortcoming, as they can be linked directly with the environmental simulation models used as part of the commonly used informal optimization approach outlined above, while allowing large search spaces to be explored in a computationally efficient

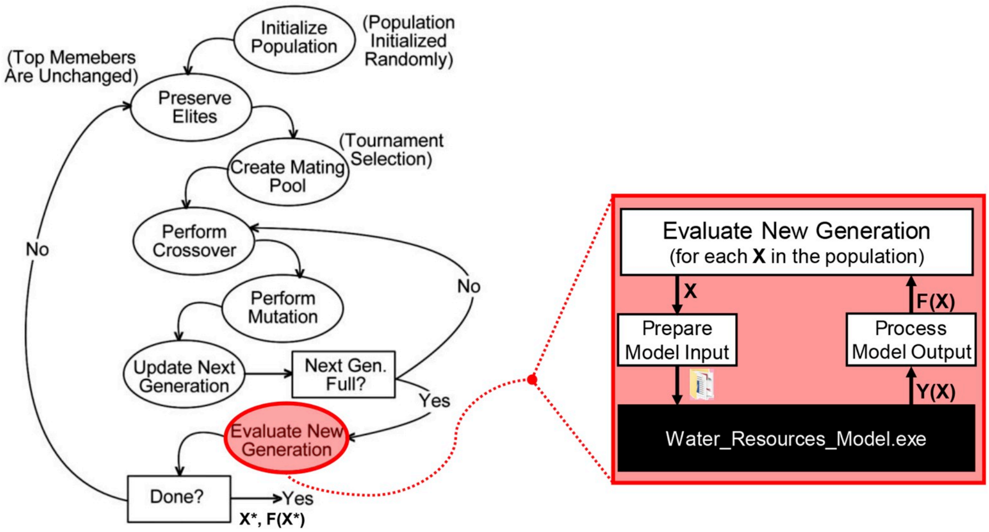


Fig. 17. Illustration of the 'black-box' approach to linking an evolutionary algorithm with a water resources model. x and f(x) are as described in equations (1)–(3) (see Section 3.1) and y(x) represents a set of model outputs (response variables) that are computed as a function of the decision variables (i.e. x). F(X) represents a vector of objective functions and/or constraints that are to be optimized and which are computed using the model outputs.

manner. This is achieved using the same process employed during informal optimization in that a trial solution is proposed, which is then evaluated using a simulation model leading to the selection of a new (hopefully improved) solution and so on. The primary difference is that the decision of which solution to try in the next iteration is not informed by the domain knowledge of the user, but by operators inspired by examples from nature, such as survival of the fittest and the way colonies of ants search for food. This generally enables the identification of solutions that achieve better environmental performance for the same cost or solutions that achieve a particular environmental outcome for reduced financial outlay, rather than solutions that result in inferior environmental outcomes for higher costs. In this way, formal optimization processes can be incorporated into integrated environmental decision-making processes, rather than prescribing a single optimal solution using a black-box process.

However, it is important to note that despite all of the advantages of EAs, they are not necessarily always the best optimization method to use. For example, if in some specific case where the analysed real-life problem could be linearized without losing its key features, and/or a planner needs an answer very quickly (e.g. hydropower scheduling), other optimization methods (such as Linear Programming) may be better suited. This highlights the importance of using any optimization method appropriately. EAs are particularly well suited to solving large, complex, discrete and non-linear real-life problems and are particularly useful for providing decision support in suitably formulated and solved optimization problems accompanied with relevant expert analyses, ideally in some iterative exploratory-type setup. When used in this way, EAs can be very helpful and can reduce the computational burden of the analysed problem, thereby enabling simulation modellers to explore many more options and trade-offs than would otherwise be possible. Thus they allow modellers to focus in general on more important aspects of the analysed problem.

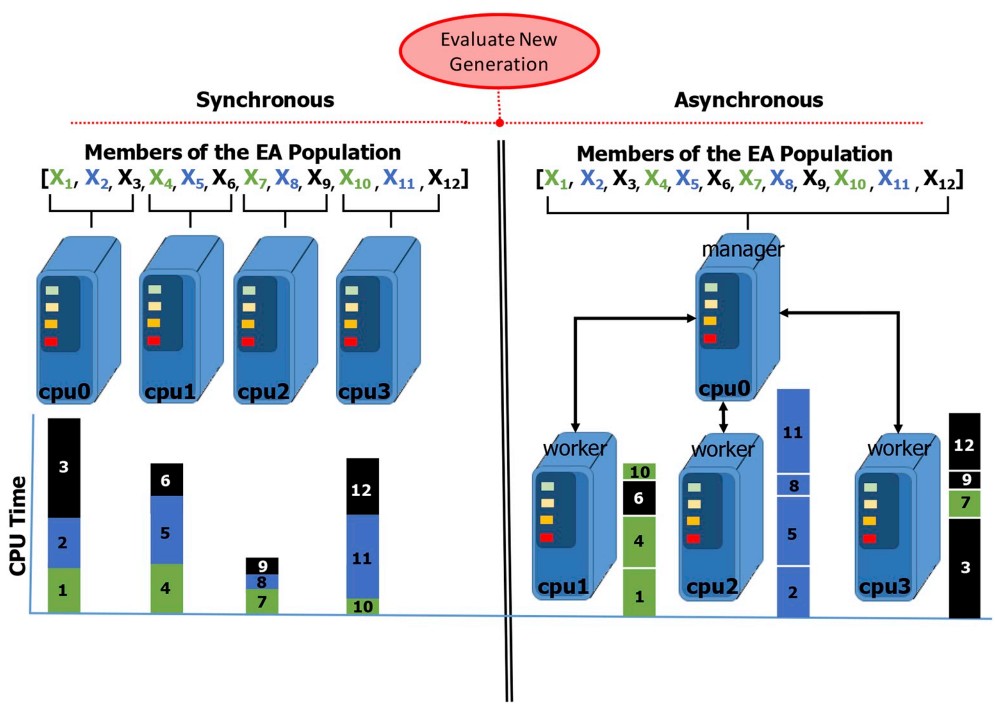


Fig. 18. Schematic comparison of synchronous vs. asynchronous approaches to EA parallelization. The hypothetical CPU timings illustrate how different evaluations can get assigned to different processors. Gaps in the asynchronous CPU time graphs represent overhead from interprocess communication.

Acknowledgements

We would like to acknowledge the following graduate students who have helped to proofread this manuscript, including Mikaela DeRousseau at the University of Colorado Boulder and Kasra Keshavarz and Mustakim Ali at the University of Saskatchewan. The authors would also like to thank David Vowels from the University of Adelaide and Tony Jakeman from the Australian National University for reading the draft manuscript and their useful comments for improving it, as well as Gayani Fernando from the University of Adelaide for assisting with compiling and formatting the references. Finally, the authors would like to thank the reviewers of this paper, Andrea Rizzoli, Jan Kwakkel and one anonymous reviewer, whose feedback have helped with improving the quality of this paper significantly.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.envsoft.2018.11.018.

References

Afshar, A., Massoumi, F., Afshar, A., Mariño, M.A., 2015. State of the art review of ant colony optimization applications in water resource management. Water Resour. Manag. 29 (11), 3891–3904.

Alavalapati, J.R.R., Adamowicz, W.L., 2000. Tourism impact modeling for resource extraction regions. Ann. Tourism Res. 27 (1), 188–202. https://doi.org/10.1016/S0160-7383(99)00064-X.

Alonso, J.J., 2015. Opportunities and Challenges for Multi-Disciplinary Analysis and Optimization at Exascale (Invited). In: 22nd AIAA Computational Fluid Dynamics Conference. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2015-3052.

Amato, F., Tonini, M., Murgante, B., Kanevski, M., 2018. Fuzzy definition of rural urban interface: an application based on land use change scenarios in Portugal. Environ. Model. Software 104, 171–187.

Ambrose, B., B.Avant, B., Han, Y., Knightes, C., Wool, T., 2017. Water Quality Assessment Simulation Program (WASP8): Upgrades to the Advanced Toxicant Module for Simulating Dissolved Chemicals, Nanomaterials, and Solids. U.S. Environmental Protection Agency, Washington, DC EPA/600/R-17/326.

Arnold, J.G., Srinivasan, R., Muttiah, R.S., Williams, J.R., 1998. Large area hydrologic modeling and assessment part I: Model development. J. Am. Water Resour. Assoc. 34 (1), 73–89.

Asadzadeh, M., Tolson, B.A., 2009. A new multi-objective algorithm, pareto archived DDS. In: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, Montreal, Qubec, Canada, . https://dl.acm.org/citation.cfm?doid=1570256.1570259.

Asadzadeh, M., Tolson, B., 2013. Pareto archived dynamically dimensioned search with hypervolume-based selection for multi-objective optimization. Eng. Optim. 45 (12), 1489–1509. https://doi.org/10.1080/0305215X.2012.748046.

Asadzadeh, M., Razavi, S., Tolson, B.A., Fay, D., 2014. Pre-emption strategies for efficient multi-objective optimization: Application to the development of Lake Superior regulation plan. Environ. Model. Software 54, 128–141. https://doi.org/10.1016/j.envsoft.2014.01.005.

Md Azamathulla, H., Wu, F.-C., Ghani, A.A., Narulkar, S.M., Zakaria, N.A., Chang, C.K., 2008. Comparison between genetic algorithm and linear programming approach for real time operation. Journal of Hydro-environment Research 2 (3), 172–181. https://doi.org/10.1016/j.jher.2008.10.001.

Babendreier, J.E., Castleton, K.J., 2005. Investigating uncertainty and sensitivity in integrated, multimedia environmental models: tools for FRAMES-3MRA. Environ. Model. Software 20 (8), 1043–1055. https://doi.org/10.1016/j.envsoft.2004.09.013.

Baró, F., Chaparro, L., Gómez-Baggethun, E., Langemeyer, J., Nowak, D.J., Terradas, J., 2014. Contribution of ecosystem services to air quality and climate change mitigation policies: the case of urban forests in Barcelona, Spain. AMBIO 43 (4), 466–479. https://doi.org/10.1007/s13280-014-0507-x.

Basher, R., 2006. Global early warning systems for natural hazards: systematic and people-centred. Phil. Trans. Roy. Soc. Lond.: Mathematical, Physical and Engineering Sciences 364 (1845), 2167–2182.

Basupi, I., Kapelan, Z., Butler, D., 2013. Reducing life-cycle carbon footprints in the redesign of water distribution systems. Journal of Water and Climate Change 4 (3), 176–192.

Beh, E.H.Y., Zheng, F., Dandy, G.C., Maier, H.R., Kapelan, Z., 2017. Robust optimization of water infrastructure planning under deep uncertainty using metamodels. Environ. Model. Software 93, 92–105. https://doi.org/10.1016/j.envsoft.2017.03.013.

Bennett, N.D., Croke, B.F., Guariso, G., Guillaume, J.H., Hamilton, S.H., Jakeman, A.J., Marsili-Libelli, S., Newham, L.T., Norton, J.P., Perrin, C., 2013. Characterising performance of environmental models. Environ. Model. Software 40, 1–20.

Blum, C., Roli, A., 2003. Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput. Surv. 35 (3), 268–308. https://doi.org/10.1145/937503.937505.

Blum, C., Puchinger, J., Raidl, G.R., Roli, A., 2011. Hybrid metaheuristics in combinatorial optimization: A survey. Appl. Soft Comput. 11 (6), 4135–4151. https://doi.org/10.1016/j.asoc.2011.02.032.

Borge, R., Lumbreras, J., Pérez, J., de la Paz, D., Vedrenne, M., de Andrés, J.M., Rodríguez, M.E., 2014. Emission inventories and modeling requirements for the development of air quality plans. Application to Madrid (Spain). Sci. Total Environ. 466, 809–819.

Boussaïd, I., Lepagnot, J., Siarry, P., 2013. A survey on optimization metaheuristics. Inf. Sci. 237, 82–117. https://doi.org/10.1016/j.ins.2013.02.041.

Bratton, D., Kennedy, J. (Eds.), 2007. Defining a standard for particle swarm optimization, In Swarm Intelligence Symposium, 2007. SIS 2007. IEEE.

Broad, D., McIver, D., Duncker, A., Dandy, G., McMichael, S. (Eds.), 2015. Incorporating Hydraulic Criteria into a Water Mass Balance Model for Optimal Operations, In Watermatex, 9th IWA Symposium.

Broad, D.R., Dandy, G.C., Maier, H.R., 2015b. A systematic approach to determining metamodel scope for risk-based optimization and its application to water distribution system design. Environ. Model. Software 69, 382–395. https://doi.org/10.1016/j.envsoft.2014.11.015.

Bullnheimer, B., Hartl, R.F., Strauss, C., 1999. A new rank based version of the Ant System. A computational study. Central European Journal for Operations Research and Economics 7 (1), 25–38.

Burn, D.H., Yulianti, J.S., 2001. Waste-load allocation using genetic algorithms. J. Water Resour. Plann. Manag. 127 (2), 121–129.

Cárdenas-Montes, M., Vega-Rodríguez Miguel, A., Rodríguez-Vázquez Juan, J., Gómez-Iglesias, A., 2016. Parallel evaluation of nonseparable functions by evolutionary algorithms on GPU. Concurrency Comput. Pract. Ex. 29 (5), e3949. https://doi.org/10.1002/cpe.3949.

Cho, J.H., Sung, K.S., Ha, S.R., 2004. A river water quality management model for optimising regional wastewater treatment using a genetic algorithm. J. Environ. Manag. 73 (3), 229–242.

Coello, C.A.C., 2002. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. Comput. Methods Appl. Mech. Eng. 191 (11–12), 1245–1287. https://doi.org/10.1016/S0045-7825(01)00323-1.

Coello, C.A.C., Lamont, G.B., Van Veldhuizen, D.A., 2007. Evolutionary algorithms for solving multi-objective problems, 2 ed. Springer.

Cohon, J.L., Marks, D.H., 1975. A review and evaluation of multiobjective programing techniques. Water Resour. Res. 11 (2), 208–220. https://doi.org/10.1029/WR011i002p00208.

Coley, D.A., 1999. An introduction to genetic algorithms for scientists and engineers. World Scientific Publishing Company, Singapore.

Côte, C.M., Moran, C.J., Hedemann, C.J., Koch, C., 2010. Systems modelling for effective mine water management. Environ. Model. Software 25 (12), 1664–1671. https://doi.org/10.1016/j.envsoft.2010.06.012.

Da Conceição Cunha, M., Sousa, J., 1999. Water distribution network design optimization: Simulated annealing approach. J. Water Resour. Plann. Manag. 125 (4), 215–221 doi: 10.1061/(ASCE)0733-9496(1999)125:4(215).

Deb, K., 2001. Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley, New York.

Deb, K., Agrawal, S., Pratap, A., Meyarivan, T. (Eds.), 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, In International Conference on Parallel Problem Solving From Nature.

Di Matteo, M., Dandy, G.C., Maier, H.R., 2017a. A multi-stakeholder portfolio optimization framework applied to stormwater best management practice (BMP) selection. Environ. Model. Software 97, 16–31. https://doi.org/10.1016/j.envsoft.2017.07.012.

Di Matteo, M., Dandy, G.C., Maier, H.R., 2017b. A multi-stakeholder portfolio optimization framework applied to stormwater best management practice (BMP) selection. Environ. Model. Software 97, 16–31. https://doi.org/10.1016/j.envsoft.2017.07.012.

Di Matteo, M., Dandy, G.C., Maier, H.R., 2018. Many-objective portfolio optimization approach for stormwater management project selection encouraging decision maker buy-in. Environ. Model. Software (in press).

Dorigo, M., Gambardella, L.M., 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput. 1 (1), 53–66. https://doi.org/10.1109/4235.585892.

Dorigo, M., Maniezzo, V., Colorni, A., 1996. Ant system: Optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. B Cybern. 26 (1), 29–41. https://doi.org/10.1109/3477.484436.

Dorigo, M., Birattari, M., Stutzle, T., 2006. Ant colony optimization. IEEE Comput. Intell. Mag. 1 (4), 28–39. https://doi.org/10.1109/MCI.2006.329691.

Dréo, J., Siarry, P., Pétrowski, A., Taillard, E., 2006. Metaheuristics for hard optimization: Methods and case studie. Springer, Berlin.

Eker, S., Kwakkel, J.H., 2018. Including robustness considerations in the search phase of many-objective robust decision making. Environ. Model. Software 105, 201–216.

Fiori, S.M., Zalba, S.M., 2003. Potential impacts of petroleum exploration and exploitation on biodiversity in a Patagonian nature reserve, Argentina. Biodivers. Conserv. 12 (6), 1261–1270. https://doi.org/10.1023/A:1023091922825.

Genetic Algorithms for Multiobjective Optimization: FormulationDiscussion and Generalization. In: Fonseca, C.M., Fleming, P.J. (Eds.), Icga.

Fu, G., Kapelan, Z., Reed, P., 2012. Reducing the complexity of multiobjective water distribution system optimization through global sensitivity analysis. J. Water Resour. Plann. Manag. 138 (3), 196–207. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000171.

García-Martínez, C., Cordón, O., Herrera, F., 2007. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. Eur.

J. Oper. Res. 180 (1), 116–148.
Garner, G.G., Keller, K., 2018. Using direct policy search to identify robust strategies in adapting to uncertain sea-level rise and storm surge. Environ. Model. Software 107, 96–104. https://doi.org/10.1016/j.envsoft.2018.05.006.
Gibbs, M.S., Maier, H.R., Dandy, G.C., 2011. Relationship between problem characteristics and the optimal number of genetic algorithm generations. Eng. Optim. 43 (4), 349–376.
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. Software 69, 226–239.
Gibson, G.A., Meter, R.V., 2000. Network attached storage architecture. Commun. ACM 43 (11), 37–45. https://doi.org/10.1145/353360.353362.
Giupponi, C., 2007. Decision support systems for implementing the European water framework directive: the MULINO approach. Environ. Model. Software 22 (2), 248–258.
Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. Comput. Oper. Res. 13 (5), 533–549. https://doi.org/10.1016/0305-0548(86)90048-1.
Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA.
Golding, P., Kapadia, S., Naylor, S., Schulz, J., Maier, H.R., Lall, U., van der Velde, M., 2017. Framework for minimising the impact of regional shocks on global food security using multi-objective ant colony optimisation. Environ. Model. Software 95, 303–319. https://doi.org/10.1016/j.envsoft.2017.06.004.
Gropp, W., Hoefler, T., Thakur, R., Lusk, E., 2014. Using advanced MPI: Modern features of the message-passing interface. MIT Press.
Gupta, H.V., Sorooshian, S., Yapo, P.O., 1999. Status of automatic calibration for hydrologic models: Comparison with multilevel expert calibration. J. Hydrol. Eng. 4 (2), 135–143.
Hadka, D., 2016. Beginner's Guide to the MOEA Framework. CreateSpace Independent Publishing Platform.
Hadka, D., Reed, P., 2013. Borg: An auto-adaptive many-objective evolutionary computing framework. Evol. Comput. 21 (2), 231–259.
Hadka, D., Reed, P., 2015. Large-scale parallelization of the Borg multiobjective evolutionary algorithm to enhance the management of complex environmental systems. Environ. Model. Software 69, 353–369. https://doi.org/10.1016/j.envsoft.2014.10.014.
Hamilton, S.H., ElSawah, S., Guillaume, J.H., Jakeman, A.J., Pierce, S.A., 2015. Integrated assessment and modelling: overview and synthesis of salient dimensions. Environ. Model. Software 64, 215–229.
Hansen, N., Auger, A., Mersmann, O., Tusar, T., Brockhoff, D., 2016. COCO: A platform for comparing continuous optimizers in a black-box setting. arXiv preprint arXiv:1603.08785.
Hashimoto, T., Stedinger, J.R., Loucks, D.P., 1982. Reliability, resiliency, and vulnerability criteria for water resource system performance evaluation. Water Resour. Res. 18 (1), 14–20.
He, C., 2003. Integration of geographic information systems and simulation model for watershed management. Environ. Model. Software 18 (8–9), 809–813.
Herstein, L., Filion, Y., Hall, K., 2010. Evaluating the environmental impacts of water distribution systems by using EIO-LCA-based multiobjective optimization. J. Water Resour. Plann. Manag. 137 (2), 162–172.
Holland, J., 1975. Adaptation in artificial and natural systems. The University of Michigan Press. https://github.com/jeffrey-newman/WSMGA-with-Wrapper-and-Analytics, Accessed date: 26 November 2018.
Humphrey, G.B., Gibbs, M.S., Dandy, G.C., Maier, H.R., 2016. A hybrid approach to monthly streamflow forecasting: integrating hydrological model outputs into a Bayesian artificial neural network. J. Hydrol. 540, 623–640.
Hunter, J.M., Maier, H.R., Gibbs, M.S., Foale, E.R., Grosvenor, N.A., Harders, N.P., Kikuchi-Miller, T.C., 2018. Framework for developing hybrid process-driven, artificial neural network and regression models for salinity prediction in river systems. Hydrol. Earth Syst. Sci. 22 (5), 2987–3006.
Hyde, K.M., Maier, H.R., 2006. Distance-based and stochastic uncertainty analysis for multi-criteria decision analysis in excel using visual basic for applications. Environ. Model. Software 21 (12), 1695–1710. https://doi.org/10.1016/j.envsoft.2005.08.004.
Jakeman, A.J., Barreteau, O., Hunt, R.J., Rinaudo, J.D., Ross, A. (Eds.), 2016. Integrated Groundwater Management: Concepts, Approaches and Challenges. Springer (open access), pp. 762.
Kaim, A., Cord, A.F., Volk, M., 2018. A review of multi-criteria optimization techniques for agricultural land use allocation. Environ. Model. Software 105, 79–93.
Kasprzyk, J.R., Nataraj, S., Reed, P.M., Lempert, R.J., 2013. Many objective robust decision making for complex environmental systems undergoing change. Environ. Model. Software 42, 55–71.
Particle swarm optimization. In: Kennedy, J., Eberhart, R. (Eds.), Proceedings of IEEE international conference on neural networks, Perth, Australia.
Kessler, A., Shamir, U., 1989. Analysis of the linear programming gradient method for optimal design of water supply networks. Water Resour. Res. 25 (7), 1469–1480. https://doi.org/10.1029/WR025i007p01469.
Kingston, G.B., Maier, H.R., Lambert, M.F., 2005. Calibration and validation of neural networks to ensure physically plausible hydrological modeling. J. Hydrol. 314 (1–4), 158–176.
Knowles, J.D., Corne, D.W., 2000. Approximating the nondominated front using the Pareto archived evolution strategy. Evol. Comput. 8 (2), 149–172. https://doi.org/10.1162/106365600568167.
Kurschl, W., Pimminger, S., Wagner, S., Heinzelreiter, J. (Eds.), 2014. Concepts and requirements for a cloud-based optimization service, In 2014 Asia-Pacific Conference on Computer Aided System Engineering (APCASE), . https://ieeexplore.ieee.org/document/6924464/.
Kwakkel, J.H., 2017. The Exploratory Modeling Workbench: An open source toolkit for exploratory modeling, scenario discovery, and (multi-objective) robust decision making. Environ. Model. Software 96, 239–250. https://doi.org/10.1016/j.envsoft.2017.06.054.
Laniak, G.F., Olchin, G., Goodall, J., Voinov, A., Hill, M., Glynn, P., Whelan, G., Geller, G., Quinn, N., Blind, M., Peckham, S., Reaney, S., Gaber, N., Kennedy, R., Hughes, A., 2013. Integrated environmental modeling: A vision and roadmap for the future. Environ. Model. Software 39, 3–23. https://doi.org/10.1016/j.envsoft.2012.09.006.
Liebman, J.C., 1976. Some simple-minded observations on the role of optimization in public systems decision-making. Interfaces 6 (4), 102–108. https://doi.org/10.1287/inte.6.4.102.
Liner, B.L., Maier, H.R., 2015. H2optimization: An Introduction to Optimization and Operations Research for Infrastructure Professionals. Aldera.
Liu, W., Chen, W., Peng, C., 2014. Assessing the effectiveness of green infrastructures on urban flooding reduction: A community scale study. Ecol. Model. 291, 6–14. https://doi.org/10.1016/j.ecolmodel.2014.07.012.
Lozano, M., García-Martínez, C., 2010. Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. Comput. Oper. Res. 37 (3), 481–497. https://doi.org/10.1016/j.cor.2009.02.010.
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. Environ. Model. Software 62, 271–299. https://doi.org/10.1016/j.envsoft.2014.09.013.
Maier, H., Kapelan, Z., Kasprzyk, J., Matott, L., 2015. Thematic issue on evolutionary algorithms in water resources. Environ. Model. Software 69 (C), 222–225.
Maier, H.R., Guillaume, J.H.A., van Delden, H., Riddell, G.A., Haasnoot, M., Kwakkel, J.H., 2016. An uncertain future, deep uncertainty, scenarios, robustness and adaptation: How do they fit together? Environ. Model. Software 81, 154–164. https://doi.org/10.1016/j.envsoft.2016.03.014.
Mala-Jetmarova, H., Sultanova, N., Savic, D., 2017. Lost in optimisation of water distribution systems? A literature review of system operation. Environ. Model. Software 93, 209–254.
Matott, L.S., Babendreier Justin, E., Purucker, S.T., 2009. Evaluating uncertainty in integrated environmental models: A review of concepts and tools. Water Resour. Res. 45 (6). https://doi.org/10.1029/2008WR007301.
McJeon, H., Edmonds, J., Bauer, N., Clarke, L., Fisher, B., Flannery, B.P., Hilaire, J., Krey, V., Marangoni, G., Mi, R., 2014. Limited impact on decadal-scale climate change from increased use of natural gas. Nature 514 (7523), 482.
McPhail, C., Maier, H., Kwakkel, J., Giuliani, M., Castelletti, A., Westra, S., 2018. Robustness metrics: how are they calculated, when should they be used and why do they give different results? Earth's Future 6 (2), 169–191.
Merritt, W., Fu, B., Ticehurst, J., El Sawah, S., Vigiak, O., Roberts, A., Dyer, F., Pollino, C., Guillaume, J., Croke, B., 2017. Realizing modelling outcomes: A synthesis of success factors and their use in a retrospective analysis of 15 Australian water resource projects. Environ. Model. Software 94, 63–72.
Michalewicz, Z., 1992. Genetic algorithms+ Data structures= Evolution program. Artificial Intelligence. Springer, Berlin 1992.
Moore, A.D., Holzworth, D.P., Herrmann, N.I., Brown, H.E., de Voil, P.G., Snow, V.O., Zurcher, E.J., Huth, N.I., 2014. Modelling the manager: Representing rule-based management in farming systems simulation models. Environ. Model. Software 62, 399–410. https://doi.org/10.1016/j.envsoft.2014.09.001.
Morley, D.W., Gulliver, J., 2018. A land use regression variable generation, modelling and prediction tool for air pollution exposure assessment. Environ. Model. Software 105, 17–23.
Newland, C.P., Maier, H.R., Zecchin, A.C., Newman, J.P., van Delden, H., 2018. Multi-objective optimisation framework for calibration of Cellular Automata land-use models. Environ. Model. Software 100, 175–200. https://doi.org/10.1016/j.envsoft.2017.11.012.
Newman, J.P., Maier, H.R., Riddell, G.A., Zecchin, A.C., Daniell, J.E., Schaefer, A.M., van Delden, H., Khazai, B., O'Flaherty, M.J., Newland, C.P., 2017. Review of literature on decision support systems for natural hazard risk reduction: current status and future research directions. Environ. Model. Software 96, 378–409.
Nguyen, D.C.H., Maier, H.R., Dandy, G.C., Ascough, J.C., 2016. Framework for computationally efficient optimal crop and water allocation using ant colony optimization. Environ. Model. Software 76, 37–53. https://doi.org/10.1016/j.envsoft.2015.11.003.
Nicklow, J., Reed, P., Savic, D., Dessalegne, T., Harrell, L., Chan-Hilton, A., Karamouz, M., Minsker, B., Ostfeld, A., Singh, A., 2009. State of the art for genetic algorithms and beyond in water resources planning and management. J. Water Resour. Plann. Manag. 136 (4), 412–432.
Pareto, V., 1896. Course of Political Economy. Lausanne.
Paton, F.L., Maier, H.R., Dandy, G.C., 2014. Including adaptation and mitigation responses to climate change in a multiobjective evolutionary algorithm framework for urban water supply systems incorporating GHG emissions. Water Resour. Res. 50 (8), 6285–6304.
Piscopo, A.N., Kasprzyk, J.R., Neupauer, R.M., 2015. An iterative approach to multi-objective engineering design: Optimization of engineered injection and extraction for enhanced groundwater remediation. Environ. Model. Software 69, 253–261. https://doi.org/10.1016/j.envsoft.2014.08.030.
Quinn, J.D., Reed, P.M., Keller, K., 2017. Direct policy search for robust multi-objective management of deeply uncertain socio-ecological tipping points. Environ. Model. Software 92, 125–141. https://doi.org/10.1016/j.envsoft.2017.02.017.

Ravalico, J.K., Dandy, G.C., Maier, H.R., 2010. Management Option Rank Equivalence (MORE)–A new method of sensitivity analysis for decision-making. Environ. Model. Software 25 (2), 171–181.
Razavi, S., Gupta, H.V., 2015. What do we mean by sensitivity analysis? The need for comprehensive characterization of "global" sensitivity in Earth and Environmental systems models. Water Resour. Res. 51 (5), 3070–3092.
Razavi, S., Tolson, B.A., 2013. An efficient framework for hydrologic model calibration on long data periods. Water Resour. Res. 49 (12), 8418–8431.
Razavi, S., Tolson, B.A., Matott, L.S., Thomson, N.R., MacLean, A., Seglenieks, F.R., 2010. Reducing the computational cost of automatic calibration through model preemption. Water Resour. Res. 46 (11). https://doi.org/10.1029/2009WR008957.
Razavi, S., Tolson, B.A., Burn, D.H., 2012. Review of surrogate modeling in water resources. Water Resour. Res. 48 (7). https://doi.org/10.1029/2011WR011527.
Razavi, S., Asadzadeh, M., Tolson, B., Fay, D., Moin, S., Bruxer, J., Fan, Y., 2013. Evaluation of new control structures for regulating the Great Lakes system: Multiscenario, multireservoir optimization approach. J. Water Resour. Plann. Manag. 140 (8), 04014018.
Reddy, M.J., Kumar, D.N., 2006. Optimal reservoir operation using multi-objective evolutionary algorithm. Water Resour. Manag. 20 (6), 861–878.
Reed, P.M., Hadka, D., Herman, J.D., Kasprzyk, J.R., Kollat, J.B., 2013. Evolutionary multiobjective optimization in water resources: The past, present, and future. Adv. Water Resour. 51, 438–456.
Reis, S., Seto, E., Northcross, A., Quinn, N.W., Convertino, M., Jones, R.L., Maier, H.R., Schlink, U., Steinle, S., Vieno, M., 2015. Integrating modelling and smart sensors for environmental and human health. Environ. Model. Software 74, 238–246.
Rossman, L.A., 2000. EPANET 2 user manual. U.S. Environmental Protection Agency, Cincinnati.
Samani, H.M.V., Mottaghi, A., 2006. Optimization of water distribution networks using integer linear programming. J. Hydraul. Eng. 132 (5), 501–509 doi: 10.1061/(ASCE)0733-9429(2006)132:5(501).
Shin, M.-J., Guillaume, J.H., Croke, B.F., Jakeman, A.J., 2015. A review of foundational methods for checking the structural identifiability of models: results for rainfall-runoff. J. Hydrol. 520, 1–16.
Simon, H.A., 1996. The sciences of the artificial. MIT press.
Singh, A., 2014. Groundwater resources management through the applications of simulation modeling: A review. Sci. Total Environ. 499, 414–423.
Smith, R., Kasprzyk, J., Zagona, E., 2015. Many-objective analysis to optimize pumping and releases in multireservoir water supply network. J. Water Resour. Plann. Manag. 142 (2), 04015049. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000576.
Smith, R., Kasprzyk, J., Dilling, L., 2017. Participatory Framework for Assessment and Improvement of Tools (ParFAIT): Increasing the impact and relevance of water management decision support research. Environ. Model. Software 95, 432–446. https://doi.org/10.1016/j.envsoft.2017.05.004.
Sörensen, K., 2013. Metaheuristics—the metaphor exposed. Int. Trans. Oper. Res. 22 (1), 3–18. https://doi.org/10.1111/itor.12001.
Stokes, C.S., Maier, H.R., Simpson, A.R., 2014a. Water distribution system pumping operational greenhouse gas emissions minimization by considering time-dependent emissions factors. J. Water Resour. Plann. Manag. 141 (7), 04014088.
Stokes, C.S., Simpson, A.R., Maier, H.R., 2014b. The cost–greenhouse gas emission nexus for water distribution systems including the consideration of energy generating infrastructure: an integrated conceptual optimization framework and review of literature. Earth Perspectives 1 (1), 9.
Stokes, C.S., Maier, H.R., Simpson, A.R., 2015a. Effect of storage tank size on the minimization of water distribution system cost and greenhouse gas emissions while considering time-dependent emissions factors. J. Water Resour. Plann. Manag. 142 (2), 04015052.
Stokes, C.S., Simpson, A.R., Maier, H.R., 2015b. A computational software tool for the minimization of costs and greenhouse gas emissions associated with water distribution systems. Environ. Model. Software 69, 452–467.
Stützle, T., Hoos, H.H., 2000. MAX–MIN ant system. Future Generat. Comput. Syst. 16 (8), 889–914.
Swan, J., Adriaensen, S., Bishr, M., Burke, E.K., Clark, J.A., De Causmaecker, P., Durillo, J., Hammond, K., Hart, E., Johnson, C.G. (Eds.), 2015. A research agenda for metaheuristic standardization, In Proceedings of the XI Metaheuristics International Conference.
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, wetlands, and floodplains. Water Resour. Res. 48 (8). https://doi.org/10.1029/2011WR011276.
Szemis, J.M., Maier, H.R., Dandy, G.C., 2014. An adaptive ant colony optimization framework for scheduling environmental flow management alternatives under varied environmental water availability conditions. Water Resour. Res. 50 (10), 7606–7625.
Tang, Y., Reed, P.M., Kollat, J.B., 2007. Parallelization strategies for rapid and robust evolutionary multiobjective optimization in water resources applications. Adv. Water Resour. 30 (3), 335–353. https://doi.org/10.1016/j.advwatres.2006.06.006.
Teegavarapu, R.S.V., Simonovic, S.P., 2002. Optimal operation of reservoir systems using simulated annealing. Water Resour. Manag. 16 (5), 401–428.
Teng, J., Jakeman, A., Vaze, J., Croke, B.F., Dutta, D., Kim, S., 2017. Flood inundation modelling: A review of methods, recent advances and uncertainty analysis. Environ. Model. Software 90, 201–216.
Timani, B., Peralta, R., 2017. Aggregated surrogate simulator for groundwater-surface water management via simulation-optimization modeling: Theory, development and tests. Environ. Model. Software 96, 210–231.
Tolson, B.A., Khedr, A., Asadzadeh, M., 2012. The battle of the water networks (BWN-II): PADDS based solution approach. In: WDSA 2012: 14th Water Distribution Systems Analysis Conference, 24-27 September 2012 in Adelaide, South Australia. http://www.civil.uwaterloo.ca/btolson/software.aspx.
van Keulen, H., Asseng, S., 2018. Simulation models as tools for crop management. Encyclopedia of Sustainability Science and Technology 1–20.
Vasquez, J.A., Maier, H.R., Lence, B.J., Tolson, B.A., Foschi, R.O., 2000. Achieving water quality system reliability using genetic algorithms. J. Environ. Eng. 126 (10), 954–962.
Verstegen, J.A., Jonker, J.G.G., Karssenberg, D., van der Hilst, F., Schmitz, O., de Jong, S.M., Faaij, A.P., 2017. How a Pareto frontier complements scenario projections in land use change impact assessment. Environ. Model. Software 97, 287–302.
Vrugt, J.A., Gupta, H.V., Bastidas, L.A., Bouten, W., Sorooshian, S., 2003. Effective and efficient algorithm for multiobjective optimization of hydrologic models. Water Resour. Res. 39 (8). https://doi.org/10.1029/2002WR001746.
Vrugt, J.A., Robinson, B.A., Hyman, J.M., 2009. Self-adaptive multimethod search for global optimization in real-parameter spaces. IEEE Trans. Evol. Comput. 13 (2), 243–259.
Wang, Q., Savić, D., Kapelan, Z., 2017. GALAXY: A new hybrid MOEA for the optimal design of water distribution systems. Water Resour. Res. 53 (3), 1997–2015.
Weyland, D., 2012. A rigorous analysis of the harmony search algorithm: how the research community can be Misled by a "Novel" methodology. In: Modeling, Analysis, and Applications in Metaheuristic Computing: Advancements and Trends. IGI Global, pp. 72–83.
Weyland, D., 2015. A critical analysis of the harmony search algorithm—How not to solve sudoku. Operations Research Perspectives 2, 97–105. https://doi.org/10.1016/j.orp.2015.04.001.
Wilhelmstötter, F., 2018. Jenetics library user's manual 4.1. http://jenetics.io/, Accessed date: 12 April 2018.
Wu, W., Maier, H.R., Simpson, A.R., 2010a. Single-objective versus multiobjective optimization of water distribution systems accounting for greenhouse gas emissions by carbon pricing. J. Water Resour. Plann. Manag. 136 (5), 555–565.
Wu, W., Simpson, A.R., Maier, H.R., 2010b. Accounting for greenhouse gas emissions in multiobjective genetic algorithm optimization of water distribution systems. J. Water Resour. Plann. Manag. 136 (2), 146–155.
Wu, W., May, R.J., Maier, H.R., Dandy, G.C., 2013. A benchmarking approach for comparing data splitting methods for modeling water resources parameters using artificial neural networks. Water Resour. Res. 49 (11), 7598–7614.
Wu, W., Maier, H.R., Dandy, G.C., Leonard, R., Bellette, K., Cuddy, S., Maheepala, S., 2016. Including stakeholder input in formulating and solving real-world optimisation problems: Generic framework and case study. Environ. Model. Software 79, 197–213. https://doi.org/10.1016/j.envsoft.2016.02.012.
Wu, W., Dandy, G.C., Maier, H.R., Maheepala, S., Marchi, A., Mirza, F., 2017. Identification of optimal water supply portfolios for a major city. J. Water Resour. Plann. Manag. 143 (9), 05017007.
Yandamuri, S.R., Srinivasan, K., Murty Bhallamudi, S., 2006. Multiobjective optimal waste load allocation models for rivers using nondominated sorting genetic algorithm-II. J. Water Resour. Plann. Manag. 132 (3), 133–143.
Yapo, P.O., Gupta, H.V., Sorooshian, S., 1998. Multi-objective global optimization for hydrologic models. J. Hydrol. 204 (1–4), 83–97.
Yigitcanlar, T., Teriman, S., 2015. Rethinking sustainable urban development: towards an integrated planning and development process. Int. J. Environ. Sci. Technol. 12 (1), 341–352.
Zavala, G.R., Nebro, A.J., Luna, F., Coello Coello, C.A., 2014. A survey of multi-objective metaheuristics applied to structural optimization. Struct. Multidiscip. Optim. 49 (4), 537–558. https://doi.org/10.1007/s00158-013-0996-4.
Zecchin, A.C., Simpson, A.R., Maier, H.R., Nixon, J.B., 2005. Parametric study for an ant algorithm applied to water distribution system optimization. IEEE Trans. Evol. Comput. 9 (2), 175–191.
Zecchin, A.C., Maier, H.R., Simpson, A.R., Leonard, M., Nixon, J.B., 2007. Ant colony optimization applied to water distribution system design: Comparative study of five algorithms. J. Water Resour. Plann. Manag. 133 (1), 87–92.
Zheng, F., Zecchin, A.C., Newman, J.P., Maier, H.R., Dandy, G.C., 2017. An adaptive convergence-trajectory controlled ant colony optimization algorithm with application to water distribution system design problems. IEEE Trans. Evol. Comput. 21 (5), 773–791.
Zheng, F., Maier, H.R., Wu, W., Dandy, G.C., Gupta, H.V., Zhang, T., 2018. On lack of robustness in hydrological model development due to absence of guidelines for selecting calibration and evaluation data: demonstration for data-driven models. Water Resour. Res. 54 (2), 1013–1030.
Zitzler, E., Laumanns, M., Bleuler, S., 2004. A tutorial on evolutionary multiobjective optimization. In: Metaheuristics for multiobjective optimisation. Springer, pp. 3–37.
Ziztler, E., Laumanns, M., Thiele, L., 2002. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou, K., Tsahalis, D., Periaux, J., Papailiou, K., Fogarty, T. (Eds.), Evolutionary Methods for Design, Optimization, and Control, Barcelona, Spain, pp. 95–100.