energies

MDPI

Article

Metaheuristic Optimization of Power and Energy Systems: Underlying Principles and Main Issues of the 'Rush to Heuristics'

Gianfranco Chicco * and **Andrea Mazza**

Dipartimento Energia "Galileo Ferraris", Politecnico di Torino, 10129 Torino, Italy; andrea.mazza@polito.it
* Correspondence: gianfranco.chicco@polito.it

check for updates

Abstract: In the power and energy systems area, a progressive increase of literature contributions that contain applications of metaheuristic algorithms is occurring. In many cases, these applications are merely aimed at proposing the testing of an existing metaheuristic algorithm on a specific problem, claiming that the proposed method is better than other methods that are based on weak comparisons. This 'rush to heuristics' does not happen in the evolutionary computation domain, where the rules for setting up rigorous comparisons are stricter but are typical of the domains of application of the metaheuristics. This paper considers the applications to power and energy systems and aims at providing a comprehensive view of the main issues that concern the use of metaheuristics for global optimization problems. A set of underlying principles that characterize the metaheuristic algorithms is presented. The customization of metaheuristic algorithms to fit the constraints of specific problems is discussed. Some weaknesses and pitfalls that are found in literature contributions are identified, and specific guidelines are provided regarding how to prepare sound contributions on the application of metaheuristic algorithms to specific problems.

1. Introduction

Large-scale complex optimization problems, in which the number of variables is high and the structure of the problem contains non-linearities and multiple local optima, are computationally challenging to solve, as they would require computational time beyond reasonable limits, and/or excessive memory with respect to the available computing capabilities. These problems appear in many domains, with different characteristics. Objective functions with non-smooth surfaces that correspond to the solution points, the presence of discrete variables, and several local minima, as well as the combinatorial explosion of the number of cases to be evaluated to search for the global optimum, characterize a number of optimization problems. The algorithms used to solve these problems have to be able to perform efficient *global optimization*.

In general, the solution algorithms are based on two types of methods:

(1) *deterministic methods*, in which the solution strategy is driven by well-identified rules, with *no random* components; and,

(2) *probability-based methods*, whose evolution depends on *random* choices carried out during the evolution of the solution process.

If the nature and size of the problem enable convenient formulations (e.g., linearizing the non-linear components by means of piecewise linear representations in a convex problem structure), some *exact*

deterministic methods can reach the global optimum under specific data representations. However, in general, the structure of the problems can be so complex to make it impossible or impracticable to use methods that can guarantee to reach the global optimum. Furthermore, in some cases, even the structure of the solution space is unknown, thus needing a specific approach to obtain information from the solutions themselves. In these cases, the use of metaheuristics is a viable approach.

What is a metaheuristic? In synthesis, the term *heuristic* identifies a tool that helps us discover 'something'. The term *meta* is typically added to represent the presence of a higher-level strategy that drives the search of the solutions. The metaheuristics could depend on the specific problem [1]. Many metaheuristics are based on translating the representation of *natural* phenomena or *physical processes* into computational tools [2].

Some solution methods are *metaheuristics* that are based on one of the following mechanisms:

(a) *Single solution update*: a succession of solutions is calculated, each time updating the solution only if the new one satisfies a predefined criterion. These methods are also called *trajectory* methods.

(b) *Population-based search*: many entities are simultaneously sent in parallel to solve the same problem. Subsequently, the collective behavior can be modeled to link the different entities with each other, and, in general, the best solution is maintained for the next phase of the search.

Detailed surveys of single solution-based and population-based metaheuristics are presented (among others) in [3–5].

On another point of view, when considering the number of optimization objectives, a distinction can be indicated among:

(i) *Single objective* optimization, in which there is only one objective to be minimized or maximized.

(ii) *Multi-objective* optimization, in which there are two or more objectives to be minimized or maximized. Multi-objective optimization tools are significant to assist decision-making processes, when the objectives are conflicting with each other. In this case, an approach that is based on Pareto-dominance concepts becomes useful. In this approach, a solution is *non-dominated* when no other solution does exist with better values for all of the individual objective functions. The set of non-dominated solutions forms the *Pareto front*, which contains the compromise solutions among which the decision-maker can choose the preferred one. If the Pareto front is convex, the weighted sum of the objectives can be used to track the compromise solutions. However, in general, the Pareto front has a non-convex shape, which calls for appropriate solvers to construct it. During the solution process, the best-known Pareto front is updated until a specific stop criterion is satisfied. The best-known Pareto front should converge to the true Pareto front (that could be unknown). The solvers need to balance *convergence* (i.e., approaching a stable Pareto front) with *diversity* (i.e., keeping the solutions spread along the Pareto front, avoiding concentrating the solutions in limited zones). Diversity is represented by estimating the density of the solutions that are located around a given solution. For this purpose, a dedicated parameter, called *crowding distance*, is defined as the average distance between the given solution and the closest solutions belonging to the Pareto front (the number of solutions is user-defined).

(iii) *Many-objective* optimization, a subset of multi-objective optimization in which, conventionally, there are more than two objectives. This distinction is important, as some problems that are reasonably solvable in two dimensions, such as finding a balance between convergence and diversity, become much harder to solve in more than two dimensions. Moreover, by increasing the number of objectives, it becomes more intrinsically difficult to visualize the solutions in a way convenient for the decision-maker. The main challenges in many-objective optimization are summarized in [6]. When the number of objectives increases, the number of non-dominated solutions largely increases, even reaching situations in which almost all of the solutions become non-dominated [7]. This aspect heavily impacts on slowing down the solution process in methods that use Pareto dominance as a criterion to select the solutions. A large number of non-dominated

solutions may also require increasing the size of the population to be used in the solution method, which again results in a slower solution process. Finally, the calculation of the hyper-volume as a metric for comparing the effectiveness of the Pareto front construction from different methods [8] is geometrically simple in two dimensions, but it becomes progressively harder [9], with the exponential growth of the computational burden, when the number of objectives increases [10].

For multi-objective and many-objective problem formulations, the metaheuristic approach has gained momentum, because of the issues that exist in the application of gradient search and numerical programming methods. However, a number of issues appear, mainly concerning the characteristics of the search space (such as non-convexity and multimodality), and the presence of discrete non-uniform Pareto fronts [11].

The above indications set up the framework of analysis used in this paper. The main aims are to start from the concepts referring to the formulation of the metaheuristics and discuss a number of both correct and inappropriate practices found in the literature. Some details are provided on the applications in the power and energy systems domain, in which hundreds of papers that are based on the use of metaheuristics for solving optimization problems have been published.

The specific contributions of this paper are:

■ A systematic analysis of the state of the art regarding the main issues on the convergence of metaheuristics, and the comparisons among metaheuristic algorithms with suitable metrics for single-objective and multi-objective optimization problems.

■ The identification of a set of underlying principles that explain the characteristics of the various metaheuristics, and that can be used to search for similarities and complementarities in the definition of the metaheuristic algorithms.

■ The indication of some pitfalls and inappropriate statements sometimes found in literature contributions on global optimization through metaheuristic algorithms, which partially foster the proliferation of articles on the application of metaheuristics, are not always justified by a rigorous methodological approach, leading to an almost uncontrollable 'rush to heuristics'.

■ The discussion on the characteristics of some problems in the power and energy system domain, which are solved with metaheuristic optimization, highlight some problem-related customizations of the classical versions of the metaheuristic algorithms.

■ The indication of some hints for preparing sound contributions on the application of metaheuristic algorithms to power and energy system problems with one or more objective functions, while using statistically significant and sufficiently strong metrics for comparing the solutions, in such a way to mitigate the 'rush to heuristics'.

The next sections of this paper are organized, as follows. Section 2 summarizes the underlying principles that can be found in the construction of metaheuristic algorithms. Section 3 discusses the application of metaheuristic algorithms to specific problems for power and energy systems. Section 4 recalls the convergence properties of some metaheuristics. Section 5 provides a critical discussion and new insights on the comparison of metaheuristic algorithms. Section 6 deals with the hybridization of the metaheuristics. Section 7 addresses the use of metaheuristics for solving multi-objective problems. Section 8 discusses the effectiveness of metaheuristic-based optimization, pointing out a number of weak statements that should not appear in scientific contributions. The last section contains the Conclusions, which include specific guidelines for preparing sound contributions on the application of metaheuristic algorithms to power and energy system problems.

2. Materials and Methods

2.1. Evolution of the Metaheuristics

The evolution of the metaheuristics has progressively increased in time. New algorithms appear each year, and it is not clear-cut whether these algorithms bring new contents for the research on

evolutionary computation. Table A1 in the Appendix A reports a non-exhaustive list of over one hundred metaheuristics that have been applied in the power and energy systems field, together with the corresponding references (the years indicated refer to the first date of publication of relevant articles or books). Figure 1 shows the corresponding number of metaheuristics used during time. The number of metaheuristics that appeared in the last years is underestimated, as some recent metaheuristics (not indicated) have not yet found application in power and energy systems. Moreover, the list in Table A1 refers to basic versions of the metaheuristics only, without accounting for the proposed variants and hybridizations among heuristics; otherwise, the number of contributions would quickly rise to significantly higher numbers. However, the rush to apply a new metaheuristic to all of the engineering problems is a vulnerable point for scientific research [12], especially when each "new" method or variant applied to a given problem is claimed to become the best method, pretending to show its superiority with respect to any other existing method. Apparently, this 'rush to heuristics' is producing hundreds of articles, most of them being questionable in terms of methodological advances that are provided in the evolutionary computation field.
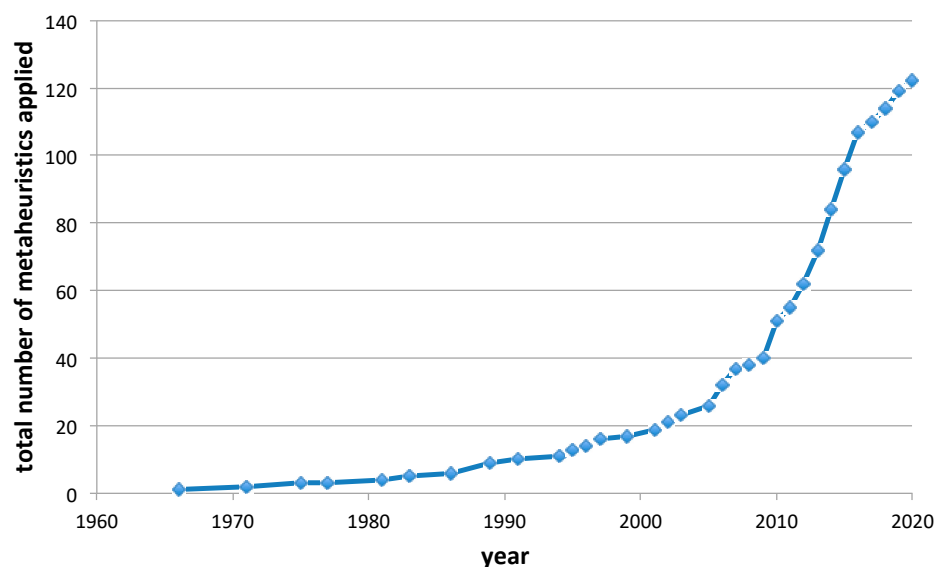

Figure 1. Number of metaheuristics available (variants and hybrid versions excluded).

The need to better understand the characteristics of the various algorithms has started specific discussions since the early phase of the development of new algorithms. Two decades ago, the unified view that was proposed in [13] started from the consideration that the implementation of the solvers was increasingly similar. The unified view was introduced under the name Adaptive Memory Programming (AMP), synthesizing a series of basic steps for the solution procedure valid for most metaheuristics (AMP is not applicable to single-update methods, such as Simulated Annealing [14]):

(1)   store a set of solutions;
(2)   construct a provisional solution using the available data;
(3)   improve the provisional solution with *local search* or another algorithm; and,
(4)   update the set of available solutions with the new solution.

These steps indicate four basic principles that re used to set up a metaheuristic algorithm, namely, memory (i.e., storage of information), the presence of a constructive mechanism, a local search strategy, and the definition of a mechanism for solution update. Taillard et al. [13] consider *memory* as a key principle for describing the possible similarities between the algorithmic structures of the metaheuristics. Indeed, memory is fundamental in the definition of the metaheuristics. However, memory can be seen as a general term with different meanings for different algorithms. As such, memory is not considered

here to be sufficiently specific as a detailed underlying principle, and more underlying principles are used for representing the characteristics of the various types of metaheuristics.

2.2. Underlying Principles

Each metaheuristic algorithm applies specific mechanisms in the solution procedure. The presence of a multitude of algorithms raises a fundamental question: are all of the metaheuristic algorithms used really different from each other?

To address this issue, the solution procedures have been revisited by identifying a set of underlying principles that form a common basis for the various methods [15]. On the other side, these principles embed the structural differences among the methods.

The following list of underlying principles has been found, which also highlights some contents that refer to typical issues that appear in the power and energy systems domain:

■ parallelism;
■ acceptance;
■ elitism;
■ selection;
■ decay (or reinforcement);
1. immunity;
■ self-adaptation; and,
■   topology.

A brief description of these principles follows.

2.3. Parallelism

The parallelism principle appears in population-based search, in which more entities are sent in parallel to perform the same task, and the obtained results are then compared. On the basis of the comparison, further principles are applied in order to determine the evolution of the individuals within the population or to create new populations.

2.4. Acceptance

The principle of acceptance appears in a threefold way:

1. *Temporarily* accept solutions that lead to objective function *worsening*, with the rationale of broadening the search space
2. In the treatment of the *constraints* applied to the objective function. The constraints can be handled in two different ways. The first way is to discard all solutions in which any violation appears. This way is applied to algorithms that use a non-penalized objective function, in which the initial conditions have to correspond to a feasible solution (for single-update methods) or to all feasible solutions (in population-based methods). The second way is to use a penalized objective function, which makes it possible to find a numerical value to any solution and avoid discarding any solution. In this case, all solutions are automatically accepted, and the initial conditions could correspond to infeasible solutions. The penalty factors that are used in the penalized objective functions have to be sufficiently high to obtain high values for the solutions with violations. However, if the penalty factor is too high, then very high values could appear for too many solutions, which makes it difficult to drive the search in the direction of exploring the search space efficiently.
3. Introducing a *threshold* for only accepting solutions that improve the current best solution at least of the value of the threshold. This way could help to avoid numerical issues in the comparison between values that result from previous calculations, e.g., when the same number is represented in different ways depending on numerical precisions.

2.5. Elitism

In the iterative population-based methods (in which more individuals are generated at the same iteration from probability-based criteria), if no action is done, then it is possible to lose the best solution passing from one iteration to another. The basic versions of metaheuristics (such as simulated annealing, genetic algorithms, and others) privilege the randomness of the search and do not contain mechanisms to preserve the best solutions. To avoid this, the *elitism* principle is applied by storing the individual with the best objective function found so far and passing it from one iteration to the next one. The best solution can be used as a reference individual to form other modified solutions, and it is immediately updated when another best solution is found. In a more extensive way, the elitism principle can also be applied to more than one individual, passing an *élite group* of solutions to the next iteration. The elitism principle has resulted in being very effective in practical applications. For the elitistic versions of some metaheuristics, it has been possible to prove convergence to the global optimum under specified conditions (see Section 3).

2.6. Selection

In a probability-based method, a mechanism has to be identified in order to extract a number of individuals at random from an available population, possibly associating weights to the probabilistic choices. In particular, for problems with variables that are described in a discrete way, the extraction mechanism is driven by the conventional way to extract a point from a given probability distribution. The Cumulative Distribution Function (CDF) is constructed by considering a quality measure (fitness) of the solutions, reported in a normalized way, such as the individuals corresponding to better values of the objective function have higher fitness (hence, higher probability to be chosen). For example, let us consider a set of M individuals, whose objective function values are $\{v_m > 0, \ m = 1, \ldots, M\}$, and the objective function has to be minimized [16]. The fitness is defined as $\psi_m = v_m / \sum_{q=1}^{M} v_q$. Subsequently, a random number r is extracted from a uniform probability distribution in [0, 1] and is entered on the vertical axis of the CDF. The individual corresponding to the discrete position on the *horizontal* axis is then selected. Figure 2 exemplifies the situation, with four individuals and the related fitness values of 0.2, 0.3, 0.4, and 0.1, respectively. By extracting a random number (e.g., 0.62), the individual number 3 is selected.
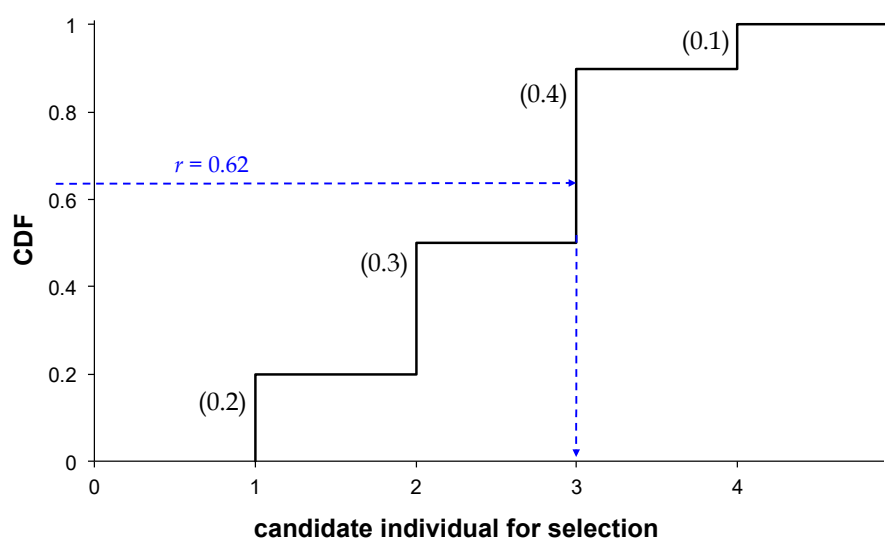

Figure 2. Random selection from Cumulative Distribution Function (CDF).

This method is equivalent to the so-called biased roulette wheel, in which the slices have a different amplitude (proportional to the fitness that is associated to each individual). The selected variable is the

one seen from the observation at which the roulette stops. In the example of Figure 3, individual D has the largest probability of being selected, but *any* individual can be selected (e.g., the individual H is selected in Figure 3).
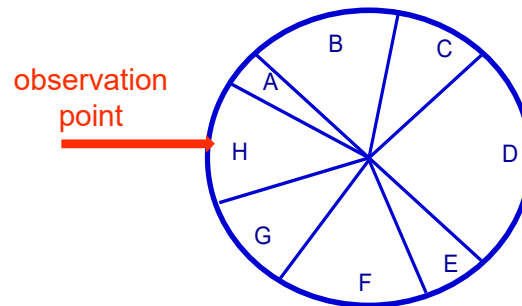

Figure 3. Random selection from biased roulette wheel.

2.7. Decay (or Reinforcement)

The decay principle may be applied in order to enable larger initial flexibility in the application of the method, followed by progressive restrictions of that flexibility. The application of a decay rate to the parameter (cooling rate) that drives the external cycle in the simulated annealing method is a direct example. Decay is typically considered by using a multiplicative factor lower than unity, which is applied at successive iterations. In some cases, reinforcement is applied in a similar way by using a multiplicative factor that is higher than unity.

The decay principle can also reduce the strength of some search paths that are less convenient than others or have not been recently visited. This application has been introduced in the ant colony optimization algorithms [17], in which the paths can also be reinforced if they were found to be convenient. Relative decay has also been considered in the hyper-cube ant colony optimization framework [18], in which decay or reinforcement are first applied, and then the overall outcomes are normalized to fit a hyper-cube with dimensions limited inside the interval [0, 1].

2.8. Immunity

Immunity is applied by identifying some properties of the solutions, where such properties lead to satisfactory configurations. Immunity gives *priority* to the solutions that have characteristics similar to those properties.

2.9. Self-Adaptation

Self-adaptation consists of changing the parameters of the algorithms in an automatic way, depending on the evolution of the procedure.

2.10. Topology

The topology principle is applied when the problem under analysis needs to satisfy specific constraints, such as the definition on a graph or connectivity requirements. A relevant example is the *graph* corresponding to the operational configuration of an electrical distribution system. The principle of topology is linked, for example, to the generation of radial structures during the execution of the algorithms. The representation of the topology is associated with how the information regarding the connections is coded, which can be more or less effective to ensure that only radial structures are progressively generated. For example, for an electrical network, the information coding is typically carried out in one of these ways:

(a)   creating the list of the open branches;

(b)   forming the list of the loops and identifying the branches of each loop with a progressive number; or,

(c)   using a *binary string* of length that is equal to the number of branches, containing the status (on/off) of the branches.

2.11. Remarks on the Underlying Principles

The identification of the underlying principles has clarified that the memory term can be intended in different ways, e.g., pheromone for ant colony optimization, the presence of the previous population for genetic algorithms, the list of past moves for tabu search, and so on. Elitism itself is a form of memory.

In the use of metaheuristics, a balance is generally sought between exploration and exploitation of the search space. Exploration means the ability to reach all of the points in the search space, while exploitation refers to the use of knowledge from the solutions that are already found to drive the search towards more convenient regions. The underlying principles may affect both exploration and exploitation in different ways. For example, the selection principle applied to a given population could mainly refer to exploitation [19], as it drives the search towards the choice of the best individuals [20]. However, selection may, to a given extent, also refer to exploration, by varying the width of the population involved [21].

Finally, the synthesis of the underlying principles can also be a way to generate new metaheuristic algorithms or variants. Even automatic generation of algorithms could be considered, for which there is wide literature referring to deterministic and other algorithms [22]. Indeed, conceptually, there are different ways to proceed to define new metaheuristic algorithms:

(a)   Taking existing algorithms and constructing new ones by changing the context and the nomenclature; indeed, this practice is not advancing the state of the art and only contributes to add entropy to the evolutionary computation domain [12].

(b)   Synthesizing the underlying principles and combining them to obtain new algorithms; also, in this case, it is only a recombination of existing principles, which, in general, could not add significant contributions and would just play into the 'rush to heuristics'.

(c)   Generate new algorithms by using a set of components taken from promising approaches [23]. This line of research is also useful for identifying appropriate reusable portions of subprograms [24], and also leads to using *hyper-heuristics* to select or generate (meta-)heuristics by exploring a search space of a number of heuristics for identifying the most effective ones [25,26].

Some useful variants can be found when the existing metaheuristics are customized in order to solve specific problems, incorporating specific constraints, as indicated in the next section.

3. Specific Problems for Power and Energy Systems

3.1. Main Problems Solved with Metaheuristic Algorithms

In the power and energy domain, metaheuristic optimization is widely used to solve many problems referring to operation, planning, control, forecasting, reliability, security, and demand management. A set of typical problems that are solved with metaheuristic optimization have been considered in [27–29], including unit commitment, economic dispatch, optimal power flow, distribution system reconfiguration, power system planning, distribution system planning, load forecasting, and maintenance scheduling. Table 1 shows a selection of the metaheuristics most applied to these typical problems. From this set of problems, it emerges that the genetic algorithm is the most used or mentioned method for all of the problems, followed by particle swarm optimization (or simulated annealing in two cases, or tabu search in another case). Further review of the particle swarm optimization applications to power systems is presented in [30].

Table 1. Most used metaheuristic algorithms used to solve some power and energy systems problems [28,29].

Power and Energy Systems Problem	Most Used Metaheuristics
Unit commitment (UC)	Genetic algorithms, Particle swarm optimization, Evolutionary algorithms
Economic Dispatch (ED)	Genetic algorithms, Particle swarm optimization, Differential evolution, Evolutionary algorithms
Optimal Power Flow (OPF)	Genetic algorithms, Particle swarm optimization, Evolutionary algorithms, Differential evolution
Distribution System Reconfiguration (DSR)	Genetic algorithms, Particle swarm optimization, Simulated annealing, Ant colony optimization
Transmission Network Expansion Planning (TNEP)	Genetic algorithms, Simulated annealing, Tabu search, Particle swarm optimization
Distribution System Planning (DSP)	Genetic algorithms, Tabu search, Particle swarm optimization
Load and Generation Forecasting (LGF)	Genetic algorithms, Particle swarm optimization, Evolutionary algorithms, Simulated annealing
Maintenance Scheduling (MS)	Genetic algorithms, Simulated annealing, Particle swarm optimization, Tabu search

Concerning the information coding, the most successful implementations of genetic algorithms do not use binary coding of the strings, but use representations that are adapted to the application, and the crossover and mutation operators are re-defined accordingly [13]. However, binary coding schemes are still mainly used in power and energy system problems. In alternative, evolutionary programming schemes, in which the binary values are replaced with integer or real numbers, are appropriate for specific problems.

Some specific examples are presented below, in order to indicate how metaheuristics may be a viable alternative to (or a more successful option than) mathematical programming tools for the solution of large-scale optimization problems in the power and energy systems area. A general remark is that the size of the problem matters. If the size of the problem is limited, for which exhaustive search could be practicable, or mathematical programming tools that are able to provide exact solutions can be used with reasonable computational burden, then the use of a metaheuristic algorithm is not justified. The only exception is the case in which a metaheuristic algorithm is tested on a problem with known global optimum for checking its effectiveness in finding the global optimum, before applying it to large-scale problems: if the algorithm fails to find the global optimum on a small-size problem after adequate testing (e.g., with hundreds or thousands of executions), then its implementation or parameter setting are likely to be ineffective.

3.2. Unit Commitment (UC)

The UC problem consists of scheduling the generation units (typically thermal units) in order to serve the forecast demand in future periods (e.g., from one day to one week), by minimizing the total generation costs. The output is the start-up and shut-down schedule of these generation units. The problem has integer and continuous variables, and a complex set of constraints also involving time-dependent constraints for the units, such as minimum up and down times, start-up ramps, and time-dependent start-up costs. The UC problem has been traditionally solved with mathematical programming and stochastic programming tools [31]. However, these tools exhibit some drawbacks. For example, for dynamic programming, the computation time could become prohibitive for real-size systems, and time-dependent constraints are hard to be successfully implemented. Lagrangian relaxation has no problem with time-dependent constraints and it optimizes each unit separately. Thus, the dimension of the system is not an issue. The problem is solved using duality

theory, maximizing the dual objective function for a given original problem. However, because of the non-convexity of the original problem, the solution of the dual problem cannot guarantee the feasibility of the primal problem, and optimal values for the original and dual problems could be different. Robust optimizations with bi-level or three-level are computationally less demanding than stochastic programming models, but they may lead to over-conservative solutions. A framework for comparing mathematic programming algorithms to solve the UC problem has been formulated in [32], and it has been applied to three recently developed algorithms.

Metaheuristics have been successfully used to solve the UC problem to overcome these difficulties. First of all, the binary coding common to various metaheuristics is fully appropriate to represent the on/off status of the units. Thereby, the information on the status of each unit is included in a binary string with length that is equal to the number of time intervals considered. This information coding is naturally leading to the use of genetic algorithms [33], in which a unique string (called chromosome) is constructed as the ordered succession of the strings referring to the individual generation units. The whole information on the scheduling is then available at each time. However, Kazarlis et al. [33] showed that the straightforward application of the basic version of the genetic algorithm does not lead to acceptable performance, requiring the addition of specific problem-related operators to significantly enhance the algorithm performance. The need to define specific operators has also been confirmed in next implementations, for example, in [34]. Further review on the application of metaheuristics to the UC problem is presented in [35].

3.3. Economic Dispatch (ED)

Once the schedule of the generators has been fixed from UC, the share of the load satisfied by every generation unit is fixed through the ED, in such way that the total generation cost can be minimized. The ED problem is solved per single time step, by verifying generation (power bounds) and transmission system (line capacities) constraints. In ED, a specific cause of non-linearity is the valve-point effect that appears in the input-output curve of the generation units [36].

It is worth noting that, in the UC problem, the transmission system constraints are not considered, while, in the ED problem, some technical constraints (like ramping limits) are not taken into account. The joint solution of both UC and ED leads to the Network-Constrained Unit Commitment (NCUC) problem [37]. Its aims are to find (i) the time steps when the different generators are in operation and (ii) the power output of all the these generators over the time horizons, by taking into account power balance equations (equality constraints) and inequality constraints (such as minimum and maximum power, prohibited operating zones, multiple fuel options, transmission limits, etc.). The large-scale combinatorial nature of UC in the NCUC reduces the possibility of using deterministic methods, due to the difficulties of incorporating the various constraints, by opening the possibility of using metaheuristic methods. Various traditional programming algorithms are used, in particular dynamic programming [38], the interior point method [39,40], and further scenario-based decomposition methods that are applied to stochastic ED, e.g., using the asynchronous block iteration method to reduce the computational burden by using multicore computational architectures [41]. Metaheuristic algorithms, mainly genetic algorithms [36,42] and particle swarm optimization [43], have been used in the last decades to solve the ED problem, addressing specific challenges due to the non-convexity of the domain of definition of the variables. Further customized solvers have been set up using hybridizations of mathematical programming and metaheuristics, e.g., interior point and differential evolution [44], or hybrid versions that are based on particle swarm optimization with other methods [45].

3.4. Optimal Power Flow (OPF)

The OPF aims to find the steady state operating point in such a way that the system under analysis can be run in optimal way, by considering both single- and multi-objective formulations (including beyond costs also environmental or network compensation aspects). The control variables (that affect voltages and active powers at specific nodes) have to be chosen to satisfy a number of

constraints on the system components and operation. The OPF problem is a mixed-integer non-linear and non-convex problem. The OPF problem can be formulated in different ways, depending on which characteristics are considered (e.g., including reactive power-related aspects). In particular, the Security Constrained Optimal Power Flow (SCOPF) is a widely used formulation. A number of constraints, which are related to the network and the generators exists, and also constraints related to contingencies, are included. The tools adopted to solve the OPF problem are many, including a wide set of mathematical programming tools, also used to solve challenging real-time OPF problems [46]. Among them, interior point methods (e.g., [47]) have emerged among the most efficient solvers.

However, the nature of the OPF problem makes metaheuristic algorithms appropriate in providing effective solutions. Genetic algorithms, particle swarm optimization, evolutionary algorithms, and differential evolution are the most used metaheuristics [48]. Other methods come from well-designed hybridizations. For example, the differential evolutionary particle swarm optimization (DEEPSO) [49] was the winner of the 2014 competition that was organized by the Working Group on Modern Heuristic Optimization (under the IEEE Power and Energy Society Analytic Methods in Power Systems), dedicated to the solution of OPF problems. DEEPSO is a hybrid metaheuristic that applies the underlying principles of three heuristics (particle swarm optimization, evolutionary programming, and differential evolution) in order to construct an efficient tool. Its creation has followed well-studied criteria, which make it an example of best practice in the development of metaheuristics.

3.5. Distribution System Reconfiguration (DSR)

The DSR problem concerns the selection, within a weakly-meshed network, of the set of network branches to keep open to (i) obtain a radial network and (ii) optimize a predefined objective (or multi-objective) function. The main constraints refer to the need to operate a *radial* network, together with the equality constraint on the power balance and inequality constraints that involve node voltages, branch currents, short circuit currents, and others [50]. In this case, the discrete variables are the open/closed states of the network branches (or switches, with two switches for each branch, located at the branch terminals). The binary coding of the information is easily applicable. The length of the string is equal to the number of branches (or switches). Additional information, such as the branch list, can be added [51]. The number of the possible radial configurations in real-size systems is too high to allow for the construction of all radial configurations [52]. The structure of the problem makes it difficult to identify a neighborhood of the solutions and other regularities that could drive mathematical programming approaches. As such, metaheuristic algorithms are viable for approaching this problem.

The main issues for the DSR problem refer to the implementation of the constraints. In particular, the radiality constraint is not always easily incorporated in the metaheuristic algorithm. For some algorithms, such as simulated annealing, it is sufficient to exploit the branch-exchange mechanism, which consists of starting from the list of the open branches, closing (at random) an open branch to close (that will form a loop), identify the loop, and choosing (at random) within the loop a closed branch to open. In this way, the radial structure of the network is automatically guaranteed. However, for a genetic algorithm, the application of the crossover and mutation operators is not consistent with keeping the radial network structure. Hence, the crossover and mutation operators, and the information coding itself [53], have to be suitably re-defined to ensure that radiality is not lost during the solution process.

Nowadays, the presence of increasing share of distributed generation in the distribution network leads to consider operational conditions that are based on time-variable load and generation profiles. This aspect is posing challenges also regarding the creation of proper network samples for the algorithm tests [28]. In multi-objective problem formulations, more than one conflicting objectives are considered, such as losses and reliability indices [54].

3.6. Transmission Network Expansion Planning (TNEP)

The general objective of the TNEP problem is the minimization of the costs that are related to the transmission infrastructure, sometimes also associated to the planning of the generators connected to the system. Another objective is the reliability of the transmission system, considering the loss of load, or the interruption costs. The actions to be taken can be the installation of new lines, the repowering of the generator (or the insertion of new generators), and the embedding of new technologies (for example, Flexible AC Transmission Systems—FACTS). The traditional solution was based on the cost optimization for a given time period by considering a given set of fixed and variable costs. Today, the solutions have to take many uncertainties in generation, demand, market conditions, and technology developments into account. Furthermore, external aspects (including vulnerability issues [55]) have an impact on the transmission system reinforcement. This requires the development of a multi-scenario analysis.

The optimal set of investments is chosen by considering the power balance (equality constraint, which is usually verified by calculating a DC power flow [56]), and inequality constraints, usually referring to the maximum number of lines to be added, the capacity of each line, limits on the capability of the different generators and, in case, also a budget constraint on the availability of financial resources during the various planning stages. Further operation and security constraints can be introduced by considering electrical and natural gas networks [57], or the integration of wind systems with maximum wind capacity in a site and maximum risk for wind power installation [58].

Early algorithms [59] that are used constructive heuristics in which the network components are added one at a time, while using sensitivity measures to decide which is the next component to add. These sensitivity measures are local and, as such, cannot drive the solution in the direction of the global optimum. For large systems, the solution could reach a poor local optimum. For avoiding these issues, the exploration of alternative expansion possibilities was successfully introduced when the solutions achieved are considered to be weak. When alternative solutions are considered, the solutions grow exponentially with the system size, and there are typically many local optima. Hence, metaheuristic algorithms become viable and effective for solving the TNEP. In particular, genetic algorithms have been largely used. In the presence of a multi-objective problem, the multi-objective versions of genetic algorithms and other metaheuristics are particularly useful in providing effective solutions.

3.7. Distribution System Planning (DSP)

The DSP problem includes expansion planning and operational planning (with conventional and "active") procedures. The difference between expansion planning and operational planning lies into the number of nodes of the system, which remains constant when the operational planning procedure is applied, while it can change in the case of expansion planning.

For the expansion planning, different time horizons may be considered: short terms (1–4 years), long term (5–20 years), and horizon year planning (more than 20 years) [60]. The distribution expansion is a mixed-integer non-linear problem, in which binary variables represent either the installation of new device or the upgrade of the existing facilities, while continue variables are more related to all the time-variant variables, i.e., the Distributed Generation (DG) profiles or curtailed load [61]. Both the expansion and the operational planning have, as aims, the minimization of the investment costs, as well as the operational costs (usually losses and maintenance), when considering the technical and operational constraints.

The difference between conventional and active operational planning lies in the management of the DG: in the conventional case, the DG is installed and managed with a "fit and forget" approach (thus, it is included with a constant power, without considering the generation profiles). Conversely, the active operational planning aims to investigate the impact of DG by considering their generation profiles (including sometimes their uncertainty) as well as several load profile scenarios. For both cases, usual investments, such as new conductors or new substation components, are also considered [62]. In the case of expansion planning, the system operator may face an increase of the loads (or more

recently also the connection of new centralized power plants based on renewable energy sources), and thus additional electrical nodes have to be added. While the static approach involves what should be installed and where, the dynamic approach also specifies when the installation has to be made [63]. In the latter case, constraints regarding the time-relationships between the investments should be taken into account. Another problem that has been recently faced is the planning, including resilience aspects, which should handle with rare event having high impact [64,65]. In real-world applications, often the optimal investment choice requires to consider a number of aspects (not only economic, but also social and environmental) that can be managed through multi-criteria approaches, as reviewed in [66].

The metaheuristic methods are easy to be implemented and are particularly useful to solve multi-objective problems. In fact, the deterministic method mostly based on 0–1 linear programming become hard to manage when the number of variables and constraints increases. Branch-and-bound techniques can reduce the computational burden at the expense of reducing the solution space. However, for large-scale system, the solutions can trap into poor local optima. The possible success of genetic algorithms was envisioned more than two decades ago [67], and the reality has confirmed the success of the metaheuristic approach. The genetic algorithms are particularly appropriate, because of their binary coding of the information that enables the handling of the possible on/off states of the components that are considered as possible candidates to be added to the distribution network.

3.8. Load and Generation Forecasting (LGF)

Electrical load forecasting is a traditional problem, which has been solved with a number of methods, from statistical methods to approaches that are based on artificial intelligence, in particular neural networks and support vector machines, or more recently based on deep learning [68,69]. Today, the increase of the generation from renewable energy sources has introduced the further high uncertainty, depending on solar irradiance, wind speed and direction, and energy prices. Thereby, there is a need of approaches that also solve the generation forecasting. Load and generation forecasting are typically maintained as separate problems, due to the different nature of the corresponding time series and to the different phenomena that impact their evolution.

The forecasting time horizons are very important to define the problems. The classical view makes a distinction among very short term (e.g., from a few seconds to tens of minutes), short term (from tens of minutes to one day or one week), medium term (from one week to some months), and long term (from some months to many years).

Persistence models, which are based on replicating past time series considered as the closest one to the future conditions, are used as general benchmarks. Statistical approaches and methods that are based on neural networks and fuzzy systems have been used for many years. Hybrid methods have been constructed by adding to the neural networks an algorithm that assists parameter tuning in the training phase. Metaheuristic algorithms have been considered in these hybridizations [70]. Genetic algorithms are the most used, while particle swarm optimization, evolutionary algorithms, and simulated annealing have been used in various applications. Ensemble-based forecast models are emerging as effective tools, with the integration of different forecasting methods in order to reach better accuracy in the results [71].

3.9. Maintenance Scheduling (MS)

The MS problem aims to find the optimal time interval among the maintenance interventions on generation units and network components, with the aim to maintain their functionality and minimize the operational costs of the system where they are installed [72]. It is possible to define two different problems from the conceptual point of view: the generation unit maintenance scheduling (GMS) and the transmission maintenance scheduling (TMS). In the first case, the idea is the definition of the period of out of service of the generation units in terms of time occurrence and duration, by considering the reliability of the system where the generators are installed, the personnel availability, and the

limitation of the ramp rates of the units to come back to the normal operation. The definition of the maintenance periods is carried out according to objective functions that can only consider the reliability of the system (including the reserve margins), only the costs (fuel, start-up costs, loss of profit), or both of them [72,73]. When the TMS is considered, the main goal is to verify that the maintenance of the network component is not affecting the functionality of the system: thus, generally the constraints are the same as in GMS. The two problems may also be considered together, in order to account for both the security of the system and its efficiency. After the restructuring of the electricity business, the two problems may be conflicting because the generators would like to make the intervention when the electricity costs is low, which can lead to some difficulties to meet the total demand. Thus, an iterative process is required in order to fix scheduling periods taking into account the request of the generators and the network operator.

From the point of view of the solution methods, both mathematical programming approaches and metaheuristics have been used. Regarding the first group, dynamic programming, mixed-integer programming, Lagrangian relaxation, branch-and-bound, and Benders decomposition have been exploited [72]. However, all of those methods are suitable with linear objective and linear constraints. Thus, metaheuristics have been introduced to handle more complex objective functions and/or constraint formulation. Population-based methods (genetic algorithm and particle swarm optimization), simulated annealing, and tabu search have mostly been used, sometimes in a coordinated manner [74].

4. Convergence Aspects of Global Optimization Problems and Metaheuristics

Metaheuristic algorithms are also applied to solve global optimization problems when the problem structure is not known. For exploring the search space, these algorithms are generally based on the use of random variables, which make it possible to follow non-deterministic paths to reach a solution. The basic versions of the metaheuristic algorithms are relatively simple to be implemented, even though their customization to engineering problems could be very challenging. The metaheuristic algorithms are counterparts of stochastic methods, such as two-phase methods, random search methods, and random function methods [75]. In the two-phase methods, the objective function is assessed in a number of points selected at random. Subsequently, a local search is carried out to refine the solutions starting from these points. In the random search methods, a sequence of points is generated in the search space by considering some probability distributions, without following with local search. In the random function methods, a stochastic process that is consistent with the properties of the objective function has to be determined. With respect to these methods, the metaheuristic approach adds a high-level strategy that drives the solutions according to a specific rationale. However, the key point for confirming the significance of metaheuristics is the possibility of proving their convergence in a rigorous way. From the mathematical point of view, convergence proofs are not established for all metaheuristics. Two examples are provided:

(1) *Genetic algorithms*: following the introduction of the concepts of genetic algorithms in [76], the canonical genetic algorithm shown in [77] did not preserve the best solutions during the evolution of the algorithm, namely, the elitism principle was not applied. In the homogeneous version of the canonical genetic algorithm, the crossover and mutation probabilities always remain constant. For this homogeneous canonical genetic algorithm, there is no proof of convergence to the global optimum. However, better results have been obtained under the condition of ensuring the survival of the best individual with probability that is equal to unity (elitist selection). In this case, finite Markov chain analysis has been used to prove probabilistic convergence to the best solution in [78]. The proof that the *elitist* homogeneous canonical genetic algorithm converges *almost surely* to a population that has an optimum point in that it has been given in [79]. Subsequently, a number of conditions to ensure asymptotic convergence of genetic algorithms to the global optimum have been given in [80]. Conceptually, at each generation, there is a non-zero probability that a new individual reaches the global optimum due to the application of the genetic operators. As such, saving the best individual at each generation (in the elitist version)

and running the algorithm for an infinite number of generations guarantees that the global optimum can be reached. Further indications to extend the proof of almost sure convergence to the elitist non-homogeneous canonical genetic algorithm are provided in [81], by considering that the mutation and crossover probabilities are allowed to change during the evolution of the algorithm [82].

(2) *Simulated annealing*: a proof of convergence has been given in [83] for a particular class of algorithms, and the asymptotic convergence has been proven for the algorithm that is shown in [84]. Further results have been indicated in [85], showing convergence to the global optimum for continuous global optimization problems under specific conditions for the cooling schedule, the function under analysis, and the feasible set.

For multiobjective optimization problems, the proofs of convergence have been set up by introducing elitism, following the successful practice that was found for single-objective functions. For some multi-objective evolutionary algorithms, convergence proofs to the global optimum are provided in [86,87]. The asymptotic convergence analysis of Simulated Annealing, an Artificial Immune System and a General Evolutionary Algorithm (with any algorithm in which the transition probabilities use a uniform mutation rule) for multiobjective optimization problems, is shown in [88].

5. Discussion and Results on the Comparisons among Metaheuristic Algorithms

5.1. No Free Lunches?

Comparing different algorithms is a very challenging task. Unfortunately, many articles concerning metaheuristics applications in the power and energy systems area (as well as in other engineering fields) are underestimating the importance of this task, and propose simplistic comparison criteria and metrics, such as the best solution obtained, the evolution in time of the objective function improvement for a single run, or related criteria.

In the literature, there is wide discussion on the algorithm comparison aspects. One of the contributions that have opened an interesting debate is the one that introduced the No Free Lunch (NFL) theorem(s) [89]. These theorems state that "any two optimization algorithms are equivalent when their performance is averaged across all possible problems" [90]. Basically, the NFL theorems state that no optimization algorithm results in the best solutions for all problems. In other words, if a given algorithm performs better than another on a certain number of problems, then there should be a comparable number of problems in which the other algorithm outperforms the first one. However, if a given problem is considered, with its objective functions and constraints, some algorithms could perform better than others, especially when these algorithms are able to incorporate specific knowledge on the problem at hand. The debate includes contributions that argue the NFL theorems are of little relevance for the machine learning research [91], in which meta-learning can be used to gain experience on the performance of a number of applications of a learning system.

5.2. Comparisons among Metaheuristics

A recent contribution [92] has addressed comparison strategies and their mathematical properties in a systematic way. The numerical comparison between optimization algorithms consists of the selection of a set of algorithms and problems, the testing of the algorithms on the problems, the identification of comparison strategy, methods and metrics, the analysis of the outcomes that were obtained from applying the metrics, and the final determination of the results.

One of the main issues for setting up the comparisons is the definition of the overall scenario in which the comparison is carried out. The use of benchmarking methodologies, such as Black-Box Optimization Benchmarking (BBOB) discussed in [93], pointed out that reaching consensus on ranking the results from evaluations of individual problems is a crucial issue. It is then hard to provide a response to the question "which is the best algorithm to solve a given problem?" [94]. However, in some cases, a response should be given, as in the case of competitions launched among algorithms.

When testing a single (existing or new) algorithm, a set of algorithms that provide good results for similar problems are typically selected to carry out the comparison. This is one of the weak points that are encountered in the literature, especially when the choice of the benchmark problems is carried out by the authors without a clear and convincing criterion. A number of mathematical functions that can be used as standard benchmarks are available [2]. Some test problems have been defined in different contexts [95–97]. However, a systematic guide on how to select the set of problems is still missing. The hint that is given in [92] is to select the whole set of optimization problems in a given domain, and not only a partial set.

Moreover, for global optimization, there is no known mathematical optimal condition to be satisfied for stopping the search for all of the problems. Thereby, the computation time is generally taken as the common limit for stopping the algorithms. For deterministic algorithms, a typical comparison metric is the performance ratio [98]; namely, the ratio between the computation time of the algorithm and the minimum computation time of all algorithms applied to the same problem), from which the performance profile is obtained as the CDF of the performance ratio. Furthermore, the data profile [99] is based on the CDF of the problems that can be solved (by reaching at least a certain target in the solution) with a number of function evaluations not higher than a given limit. With non-deterministic algorithms, the concepts used in the definition of performance profiles and data profiles could be exploited. Comparisons can be carried out by implementing all of the algorithms on the same computer and running them for the same computational time. The quality of the algorithms can then be determined by calculating the percentage of the best solutions, averaged over a given number of executions of each algorithm [13]. The series of the best solutions obtained during the execution of the algorithm in the given time is typically considered for applying a performance metric [29,92].

When the constraints are directly imposed, it may happen that unfeasible solutions are generated during the calculations. These solutions have to be skipped or eliminated from the search. In this case, less useful solutions will be found by running the solver under analysis for a given number of times, worsening the performance indicator. Similar considerations apply when the solutions to be compared are subject to further conditions, for example, in order to satisfy the *N*-1 security conditions, as requested in [100] for a transmission expansion planning problem.

Indications on the comparison strategies are provided in [92], basically identifying pairwise comparison between algorithms (with the variants one-plays-all, generally used to check a new algorithm, and all-play-all or "round-robin"), and multi-algorithm comparison, both being used in many contexts. For multi-algorithm comparison, statistical aggregations, such as the cumulative distribution function, are often used.

The comparison methods can be partitioned into static (with evaluation of the best solution, mean, standard deviation, or other statistic outcomes), dynamic ranking (which considers the succession of the best values or static rankings during the time), and the cumulative distribution functions (considered at different times during the solution process). The latter type of comparison has become increasingly interesting, also representing the confidence intervals [101,102].

Liu et al. [92] defined the problem of finding the best algorithm as a voting system, in which the algorithms are the candidates, the problems are the votes, and an algorithm performs better than the others if it exhibits better performance on more problems. However, they found the existence of the so-called "cycle ranking" or Condorcet paradox, namely, it may happen that different algorithms are winners for different problems, and it is not possible to conclude which algorithm is better overall. In practice, taking three algorithms A, B, and C, it may happen that, for different problems, A is better than B, B is better than C, and C is better than A. The same concept is shown in [29] by indicating that the relation between the solvers is non-transitive, namely, if algorithm A is better than algorithm B for some problems, and algorithm B is better than algorithm C, this does not imply that algorithm A is better than algorithm C. Another paradox that is shown in [92] is the so-called "survival of the fittest".

In this case, the winner can be different by using different comparison strategies. The probability of occurrence of the two paradoxes is calculated based on the NFL assumption.

5.3. Which Superiority?

Superiority is the term widely used to indicate that a given algorithm performs better than others. However, the way to assess superiority is often stated in a trivial and misleading way. In particular, the use of simple performance indicators, such as the best solution, the average value of the solutions, or the standard deviation of the solutions, makes it possible to exacerbate the paradoxes that are indicated in the above section. The main reason is the lack of robustness of these indicators, especially the ones that are based on a single occurrence (such as the best value) that could be found occasionally during the execution of the algorithm (or even with a "lucky" choice of the initial population). The continuous production of articles claiming that the algorithm used is superior with respect to a selected set of other algorithms is mostly due to the use of these simple performance indicators. A synthesis of the mechanism that leads to this continuous production of articles has been provided in [29], by introducing a *perpetual motion* conceptual scheme, from which it is clear that it is not possible to find a formal and rigorous way to stop the production of articles.

The only way to reduce the number of articles with questionable superiority is to introduce more robust statistics-based indicators for comparing the algorithms with each other. A number of non-parametric statistical tests are summarized in [103]. Another example is the Optimization Performance Indicator based on Stochastic Dominance (*OPISD*) indicator provided in [29], by considering the first-order stochastic dominance concepts [104] with the approach indicated in [105]. Starting from the CDFs of the solutions that were obtained from a set of algorithms run on the same problem (a qualitative example with three algorithms is shown in Figure 4), the *OPISD* indicator is formulated by considering a reference CDF together with the CDFs obtained from the given algorithms, calculating for each algorithm the area A between the corresponding CDF and the reference CDF (Figure 5). Subsequently, the indicator is defined as $OPISD = (1 + A)^{-1}$. In this way, the algorithm with the smallest area is the one that exhibits better performance. From Figure 5, algorithm 2 is the one that exhibits the best performance.
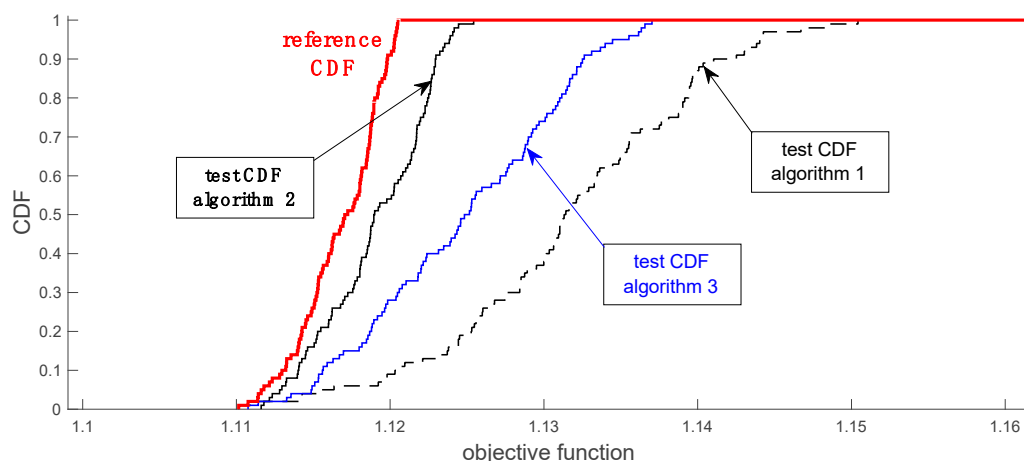

Figure 4. Determination of the reference CDF for the calculation of the Optimization Performance Indicator that is based on Stochastic Dominance (*OPISD*) indicator without knowing the global optimum.

The reference CDF is constructed in different ways, depending on whether the global optimum is known or not. If the global optimum is known, then the reference CDF is equal to zero for values that are lower than the global optimum, and then it jumps to unity at the global optimum. This enables absolute comparisons among the algorithms, even though the global optimum can only be known in a few cases of relatively small systems, and "good" algorithms should always reach the global optimum for these small systems. If the global optimum is not known, the reference CDF is determined

by starting from a given number of best solutions obtained from any of the algorithms used for comparison. In this case, only a relative comparison on the set of algorithms under analysis is possible, as the reference CDF changes each time.
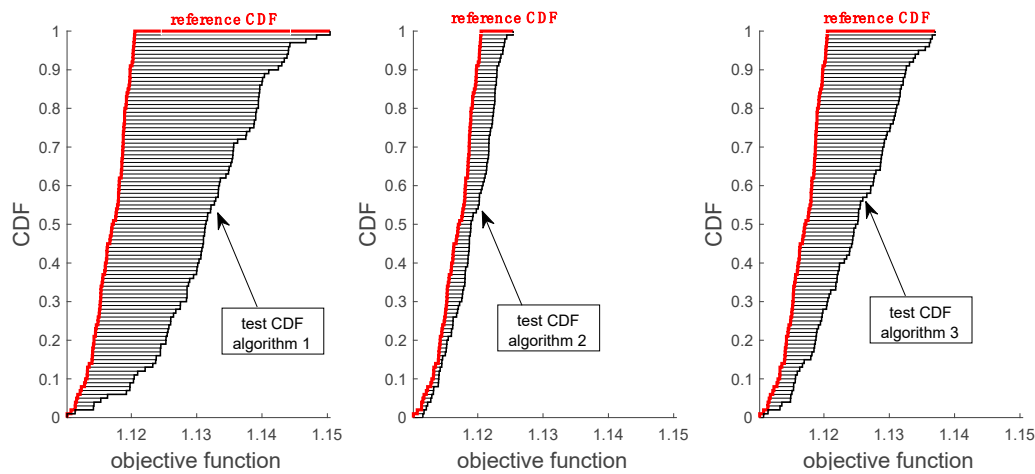

Figure 5. Determination of the areas *A* for the calculation of the *OPISD* indicator.

6. Hybridization of the Metaheuristics

The various metaheuristics have advantages and disadvantages, usually analyzed in terms of exploration and exploitation characteristics [106], and of contributions to improve the local search. In order to enhance the performance of the algorithms, one of the ways has been the formulation of hybrid optimization methods. The main types of hybridizations can be summarized, as follows:

(a)　combinations of different heuristics; and,

(b)　combinations of metaheuristics with exact methods.

Successful strategies have been found from the combined use of a heuristic that carries out an extensive search in the solution space, together with a method that is suitable for local search. A practical example is the Evolutionary Particle Swarm Optimization (EPSO), in which an evolutionary model is used together with a particle movement operator to formulate a *self-adaptive* algorithm [107]. Another useful tool is the Lévy flights [108], which is used to mitigate the issue of early convergence of metaheuristics [109,110] and obtain a better balance between exploration and exploitation. A further example of hybridization is the Differential Evolutionary Particle Swarm Optimization Algorithm [111]—the winner of the smart grid competition at the IEEE Congress on Evolutionary Computation/The Genetic and Evolutionary Computation Conference in 2019.

Depending on the problem under analysis, a useful practice can be the combination of a metaheuristic that aimed at providing a contribution to the global search, and of an exact method of proven effectiveness to perform a local search. In many other cases, hybridizations have no special meaning and they could be only aimed at producing further articles that contribute to the 'rush to heuristics'.

7. Multi-Objective Formulations

Multi-objective or many-objective optimization (see Section 1) consider more than one objective. The solutions obtained for the individual objectives are relevant when *conflicting* objectives appear. Optimization with conflicting objectives does not search only the optimal values of the individual objectives, but also identifies the compromise solutions as feasible alternatives for decision making.

Figure 6 shows the concept of dominated solution for a case with two objective functions f_1 and f_2 to be minimized. More generally, Figure 7 reports some qualitative examples of locations of the dominated solutions when the objective functions have to be maximized or minimized. Moreover,

the dominated solutions can be assigned different *levels of dominance*, for assisting their ranking when they are used within solution algorithms. Figure 8 shows an example with four levels of dominance (where the first level is the best-known Pareto front, some points of which could be located on the *true* Pareto front). Fuzzy-based dominance degrees have also been defined in [112].
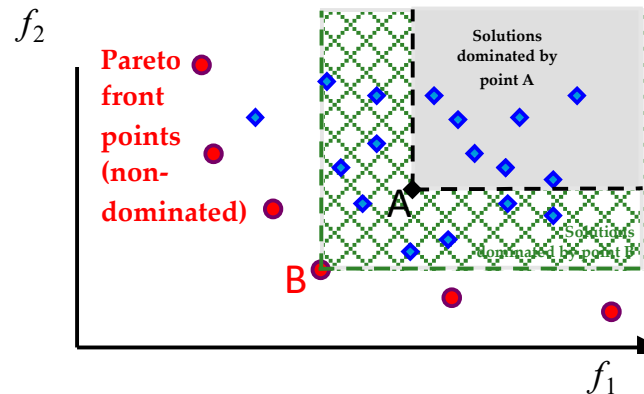

Figure 6. Concept of Pareto dominance. The functions f_1 and f_2 are minimized.
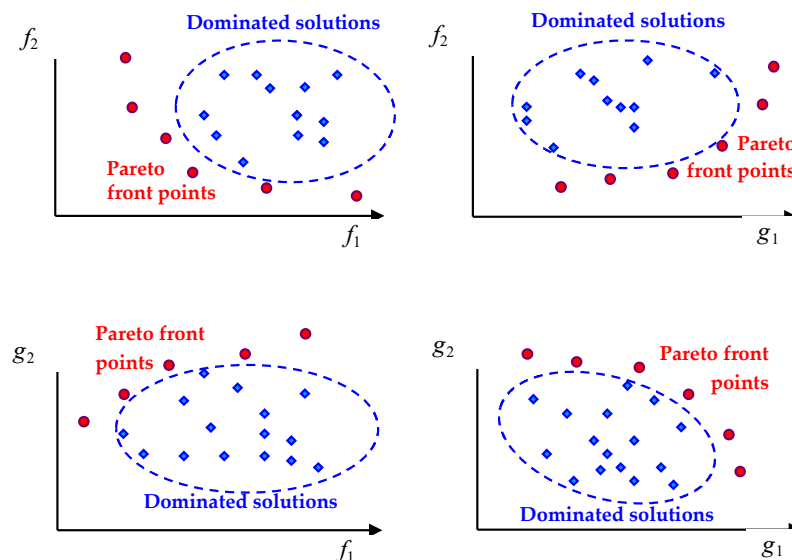

Figure 7. Pareto front points and dominated solutions for objective functions minimized (f_1 and f_2), or maximized (g_1 and g_2).
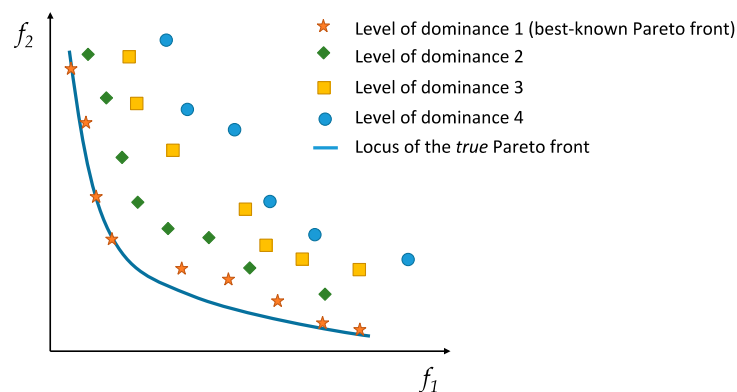

Figure 8. Levels of dominance. The functions f_1 and f_2 are minimized.

7.1. Techniques for Pareto Front Calculation

Some deterministic approaches are available. If the Pareto front is convex, then the *weighted sum* of the individual objectives can be used to track the points of the Pareto front. Otherwise, other methods have to be exploited. The first type of possibility is to use the *ε-constrained method* [113], which considers an individual objective as the target to be optimized and sets for all the other objectives a limit that is expressed by a threshold ε, and then progressively reduces the threshold and upgrades the set of non-dominated solutions. Reduction to a single objective function is also carried out by using the goal programming approach [114]. *Fuzzy logic-based* approaches are also available [115].

However, multi-objective optimization with Pareto front construction and assessment is a successful field of application of metaheuristics. In particular, the direct construction through metaheuristic approaches is an iterative process, in which, at each iteration, multiple solutions are generated, and the solution set is then reduced to only maintain the non-dominated solutions. The dominated solutions are arranged into levels of dominance to enable wider comparisons.

Practically, for most of the many metaheuristic algorithms formulated for solving single-objective optimization problems, there is also the corresponding multi-objective optimization algorithm. As such, a list of metaheuristics for multi-objective optimization is not provided here. Only a few algorithms are mentioned because of their key historical and practical relevance: the Strength Pareto Evolutionary Approach (SPEA2 [116]), the Pareto Archived Evolution Strategy (PAES [117]), and two versions of Non-dominated Sorting Genetic Algorithm, namely NSGA-II [118] and NSGA-III [119].

7.2. No Free Lunches and Comparisons among Algorithms

The discussion about the NFL theorem(s) is also valid for multi-objective optimization. When considering all of the problems under analysis, according to the NFL theorem(s) all of the algorithms outperform the other algorithms for some problems. However, for multi-objective optimization Corne & Knowles [120] showed that the NFL does not generally apply when absolute performance metrics are used. This means that some multi-objective approaches can be better than others to construct the Pareto front. In fact, the best-known Pareto front should be sufficiently wide in order to contain a number of points relatively far from each other, and an approach that only finds a set of points concentrated in a limited region can be considered to be less efficient than another one that provides more dispersed compromise solutions.

On these bases, developing comparison metrics or quality indicators for multi-objective optimization algorithms is a challenging but worthwhile task. Some principles that are indicated in [11] for the construction of effective multi-objective comparison metrics include:

(a)   The minimization of the distance between the best-known Pareto front and the true optimal Pareto front (when the latter is known).
(b)   The presence of a distribution of the solutions as uniform as possible.
(c)   For each objective, the presence of a wide range of values in the best-known Pareto front.

For comparison purposes, it is convenient to represent the quality of each Pareto front obtained using a multi-objective optimization algorithm by using a scalar value (a real number). This number can be the average distance between the points located onto the Pareto front under analysis and the closest points of the best-known Pareto front. A survey of the indicators proposed in the literature is provided in [121,122]. Other indicators are assessed with a chi-square-like deviation measure, in order to exploit the Pareto front diversity [121,123]. An Inverted Generational Distance indicator has been proposed in [124] to deal with the tradeoff between proximity and diversity preservation of the solutions in multi-objective optimization problems.

Furthermore, the hyper-volume indicator [8,10,121] has been used both for performance assessment and guiding the search in various hyper-volume-based evolutionary optimizers [125]. For a problem in which all of the objectives are minimized, and all of the points that form the Pareto front are positive, the hyper-volume can be calculated by setting up the maximum value for each objective, in such a way

to obtain a regular polyhedron. Subsequently, given the Pareto front, the hyper-volume is determined by calculating the volume that is found starting from the origin of the axes and is limited by the Pareto front or by the regular polyhedron surfaces. If the objective function has negative values, then the same concepts apply by using translation operators in such a way that the Pareto front is located inside the regular polyhedron.

A weighted hyper-volume indicator has been introduced in [126], including weight functions to express user preferences. The selection of the weight functions and transformation of the user preferences into weight functions has been addressed in [127]. While the idea of exploiting the hyper-volume calculation is interesting and based on geometric considerations, the determination of efficient algorithms to determine the hyper-volume when the number of dimensions increases is an open research field. For many objectives, a diversity metric has been proposed in [128] by summing up the dissimilarity of the solutions to the rest of the population.

For power and energy systems, many multi-objective problems are defined with two or three objectives. In these cases, the hyper-volume can be calculated from the available methods [9,129]. In this case, it is also possible to extend previous results regarding the comparison among metaheuristics. Let us consider a number of objectives to be minimized. Following the concepts introduced in [29], when multiple metaheuristic algorithms have to be compared on a given multi-objective optimization problem, it is possible to determine the best-known Pareto front that results from all of the executions of the algorithm for a given time. Subsequently, the comparison of all the Pareto fronts, which result from the various methods, provide a quality indicator given by the hyper-volume included between the Pareto front under analysis and the best-known Pareto front. In this case, each solution is represented by using a scalar value. Lower values of this scalar value mean better quality of the result. The CDF of these scalar values can be constructed, and it is considered as the reference CDF for *OPISD* calculation. The comparison between the CDFs of the individual metaheuristic algorithms and the reference CDF provides the area *A* to be used for *OPISD* calculation.

For comparing multi-objective metaheuristic algorithms, the definition of a suitable set of *test functions* is needed as a benchmark. For this purpose, classical test functions have been introduced in [11], with the ZDT functions that contain two objectives, chosen to represent different cases with specific features. Further test functions have been introduced with nine DTLZ functions in [130], as well as in [131]. However, for power and energy systems, these benchmarks do not take into account the typical constraints that appear in specific problems and, as such, an algorithm that shows good performance on these mathematical benchmarks could behave with poor performance on these problems. A lack of dedicated benchmarking for a wide set of power and energy problems does not enable the scientists in the power and energy domain presenting sufficiently broad results on the metaheuristic algorithm performance.

7.3. Multi-Objective Solution Ranking

The last, but not less, important aspect concerning the multi-objective optimization outcomes is the possible *ranking* of the solutions that were determined by numerical calculations, in order to assist the decision-maker in the task of identifying the preferable solution. The methods that are available for this task require, in some way, to obtain the opinion of the expert to express preferences about the objectives considered. These methods belong to multi-criteria decision-making, where the criteria coincide with the objectives under consideration here. Some tools widely adopted are the Analytic Hierarchy Process [132], in which a nine-point scale quantifies the relative preferences between pairs of objectives, and the overall feasibility of the process is confirmed if an appropriately defined consistency criterion is satisfied. Furthermore, in the Ordered Weighted Averaging approach [133] the weights are ordered according to their relative importance, and a procedure that is driven by a single parameter is set up by using a transformation function that modifies the weighted values of the objectives. The Technique of Order Preference by Similarity to Ideal Solution (TOPSIS) method is based on the evaluation of the objectives, depending on their distance to reference (ideal) points [134,135].

Further methods, such as ELECTRE [136] and PROMETHEE [137], are based on comparing pairs of weights. Other methods have been formulated using fuzzy logic-based tools [112,130]. For example, for a transmission expansion planning problem, the fuzzy logic-based tools are used in [138,139], and in [58], where the rank of each solution is directly established in each Pareto front.

The reduction of personal judgment from the decision-maker is sometimes desired, especially when the problem is highly technical and the decision-maker has different qualifications. Moreover, in some cases, the introduction of automatic procedures to determine the relative importance of the objectives is needed, in particular when the judgment is included in an iterative process and the relative importance has to be established many times when considering the variation of the objective function values.

These cases may typically occur when dealing with technical aspects in the power and energy systems domain. For example, a criterion for comparing non-dominated solutions based on power systems concepts has been introduced in [140], in which the Incremental Cost Benefit ratio has been defined by calculating the ratio between the congestion cost reduction with respect to the base case and the investment referring to the solution considered. The automatic creation of the entries of the pair comparison matrices for the AHP approach has been introduced in [54] for a distribution system reconfiguration problem, using an affine function that maps the objective function values onto the Saaty interval from 1 to 9.

8. Discussion on the Effectiveness of Metaheuristic-Based Optimization: Pitfalls and Inappropriate Statements

In their articles aimed at applying metaheuristic optimization methods, various authors include inappropriate statements on the effectiveness of the methods used. These statements are also one of the main reasons of the rejection of many papers sent to scientific journals or conferences. The most significant (and sometimes common) situations are recalled in this section, with corresponding discussion on whether more appropriate solutions could be adopted.

8.1. On reaching the Global Optimum

Some articles report that the scope of the analysis is "to reach the global optimum". This statement is *never correct* for a heuristic run on a large-scale problem. In the previous sections, it has been clarified that no metaheuristic can guarantee to find the global optimum for any finite time or number of iterations. At most, asymptotic convergence to the global optimum has been proven for some metaheuristics; namely, by executing an infinite number of iterations [141]. In practice, *no heuristic* is able to guarantee that the global optimum can be reached in a finite time, as it would be needed for engineering problems.

8.2. Adaptive Stop Criterion

As a consequence of the previous point, how to decide when to stop the execution of a metaheuristic algorithm becomes a crucial issue. Setting up a sound stop criterion (or termination criterion) is needed. Quite surprisingly, many algorithms that are used in available publications consider the maximum number of iterations N_{max} as the sole stop criterion. However, this choice is generally inappropriate. In fact, two different issues could occur [142]:

(1) *early stopping*, in which the execution could be stopped when the evolution of the *objective function* is still providing *significant* improvements (Figure 9a); or,

(2) *late stopping*, in which the execution could be stopped when the solution had no variations (or no significant changes, in a milder version) for many of the last iterations (Figure 9b); in this case, the last part of the execution, with many constant values, is unnecessary and could have been avoided.
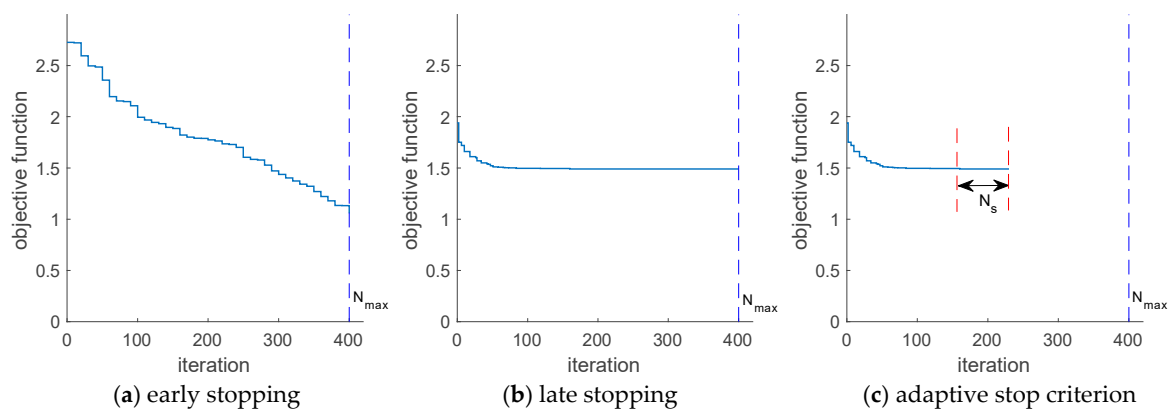
Figure 9. Effectiveness of the adaptive stop criterion for objective functions to be minimized. N_{max}: fixed maximum number of iterations. N_{s}: number of successive iterations without objective function improvements.

Improvements in the objective function(s) could appear at any time. However, the identification of a sound stop criterion is important to use the computation time in the best way.

The definition of an *adaptive stop criterion* is the most appropriate solution to the above-indicated issues. In the adaptive stop criterion (sometimes indicated as *stagnation* criterion), the algorithm terminates when no change occurs in the best objective function found so far, after a given number N_{s} of successive iterations of the algorithm (Figure 9c). In this case, early stopping and late stopping are both avoided. The number of successive iterations is a user-defined parameter that can also be chosen based on the experience on the variability of the objective function for specific problems. The maximum number of iterations, being set to a very high value, could remain as a *last-resource* stop criterion to trap possible unlimited executions.

If the metaheuristic algorithm is run for performance comparisons (see Section 4 (2)), then the stop criterion (equal for all algorithms) could become the computation time. Additionally, in this case, the use of the maximum number of iterations is not needed. Thereby, the maximum number of iterations should not be used as the primary criterion to terminate the execution of a metaheuristic algorithm.

8.3. Not Only Metaheuristics

A common but inappropriate trend found in many papers is to use a metaheuristic method or variant and only compare it with other metaheuristics. While this trend is highly questionable because of the 'rush to heuristics' issue (see Section 2), the availability of many alternative methods beyond metaheuristics has to be considered. Indeed, for specific problems, there can be many algorithms of different types, and the rationale for using a "new" metaheuristic has to be explicitly stated. It is not sufficient to compare a few algorithms that are chosen at random. In general, metaheuristic algorithms are relatively simple to be implemented (at least in their standard versions and for problems that do not require setting up equality or inequality constraints referring to complex structures). For this reason, it could be easier to perform comparisons among metaheuristics taken from existing libraries or implemented by the same authors. However, a fair comparison requires choosing a set of metaheuristics that have exhibited good performances for a number of related problems. The absence of well-established benchmarks leaves the situation somehow confused.

8.4. The Importance of Fast Convergence

Some contributions give key importance to achieving *fast convergence* (in terms of the *number of iterations* employed to provide the algorithm's result) and they show the case with the fastest convergence as an example of the "superiority" of the proposed algorithm. An example of statement for this case is "The convergence of the proposed metaheuristic is better than for other methods".

What in reality happens is that, during the iterative process, the objective function of the proposed metaheuristic improves in *fewer iterations* than for other methods tested.

First of all, the computation time for solving one iteration for a given algorithm is generally different with respect to other methods. As such, just considering the number of iterations is rather meaningless. Moreover, the improvements that occur in fewer iterations *could not mean anything*. In general, fast convergence can be achieved from good (i.e., lucky) choices of the initial case for single-update methods, or of one or more individuals of the initial population in population-based methods. In the extreme case, if the initial choice happens to contain the global optimum (without knowing it neither in advance nor having any way to be sure of that), the convergence to the best solution is immediate. However, by no way, this could mean that the algorithm used is better than another one.

8.5. How Many Executions?

A typical drawback of many articles is the limited number of executions performed with the metaheuristic method. The number of executions has to be sufficiently high, indicatively not less than 100, in order to reach statistical significance for the utilized approach. However, if the problem can be solved thousands of times within a reasonable computation time, it would be even better. As an obvious corollary, if the iterative process has been run only *once* for each method, a comparison among *one-shot* cases is not significant to reach any conclusion.

When a comparison that is based on a given computation time limit is carried out (e.g., within a competition), the number of executions will be driven by the computation time limit. In that case, writing an efficient and fast programming code would be clearly relevant.

8.6. Avoiding Fallacies

During the testing of a metaheuristic algorithm on a well-known problem, a possible issue is that "the results obtained are impressively better than those appearing in the literature". What could happen in this case is that the proposed heuristic has provided largely better results with respect to the same or similar solutions found in other literature references from various other methods run on the same problem. The warning for such a case is that this kind of result may be *suspect*. The possible causes of this situation have to be searched on possible issues in the modeling used. For example, the considered system could not be exactly the same as in the other references. Another possibility is that the constraints may have some differences among the compared methods (in terms of number, formulation, or threshold values applied). A possible hint, in this case, is to solve the optimization on a small system for which the global optimum is known: if the results differ, then there could be something wrong in the data used or in the implementation of the algorithm.

9. Conclusions

This paper has presented a contribution to the discussion on the use of metaheuristic methods in the solution of global optimization problems. Starting from the evidence that the number of published articles on metaheuristic applications in the power and energy systems domain is increasing with an impressively high trend, some questions on the reasons of this 'rush to heuristics' have been addressed. It has emerged that there is a lack of dedicated benchmark for global optimization in the power and energy systems domain, as well as a lack of statistically significant and robust comparisons among the outcomes of the metaheuristic approach. The existing benchmarks that are used in the evolutionary computation domain are not always sufficient for representing the specific aspects encountered in power and energy systems problems. Therefore, dedicated customizations may be needed in order to execute some metaheuristic algorithms on these problems. The construction of specific benchmarks for given problems in the power and energy systems area is a challenging topic for future research. The basic literature concerning comparison metrics for single-objective and multi-objective problems solved with a metaheuristic approach has been reviewed. In many articles, the metrics used for comparisons are too weak to reach sound conclusions on the effectiveness of the proposed algorithm.

Sometimes, the superior performance of the method used over a selected set of other methods is incorrectly claimed on the basis of a few results. A set of underlying principles has been identified, in order to explain the characteristics of the metaheuristic algorithms in a systematic way and discover possible similarities among the many algorithms proposed. The underlying principles identified could serve to provide a categorization of the metaheuristic algorithms. This would require considerable work to process over one hundred of metaheuristics already available (whose number is growing rapidly) and discuss the results in a global context.

The discussion on the most effective use of metaheuristic algorithms is a challenging subject. Some pitfalls and inappropriate statements, which are sometimes found in literature contributions, have been highlighted. While there is no formal way to stop the proliferation of articles that propose new applications, variants or hybridizations of metaheuristics, it is the authors' idea that systematic indications on how to avoid these pitfalls may be useful for the researchers. For this purpose, some *guidelines* for preparing sound contributions on the application of metaheuristic algorithms to power and energy system problems (but also useful for other application fields) are summarized in the following points:

(1) Consider the size and type of the *optimization* problem. If the problem can be solved with exhaustive search or by using exact tools in reasonable computation time, then applying a metaheuristic algorithm is useless.

(2) For *proposing* a new metaheuristic method or variant referring to the mechanism of the method (including hybridizations) and not to direct customization to specific needs of problems in the power and energy area, send the contribution to journals referring to the *soft computing and evolutionary computation domains*, where specialists can carry out a conceptual and practical validation.

(3) Specify the *information coding* in details.

(4) Clarify the *relations* between the heuristic operators and the variables of the specific problem (not only describing general-purpose tools).

(5) Illustrate the treatment of the *constraints* explicitly. Customization of the classical version of a metaheuristic algorithm could be needed, and the rationale and effectiveness of the customization have to be specifically addressed. Discuss how to keep the *constraints* enforced during the evolution of the computational process.

(6) Explain *parameter settings* and values, possibly carrying out *sensitivity* analyses.

(7) Choose the algorithms to compare for obtaining a reasonably strong *benchmark* (not only selecting a few other metaheuristics for which relevant results have not been clearly stated – with an accurate look at the state of the art). Avoid the mere 'rush to heuristics'! General benchmarks defined with mathematical test functions could not be detailed enough for representing the specific issues that appear in power and energy systems domain.

(8) Implement the *adaptive stop criterion* (using the maximum number of iterations only as a secondary criterion to terminate the execution).

(9) Implement the algorithms to be compared and execute them with the *same data and problem formulation*, in order to avoid possible variations with respect to the data and problem definition used in other articles.

(10) Show the *statistics* of the results obtained on test systems and/or real networks, based on a significant number of executions (not only one execution) and on the use of appropriately robust statistical indicators.

(11) Use the correct *terminology*, avoiding declaring the superiority of an algorithm on the basis of the results obtained on a specific problem only and with limited testing.

The previous indications refer to the application side and they are directed to the scientific communities that adopt metaheuristics for solving problems that are not solvable with exact methods, or that can be solved efficiently with metaheuristic algorithms. However, the concept of "efficient"

solution has not been clearly explained yet. Substantial work is needed, and it is in progress in the evolutionary computation community, in the direction of improving the design of metaheuristics and developing modeling languages and efficient general-purpose solvers [1]. This direction does not include the 'rush to heuristics', which is just wasting a lot of energies of many researchers in a useless and non-concluding 'perpetual motion' of production of contributions with possible improper developments and incorrect attempts to declare an inexistent 'superiority'.

Author Contributions: Conceptualization, G.C. and A.M.; Methodology, G.C. and A.M.; Software, G.C. and A.M.; Validation, G.C. and A.M.; Formal Analysis, G.C. and A.M.; Investigation, G.C. and A.M.; Resources, G.C. and A.M.; Writing—Original Draft Preparation, G.C. and A.M.; Writing—Review & Editing, G.C. and A.M.; Visualization, G.C. and A.M. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

AC	Alternating Current
AHP	Analytic Hierarchy Process
AMP	Adaptive Memory Programming
BBOB	Black-Box Optimization Benchmarking
CDF	Cumulative Distribution Function
DC	Direct Current
DEEPSO	Differential EPSO
DG	Distributed Generation
DSP	Distribution System Planning
DSR	Distribution System Reconfiguration
ED	Economic Dispatch
EPSO	Evolutionary Particle Swarm Optimization
FACTS	Flexible AC Transmission Systems
GMS	Generation unit Maintenance Scheduling
IEEE	The Institute of Electrical and Electronics Engineers
LGF	Load and Generation Forecasting
MS	Maintenance Scheduling
NCUC	Network-Constrained Unit Commitment
NFL	No Free Lunch
NSGA	Non-dominated Sorting Genetic Algorithm
OPF	Optimal Power Flow
OPISD	Optimization Performance Indicator based on Stochastic Dominance
PAES	Pareto Archived Evolution Strategy
SCOPF	Security Constrained Optimal Power Flow
SPEA	Strength Pareto Evolutionary Approach
TMS	Transmission Maintenance Scheduling
TOPSIS	Technique of Order Preference by Similarity to Ideal Solution
TNEP	Transmission Network Expansion Planning
UC	Unit Commitment

Appendix A

Table A1. Over one hundred heuristics used in the power and energy systems domain.

Heuristic	Year	Heuristic	Year
Ant colony optimization [17]	1991	Ant-lion optimizer [143]	2015
Artificial algae algorithm [144]	2015	Artificial bee colony [145]	2007
Artificial cooperative search algorithm [146]	2013	Artificial ecosystem-based optimization [147]	2019

Table A1. *Cont.*

Heuristic	Year	Heuristic	Year
Artificial fish swarm algorithm [148]	2018	Artificial immune system [149]	1986
Atom search optimization [150]	2019	Auction-based algorithm [151]	2014
Bacterial foraging [152]	2002	Backtracking search algorithm [153]	2013
Bat-inspired algorithm [154]	2010	Bayesian optimization algorithm [155]	1999
Big-bang big-crunch [156]	2013	Biogeography based optimization [157]	2011
Brainstorming process algorithm [158]	2011	Cat swarm optimization [159]	2006
Chaos optimal algorithm [160]	2010	Charged system search [161]	2010
Chemical reaction based optimization [162]	2010	Civilized swarm optimization [163]	2003
Clonal selection algorithm-Clonalg [164]	2002	Cohort Intelligence [165]	2013
Coral reefs optimization [166]	2014	Covariance matrix adaptation evolution strategy [167]	2003
Colliding bodies optimization [168]	2014	Coyote optimization algorithm [169]	2018
Crisscross optimization algorithm [170]	2014	Crow search algorithm [171]	2016
Cuckoo search algorithm [109]	2009	Cultural algorithm [172]	1994
Dendritic cell algorithm [173]	2005	Differential evolution [174]	1997
Differential search algorithm [175]	2013	Diffusion limited aggregation [176]	1981
Dolphin echolocation algorithm [177]	2013	Dragonfly algorithm [178]	2016
Eagle strategy [179]	2010	Electromagnetism-like mechanism [180]	2012
Election algorithm [181]	2015	Elephant herd optimization [182]	2015
Equilibrium optimizer [183]	2020	Estimation of distribution algorithms [184]	1996
Evolutionary algorithms [185]	1966	Evolution strategies [186]	1971
Farmland fertility optimization [187]	2018	Firefly algorithm [188]	2010
Firework algorithm [189]	2010	Flower pollination algorithm [190]	2012
Front-based yin-yang-pair optimization [191]	2016	Fruit fly optimization [192]	2012
Galactic swarm optimization [193]	2016	Galaxy-based search algorithm [194]	2011
Gases Brownian motion [195]	2013	Genetic algorithms [77]	1975
Glowworm swarm optimization [196]	2005	Grasshoppers optimization [197]	2017
Gravitational search algorithm [198]	2009	Greedy randomized adaptive search procedures [199]	1989
Grenade explosion method [200]	2010	Grey wolf optimization [201]	2014
Group search optimization [202]	2006	Harmony search algorithm [203]	2013
Harris hawks optimizer [204]	2019	Imperialist competitive algorithm [205]	2007
Intelligent water drops [206]	2007	Invasive weed optimization [207]	2006
Ions motion optimization algorithm [208]	2015	Jaya algorithm [209]	2016
Kinetic gas molecule optimization [210]	2014	Krill herd algorithm [211]	2012
League championship algorithm [212]	2014	Lion optimization algorithm [213]	2016
Manta ray foraging optimization [214]	2020	Marine predators algorithm [215]	2020
Marriage in honey bees optimization [216]	2001	Mean-variance mapping optimization [217]	2010
Melody search algorithm [218]	2013	Memetic algorithms [219]	1989
Mine blast algorithm [220]	2013	Monarch butterfly optimization [221]	2015
Monkey algorithm [222]	2007	Moth-flame optimization [223]	2015
Optics inspired optimization [224]	2014	Particle swarm optimization [225]	1995
Pigeon inspired optimization [226]	2014	Population extremal optimization [227]	2001
Plant growth simulation [228]	2005	Predator–prey optimization [229]	2006
Quantum-inspired evolutionary algorithm [230]	1995	Quick group search optimizer [231]	2010
Radial movement optimization [232]	2014	Rain-fall optimization [233]	2017
Ray optimization algorithm [234]	2012	River formation dynamics [235]	2007
Salp swarm algorithm [236]	2017	Simulated annealing [14]	1983
Scatter search [237]	1977	Seagull optimization [238]	2019
Seeker optimization algorithm [239]	2006	Shuffled frog leaping algorithm [240]	2006
Sine-cosine algorithm [241]	2016	Slime mould optimization algorithm [242]	2008
Soccer league competition algorithm [243]	2014	Social group optimization [244]	2016
Social spider algorithm [245]	2015	Squirrel search algorithm [246]	2019
Stochastic fractal search [247]	2015	Symbiotic organisms search [248]	2014
Tabu search (*) [249]	1989	Teaching-learning-based optimization [250]	2011
Tree-seed algorithm [251]	2015	Variable neighborhood search [252]	1997
Virus colony search [253]	2016	Volleyball premier league [254]	2018
Vortex search algorithm [255]	2015	Water cycle algorithm [256]	2012
Water waves optimization [257]	2015	Weighted superposition attraction [258]	2016
Whale optimization algorithm [259]	2016	Wind driven optimization [260]	2010
Wolf search algorithm [261]	2012		

(*) Not based on random number extraction.

References

1. Sörensen, K.; Sevaux, M.; Glover, F. A History of Metaheuristics. In *Handbook of Heuristics*; Martí, R., Pardalos, P., Resende, M., Eds.; Springer: Cham, Switzerland, 2018.
2. Salcedo-Sanz, S. Modern meta-heuristics based on nonlinear physics processes: A review of models and design procedures. *Phys. Rep.* **2016**, *655*, 1–70. [CrossRef]
3. Boussaïd, I.; Lepagnot, J.; Siarry, P. A survey on optimization metaheuristics. *Inf. Sci.* **2013**, *237*, 82–117. [CrossRef]
4. Zedadra, O.; Guerrieri, A.; Jouandeau, N.; Spezzano, G.; Seridi, H.; Fortino, G. Swarm intelligence-based algorithms within IoT-based systems: A review. *J. Parallel Distrib. Comput.* **2018**, *122*, 173–187. [CrossRef]
5. Dokeroglu, T.; Sevinc, E.; Kucukyilmaz, T.; Cosar, A. A survey on new generation metaheuristic algorithms. *Comput. Ind. Eng.* **2019**, *137*, 106040. [CrossRef]
6. Li, K.; Deb, K.; Zhang, Q.; Kwong, S. An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition. *IEEE Trans. Evol. Comput.* **2014**, *19*, 694–716. [CrossRef]
7. Ishibuchi, H.; Tsukamoto, N.; Nojima, Y. Evolutionary many-objective optimization: A short review. In Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–6 June 2008; pp. 2419–2426. [CrossRef]
8. Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **1999**, *3*, 257–271. [CrossRef]
9. Guerreiro, A.; Fonseca, C.M. Computing and Updating Hypervolume Contributions in Up to Four Dimensions. *IEEE Trans. Evol. Comput.* **2017**, *22*, 449–463. [CrossRef]
10. While, L.; Hingston, P.; Barone, L.; Huband, S. A faster algorithm for calculating hypervolume. *IEEE Trans. Evol. Comput.* **2006**, *10*, 29–38. [CrossRef]
11. Zitzler, E.; Deb, K.; Thiele, L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evol. Comput.* **2000**, *8*, 173–195. [CrossRef]
12. Sörensen, K. Metaheuristics-the metaphor exposed. *Int. Trans. Oper. Res.* **2013**, *22*, 3–18. [CrossRef]
13. Taillard, É.D.; Gambardella, L.M.; Gendreau, M.; Potvin, J.-Y. Adaptive memory programming: A unified view of metaheuristics. *Eur. J. Oper. Res.* **2001**, *135*, 1–16. [CrossRef]
14. Kirkpatrick, S.; Gelatt, J.C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *World Scientific Lecture Notes in Physics* **1986**, *220*, 339–348. [CrossRef]
15. Batrinu, F.; Carpaneto, E.; Chicco, G. A unified scheme for testing alternative techniques for distribution system minimum loss reconfiguration. In Proceedings of the 2005 International Conference on Future Power Systems, Amsterdam, The Netherlands, 16–18 November 2005; p. 6.
16. Chicco, G. Ant colony system-based applications to electrical distribution system optimization. In *Ant Colony Optimization—Methods and Applications*; Ostfeld, A., Ed.; InTech: Rijeka, Croatia, February 2011; Chapter 16; pp. 237–262.
17. Dorigo, M.; Maniezzo, V.; Colorni, A. Positive Feedback as a Search Strategy. Politecnico di Milano: Dipartimento di Elettronica. *Technical Report (91-016)*, April 1991.
18. Blum, C.; Dorigo, M. The Hyper-Cube Framework for Ant Colony Optimization. *IEEE Trans. Syst. Man, Cybern. Part B* **2004**, *34*, 1161–1172. [CrossRef] [PubMed]
19. Chen, G.; Low, C.P.; Yang, Z. Preserving and Exploiting Genetic Diversity in Evolutionary Programming Algorithms. *IEEE Trans. Evol. Comput.* **2009**, *13*, 661–673. [CrossRef]
20. Črepinšek, M.; Liu, S.-H.; Mernik, M. Exploration and exploitation in evolutionary algorithms. *ACM Comput. Surv.* **2013**, *45*, 1–33. [CrossRef]
21. Bäck, T. Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In Proceedings of the First IEEE Conference on Evolutionary Computation, Orlando, FL, USA, 27–29 June 1994; pp. 57–62.
22. Mitsos, A.; Najman, J.; Kevrekidis, I.G. Optimal deterministic algorithm generation. *J. Glob. Optim.* **2018**, *71*, 891–913. [CrossRef]
23. Bain, S.; Thornton, J.; Sattar, A. Methods of Automatic Algorithm Generation. In *Computer Vision*; Springer: Berlin, Germany, 2004; pp. 144–153.
24. Koza, J.R. *Genetic Programming II: Automatic Discovery of Reusable Subprograms*; The MIT Press: Cambridge, MA, USA, 1994.

25. Burke, E.K.; Hyde, M.; Kendall, G.; Ochoa, G.; Özcan, E.; Woodward, J.R. A Classification of Hyper-heuristic Approaches. *Stoch. Program.* **2010**, *146*, 449–468. [CrossRef]

26. Drake, J.H.; Kheiri, A.; Özcan, E.; Burke, E.K. Recent advances in selection hyper-heuristics. *Eur. J. Oper. Res.* **2020**, *285*, 405–428. [CrossRef]

27. Lee, K.Y.; El-Sharkawi, M.A. *Modern Heuristic Optimization Techniques*; Wiley: Hoboken, NJ, USA, 2008.

28. Lee, K.Y.; Vale, Z.A. *Applications of Modern Heuristic Optimization Methods in Power and Energy Systems*; Wiley: Hoboken, NJ, USA, 2020.

29. Chicco, G.; Mazza, A. Heuristic optimization of electrical energy systems: Refined metrics to compare the solutions. *Sustain. Energy Grids Netw.* **2019**, *17*, 100197. [CrossRef]

30. Del Valle, Y.; Venayagamoorthy, G.; Mohagheghi, S.; Hernandez, J.-C.; Harley, R.G. Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. *IEEE Trans. Evol. Comput.* **2008**, *12*, 171–195. [CrossRef]

31. Zheng, Q.; Wang, J.; Liu, A.L. Stochastic Optimization for Unit Commitment—A Review. *IEEE Trans. Power Syst.* **2014**, *30*, 1913–1924. [CrossRef]

32. Tejada-Arango, D.A.; Lumbreras, S.; Sánchez-Martín, P.; Ramos, A. Which Unit-Commitment Formulation is Best? A Comparison Framework. *IEEE Trans. Power Syst.* **2020**, *35*, 2926–2936. [CrossRef]

33. Kazarlis, S.; Bakirtzis, A.G.; Petridis, V. A genetic algorithm solution to the unit commitment problem. *IEEE Trans. Power Syst.* **1996**, *11*, 83–92. [CrossRef]

34. Swarup, K.; Yamashiro, S.; Swarup, K.S. Unit commitment solution methodology using genetic algorithm. *IEEE Trans. Power Syst.* **2002**, *17*, 87–91. [CrossRef]

35. Muralikrishnan, N.; Jebaraj, L.; Rajan, C.C.A. A Comprehensive Review on Evolutionary Optimization Techniques Applied for Unit Commitment Problem. *IEEE Access* **2020**, *8*, 132980–133014. [CrossRef]

36. Walters, D.; Sheble, G. Genetic algorithm solution of economic dispatch with valve point loading. *IEEE Trans. Power Syst.* **1993**, *8*, 1325–1332. [CrossRef]

37. Conejo, A.J.; Baringo, L. *Unit Commitment and Economic Dispatch*; Springer Science and Business Media LLC: Berlin, Germany, 2017; pp. 197–232.

38. Liang, Z.-X.; Glover, J. A zoom feature for a dynamic programming solution to economic dispatch including transmission losses. *IEEE Trans. Power Syst.* **1992**, *7*, 544–550. [CrossRef]

39. Irisarri, G.; Kimball, L.; Clements, K.; Bagchi, A.; Davis, P. Economic dispatch with network and ramping constraints via interior point methods. *IEEE Trans. Power Syst.* **1998**, *13*, 236–242. [CrossRef]

40. Yan, X.; Quintana, V. An efficient predictor-corrector interior point algorithm for security-constrained economic dispatch. *IEEE Trans. Power Syst.* **1997**, *12*, 803–810. [CrossRef]

41. Fu, Y.; Liu, M.; Li, L. Multiobjective Stochastic Economic Dispatch with Variable Wind Generation Using Scenario-Based Decomposition and Asynchronous Block Iteration. *IEEE Trans. Sustain. Energy* **2015**, *7*, 139–149. [CrossRef]

42. Bakirtzis, A.G. Genetic algorithm solution to the economic dispatch problem. *IEE Proc.-Gener. Transm. Distrib.* **1994**, *141*, 377. [CrossRef]

43. Abbas, G.; Gu, J.; Farooq, U.; Asad, M.U.; El-Hawary, M.E. Solution of an Economic Dispatch Problem Through Particle Swarm Optimization: A Detailed Survey—Part I. *IEEE Access* **2017**, *5*, 15105–15141. [CrossRef]

44. Duvvuru, N.; Swarup, K.S. A Hybrid Interior Point Assisted Differential Evolution Algorithm for Economic Dispatch. *IEEE Trans. Power Syst.* **2010**, *26*, 541–549. [CrossRef]

45. Abbas, G.; Gu, J.; Farooq, U.; Raza, A.; Asad, M.U.; El-Hawary, M.E. Solution of an Economic Dispatch Problem Through Particle Swarm Optimization: A Detailed Survey–Part II. *IEEE Access* **2017**, *5*, 24426–24445. [CrossRef]

46. Tang, Y.; Dvijotham, K.; Low, S. Real-Time Optimal Power Flow. *IEEE Trans. Smart Grid* **2017**, *8*, 2963–2973. [CrossRef]

47. Momoh, J.; Zhu, J. Improved interior point method for OPF problems. *IEEE Trans. Power Syst.* **1999**, *14*, 1114–1120. [CrossRef]

48. Niu, M.; Wan, C.; Xu, Z. A review on applications of heuristic optimization algorithms for optimal power flow in modern power systems. *J. Mod. Power Syst. Clean Energy* **2014**, *2*, 289–297. [CrossRef]

49. Carvalho, L.; Loureiro, F.; Sumaili, J.; Keko, H.; Miranda, V.; Gil Marcelino, C.; Wanner, E. Statistical tuning of DEEPSO soft constraints in the Security Constrained Optimal Power Flow problem. In Proceedings of the 2015 18th International Conference on Intelligent System Application to Power Systems (ISAP), Porto, Portugal, 11–17 September 2015; pp. 1–7.

50. Carpaneto, E.; Chicco, G. Distribution system minimum loss reconfiguration in the Hyper-Cube Ant Colony Optimization framework. *Electr. Power Syst. Res.* **2008**, *78*, 2037–2045. [CrossRef]

51. Tomoiagă, B.; Chindris, M.; Sumper, A.; Sudrià-Andreu, A.; Villafafila-Robles, R. Pareto Optimal Reconfiguration of Power Distribution Systems Using a Genetic Algorithm Based on NSGA-II. *Energies* **2013**, *6*, 1439–1455. [CrossRef]

52. Andrei, H.; Chicco, G. Identification of the Radial Configurations Extracted From the Weakly Meshed Structures of Electrical Distribution Systems. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2008**, *55*, 1149–1158. [CrossRef]

53. Carreno, E.; Romero, R.; Padilha-Feltrin, A. An Efficient Codification to Solve Distribution Network Reconfiguration for Loss Reduction Problem. *IEEE Trans. Power Syst.* **2008**, *23*, 1542–1551. [CrossRef]

54. Mazza, A.; Chicco, G.; Russo, A. Optimal multi-objective distribution system reconfiguration with multi criteria decision making-based solution ranking and enhanced genetic operators. *Int. J. Electr. Power Energy Syst.* **2014**, *54*, 255–267. [CrossRef]

55. Arroyo, J.M.; Alguacil, N.; Carrion, M. A Risk-Based Approach for Transmission Network Expansion Planning Under Deliberate Outages. *IEEE Trans. Power Syst.* **2010**, *25*, 1759–1766. [CrossRef]

56. Latorre, G.; Cruz, R.; Areiza, J.; Villegas, A. A classification of publications and models on transmission expansion planning. *IEEE Trans. Power Syst.* **2003**, *18*, 938–946. [CrossRef]

57. Hu, Y.; Bie, Z.; Ding, T.; Lin, Y. An NSGA-II based multi-objective optimization for combined gas and electricity network expansion planning. *Appl. Energy* **2016**, *167*, 280–293. [CrossRef]

58. Jadidoleslam, M.; Ebrahimi, A.; Latify, M.A. Probabilistic transmission expansion planning to maximize the integration of wind power. *Renew. Energy* **2017**, *114*, 866–878. [CrossRef]

59. Villasana, R.; Garver, L.L.; Salon, S.J. Transmission network planning using linear programming. *IEEE Trans. Power Appar. Syst.* **1985**, *104*, 349–356. [CrossRef]

60. Fletcher, R.H.; Strunz, K. Optimal Distribution System Horizon Planning–Part I: Formulation. *IEEE Trans. Power Syst.* **2007**, *22*, 791–799. [CrossRef]

61. Vahidinasab, V.; Tabarzadi, M.; Arasteh, H.; Alizadeh, M.I.; Beigi, M.M.; Sheikhzadeh, H.R.; Mehran, K.; Sepasian, M.S.; Mohammadbeygi, M. Overview of Electric Energy Distribution Networks Expansion Planning. *IEEE Access* **2020**, *8*, 34750–34769. [CrossRef]

62. Georgilakis, P.S.; Hatziargyriou, N.D. A review of power distribution planning in the modern power systems era: Models, methods and future research. *Electr. Power Syst. Res.* **2015**, *121*, 89–100. [CrossRef]

63. Grond, M.; Morren, J.; Slootweg, H. Requirements for advanced decision support tools in future distribution network planning. In Proceedings of the 22nd International Conference and Exhibition on Electricity Distribution (CIRED 2013), Stockholm, Sweden, 10–13 June 2013; p. 1046.

64. Mishra, D.K.; Ghadi, M.J.; Azizivahed, A.; Li, L.; Zhang, J. A review on resilience studies in active distribution systems. *Renew. Sustain. Energy Rev.* **2021**, *135*, 110201. [CrossRef]

65. Venkateswaran, B.; Saini, D.K.; Sharma, M. Approaches for optimal planning of the energy storage units in distribution network and their impacts on system resiliency–A review. *CSEE J. Power Energy Syst.* **2020**, in press.

66. Strantzali, E.; Aravossis, K. Decision making in renewable energy investments: A review. *Renew. Sustain. Energy Rev.* **2016**, *55*, 885–898. [CrossRef]

67. Khator, S.; Leung, L. Power distribution planning: A review of models and issues. *IEEE Trans. Power Syst.* **1997**, *12*, 1151–1159. [CrossRef]

68. Hippert, H.S.; Pedreira, C.E.; Souza, R.C. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Trans. Power Syst.* **2001**, *16*, 44–55. [CrossRef]

69. Al Mamun, A.; Sohel, M.; Mohammad, N.; Sunny, M.S.H.; Dipta, D.R.; Hossain, E. A Comprehensive Review of the Load Forecasting Techniques Using Single and Hybrid Predictive Models. *IEEE Access* **2020**, *8*, 134911–134939. [CrossRef]

70. Akhter, M.N.; Mekhilef, S.; Mokhlis, H.; Shah, N.M.; Saad, M. Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques. *IET Renew. Power Gener.* **2019**, *13*, 1009–1023. [CrossRef]

71. Li, S.; Goel, L.; Wang, P. An ensemble approach for short-term load forecasting by extreme learning machine. *Appl. Energy* **2016**, *170*, 22–29. [CrossRef]

72. Froger, A.; Gendreau, M.; Mendoza, J.E.; Pinson, E.; Rousseau, L.-M. Maintenance scheduling in the electricity industry: A literature review. *Eur. J. Oper. Res.* **2016**, *251*, 695–706. [CrossRef]

73. Aik, K.C.; Lai, L.L.; Lee, K.Y.; Lu, H.; Park, J.-B.; Song, Y.-H.; Srinivasan, D.; Vlachogiannis, J.G.; Yu, I.K. Chapter 15 Applications to Power System Scheduling. In *Modern Heuristic Optimization Techniques*; Lee, K.Y., El-Sharkawi, M.A., Eds.; Wiley: Hoboken, NJ, USA, 2008.

74. Kim, H.; Hayashi, Y.; Nara, K. An algorithm for thermal unit maintenance scheduling through combined use of GA, SA and TS. *IEEE Trans. Power Syst.* **1997**, *12*, 329–335. [CrossRef]

75. Pardalos, P.M.; Romeijn, H.E.; Tuy, H. Recent developments and trends in global optimization. *J. Comput. Appl. Math.* **2000**, *124*, 209–228. [CrossRef]

76. Holland, J.H. Outline for a Logical Theory of Adaptive Systems. *J. ACM* **1962**, *9*, 297–314. [CrossRef]

77. Holland, J.H. *Adaptation in Natural and Artificial Systems*; MIT Press: Cambridge, MA, USA, 1992.

78. Eiben, A.E.; Aarts, E.H.L.; Van Hee, K.M. Global convergence of genetic algorithms: A markov chain analysis. In *Computer Vision*; Springer: Berlin, Germany, 1991; Volume 496, pp. 3–12.

79. Rudolph, G. Convergence analysis of canonical genetic algorithms. *IEEE Trans. Neural Netw.* **1994**, *5*, 96–101. [CrossRef] [PubMed]

80. Cerf, R. Asymptotic convergence of genetic algorithms. *Adv. Appl. Probab.* **1998**, *30*, 521–550. [CrossRef]

81. Rojas Cruz, J.A.; Pereira, A.G.C. The elitist non-homogeneous geneticalgorithm: Almost sure convergence. *Stat. Probab. Lett.* **2013**, *83*, 2179–2185. [CrossRef]

82. Campos, V.E.M.; Pereira, A.G.C.; Rojas Cruz, J.A. Modeling the genetic algorithm by a non-homogeneous Markov chain: Weak and strong ergodicity. *Theory Probab. Appl.* **2013**, *57*, 144–151. [CrossRef]

83. Bélisle, C.J.P. Convergence theorems for a class of simulated annealing algorithms on R^d. *J. Appl. Probab.* **1992**, *29*, 885–895. [CrossRef]

84. Romeijn, H.E.; Smith, R.L. Simulated annealing for constrained global optimization. *J. Glob. Optim.* **1994**, *5*, 101–126. [CrossRef]

85. Locatelli, M. Convergence properties of simulated annealing for continuous global optimization. *J. Appl. Probab.* **1996**, *33*, 1127–1140. [CrossRef]

86. Rudolph, G. On a multi-objective evolutionary algorithm and its convergence to the Pareto set. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (Cat No 98TH8360), Anchorage, AK, USA, 4–9 May 1998; ICEC-98. pp. 511–516.

87. Rudolph, G.; Agapie, A. Convergence properties of some multi-objective evolutionary algorithms. In Proceedings of the 2000 Congress on Evolutionary Computation CEC00 (Cat. No.00TH8512), La Jolla, CA, USA, 16–19 July 2000; Volume 2, pp. 1010–1016.

88. Villalobos-Arias, M.; Coello, C.A.C.; Hernández-Lerma, O. Asymptotic Convergence of Some Metaheuristics Used for Multiobjective Optimization. In *Computer Vision*; Springer: Berlin, Germany, 2005; Volume 3469, pp. 95–111.

89. Wolpert, D.; Macready, W. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]

90. Wolpert, D.; Macready, W. Coevolutionary Free Lunches. *IEEE Trans. Evol. Comput.* **2005**, *9*, 721–735. [CrossRef]

91. Giraud-Carrier, C.; Provost, F. Toward a justification of meta-learning: Is the no free lunch theorem a show-stopper. In Proceedings of the ICML-2005 Workshop on Meta-learning, Bonn, Germany, 7–11 August 2005; pp. 12–19.

92. Liu, Q.; Gehrlein, W.V.; Wang, L.; Yan, Y.; Cao, Y.; Chen, W.; Li, Y. Paradoxes in Numerical Comparison of Optimization Algorithms. *IEEE Trans. Evol. Comput.* **2019**, *24*, 777–791. [CrossRef]

93. Mersmann, O.; Preuss, M.; Trautmann, H.; Bischl, B.; Weihs, C. Analyzing the BBOB Results by Means of Benchmarking Concepts. *Evol. Comput.* **2015**, *23*, 161–185. [CrossRef] [PubMed]

94. Bartz-Beielstein, T.; Chiarandini, M.; Paquete, L.; Preuss, M. *Experimental Methods for the Analysis of Optimization Algorithms*; Springer: Berlin/Heidelberg, Germany, 2010.

95. Gaviano, M.; Kvasov, D.; Lera, D.; Sergeyev, Y.D. Algorithm 829: Software for generation of classes of test functions with known local and global minima for global optimization. *ACM Trans. Math. Softw.* **2003**, *9*, 469–480. [CrossRef]

96. Hansen, N.; Auger, A.; Ros, R.; Finck, S.; Pošík, P. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB. In Proceedings of the 12th Annual Conference Comp on Genetic and Evolutionary Computation—GECCO '10, New York, NY, USA, 7 July 2010; pp. 1689–1696.

97. Liang, J.J.; Qu, B.Y.; Suganthan, P.N. Problem definitions and evaluation criteria for the CEC 2013 special session and competition on real-parameter optimization. *Tech. Rep.* **2012**, *12*, 281–295.

98. Dolan, E.D.; Moré, J.J. Benchmarking optimization software with performance profiles. *Math. Program.* **2002**, *91*, 201–213. [CrossRef]

99. Moré, J.J.; Wild, S.M. Benchmarking Derivative-Free Optimization Algorithms. *SIAM J. Optim.* **2009**, *20*, 172–191. [CrossRef]

100. Wang, Y.; Cheng, H.; Wang, C.; Hu, Z.; Yao, L.; Ma, Z.; Zhu, Z. Pareto optimality-based multi-objective transmission planning considering transmission congestion. *Electr. Power Syst. Res.* **2008**, *78*, 1619–1626. [CrossRef]

101. Liu, Q.; Chen, W.-N.; Deng, J.D.; Gu, T.; Zhang, H.; Yu, Z.; Zhang, J. Benchmarking Stochastic Algorithms for Global Optimization Problems by Visualizing Confidence Intervals. *IEEE Trans. Cybern.* **2017**, *47*, 2924–2937. [CrossRef]

102. Doerr, C.; Wang, H.; Ye, F.; van Rijn, S.; Bäck, T. IOHprofiler: A Benchmarking and Profiling Tool for Iterative Optimization Heuristics. *arxiv* **2018**, arXiv:1810.05281. Available online: https://arxiv.org/abs/1810.05281 (accessed on 29 September 2020).

103. Derrac, J.; García, S.; Molina, D.; Herrera, F.; Molina, D. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [CrossRef]

104. Hadar, J.; Russell, W.R. Stochastic dominance and diversification. *J. Econ. Theory* **1971**, *3*, 288–305. [CrossRef]

105. Chicco, G.; Mazza, A. Assessment of optimal distribution network reconfiguration results using stochastic dominance concepts. *Sustain. Energy Grids Netw.* **2017**, *9*, 75–79. [CrossRef]

106. Yang, X.-S.; Deb, S.; Fong, S. Metaheuristic Algorithms: Optimal Balance of Intensification and Diversification. *Appl. Math. Inf. Sci.* **2014**, *8*, 977–983. [CrossRef]

107. Miranda, V.; Fonseca, N. EPSO—Best-of-two-worlds meta-heuristic applied to power system problems. In Proceedings of the 2002 Congress on Evolutionary Computation CEC'02 (Cat. No.02TH8600), Honolulu, HI, USA, 12–17 May 2002; Volume 2, pp. 1080–1085.

108. Gutowski, M. Lévy flights as an underlying mechanism for global optimization algorithms. *ArXiv* **2001**, arXiv:0106003.

109. Yang, X.-S.; Deb, S. Cuckoo Search via Lévy flights. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 210–214. [CrossRef]

110. Zhang, X.; Xu, Y.; Yu, C.; Heidari, A.A.; Li, S.; Chen, H.; Li, C. Gaussian mutational chaotic fruit fly-built optimization and feature selection. *Expert Syst. Appl.* **2020**, *141*, 112976. [CrossRef]

111. Garcia-Guarin, J.; Rodriguez, D.; Alvarez, D.; Rivera, S.; Cortés, C.A.; Guzmán-Pardo, M.A.; Bretas, A.; Aguero, J.R.; Bretas, N. Smart Microgrids Operation Considering a Variable Neighborhood Search: The Differential Evolutionary Particle Swarm Optimization Algorithm. *Energies* **2019**, *12*, 3149. [CrossRef]

112. Benedict, S.; Vasudevan, V. Fuzzy-Pareto-dominance and its application in evolutionary multi-objective optimization. In Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization, Guanajuato, Mexico, 9–11 March 2005; Springer: Berlin, Germany; pp. 399–412.

113. Haimes, Y.; Lasdon, L.; Wismer, D. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Trans. Syst. Man Cybern.* **1971**, *1*, 296–297.

114. Contini, B. A Stochastic Approach to Goal Programming. *Oper. Res.* **1968**, *16*, 576–586. [CrossRef]

115. Deb, K. *Multi-Objective Optimization Using Evolutionary Algorithms*; John Wiley Sons, Ltd.: Hoboken, NJ, USA, 2001; p. 497.

116. Zitzler, E.; Laumanns, M.; Thiele, L. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*; Springer: Berlin, Germany, 2001; p. 103.

117. Knowles, J.; Corne, D. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evol. Comput.* **2000**, *8*, 149–172. [CrossRef]

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

119. Deb, K.; Jain, H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEE Trans. Evol. Comput.* **2013**, *18*, 577–601. [CrossRef]

120. Corne, D.; Knowles, J. No Free Lunch and Free Leftovers Theorems for Multiobjective Optimisation Problems. In *Computer Vision*; Springer: Berlin, Germany, 2003; Volume 2632, pp. 327–341.

121. Zitzler, E.; Thiele, L.; Laumanns, M.; Fonseca, C.M.; Da Fonseca, V.G. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans. Evol. Comput.* **2003**, *7*, 117–132. [CrossRef]

122. Zitzler, E.; Knowles, J.; Thiele, L. Quality Assessment of Pareto Set Approximations. In *Intelligent Robotics and Applications*; Springer: Berlin, Germany, 2008; pp. 373–404.

123. Srinivas, N.; Deb, K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolut. Comput.* **1994**, *2*, 221–248. [CrossRef]

124. Bosman, P.; Thierens, D. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans. Evol. Comput.* **2003**, *7*, 174–188. [CrossRef]

125. Auger, A.; Bader, J.; Brockhoff, D.; Zitzler, E. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theor. Comput. Sci.* **2012**, *425*, 75–103. [CrossRef]

126. Zitzler, E.; Brockhoff, D.; Thiele, L. The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators via Weighted Integration. In *Computer Vision*; Springer: Berlin, Germany, 2007; Volume 4403, pp. 862–876.

127. Brockhoff, D.; Bader, J.; Thiele, L.; Zitzler, E. Directed Multiobjective Optimization Based on the Weighted Hypervolume Indicator. *J. Multi-Criteria Decis. Anal.* **2013**, *20*, 291–317. [CrossRef]

128. Wang, H.; Jin, Y.; Yao, X. Diversity Assessment in Many-Objective Optimization. *IEEE Trans. Cybern.* **2017**, *47*, 1510–1522. [CrossRef]

129. Beume, N.; Fonseca, C.M.; López-Ibáñez, M.; Paquete, L.; Vahrenhold, J. On the Complexity of Computing the Hypervolume Indicator. *IEEE Trans. Evol. Comput.* **2009**, *13*, 1075–1082. [CrossRef]

130. Deb, K.; Thiele, L.; Laumanns, M.; Zitzler, E. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Kanpur, India: Kanpur Genetic Algorithms. In *Evolutionary Multiobjective Optimization*; KanGAL Report 2 001 001; Springer: London, UK, 2001; pp. 105–145.

131. Huband, S.; Hingston, P.; Barone, L.; While, L. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans. Evol. Comput.* **2006**, *10*, 477–506. [CrossRef]

132. Saaty, T.L. How to make a decision: The analytic hierarchy process. *Eur. J. Oper. Res.* **1990**, *48*, 9–26. [CrossRef]

133. Malczewski, J.; Chapman, T.; Flegel, C.; Walters, D.; Shrubsole, D.; Healy, M.A. GIS–Multicriteria Evaluation with Ordered Weighted Averaging (OWA): Case Study of Developing Watershed Management Strategies. *Environ. Plan. A Econ. Space* **2003**, *35*, 1769–1784. [CrossRef]

134. Hwang, C.L.; Yoon, K. *Multiple Attribute Decision Making. Methods and Applications: A State-of-the-Art Survey*; Springer: Berlin, Germany; New York, NY, USA, 1981.

135. Mazza, A.; Chicco, G. Application of TOPSIS in distribution systems multi-objective optimization. In Proceedings of the 9th World Energy System Conference, Suceava, Romania, 28–30 June 2012; pp. 625–633.

136. Roy, B. Classement et choix en présence de points de vue multiples. *Revue Française Inform. Rech. Opér.* **1968**, *2*, 57–75. [CrossRef]

137. Brans, J.P.; Mareschal, B. Promethee Methods. In *Multiple Criteria Decision Analysis: State of the Art Surveys, International Series in Operations Research & Management Science*; Springer: New York, NY, USA, 2005; Volume 78, pp. 163–195.

138. Moeini-Aghtaie, M.; Abbaspour, A.; Fotuhi-Firuzabad, M. Incorporating Large-Scale Distant Wind Farms in Probabilistic Transmission Expansion Planning—Part I: Theory and Algorithm. *IEEE Trans. Power Syst.* **2012**, *27*, 1585–1593. [CrossRef]

139. Chung, S.; Lee, K.K.; Chen, G.J.; Xie, J.D.; Tang, G.Q. Multi-objective transmission network planning by a hybrid GA approach with fuzzy decision analysis. *Elect. Power Energy Syst.* **2003**, *25*, 187–192. [CrossRef]

140. Maghouli, P.; Hosseini, S.H.; Buygi, M.; Shahidehpour, M. A Multi-Objective Framework for Transmission Expansion Planning in Deregulated Environments. *IEEE Trans. Power Syst.* **2009**, *24*, 1051–1061. [CrossRef]

141. Villalobos-Arias, M.; Coello, C.A.C.; Hernández-Lerma, O. Asymptotic convergence of a simulated annealing algorithm for multiobjective optimization problems. *Math. Methods Oper. Res.* **2006**, *64*, 353–362. [CrossRef]

142. Chicco, G.; Mazza, A.; Mazza, A. An overview of the probability-based methods for optimal electrical distribution system reconfiguration. In Proceedings of the 2013 4th International Symposium on Electrical and Electronics Engineering (ISEEE), Galati, Romania, 10–12 October 2013; pp. 1–10.

143. Mirjalili, S. The Ant Lion Optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [CrossRef]

144. Uymaz, S.A.; Tezel, G.; Yel, E. Artificial algae algorithm (AAA) for nonlinearglobal optimization. *Appl. Soft Comput.* **2015**, *31*, 153–171. [CrossRef]

145. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [CrossRef]

146. Civicioglu, P. Artificial cooperative search algorithm for numerical opti-mization problems. *Infor. Sci.* **2013**, *229*, 58–76. [CrossRef]

147. Zhao, W.; Wang, L.; Zhang, Z. Artificial ecosystem-based optimization: A novel nature-inspired meta-heuristic algorithm. *Neural Comput. Appl.* **2019**, *32*, 9383–9425. [CrossRef]

148. Xian, S.; Zhang, J.; Xiao, Y.; Pang, J. A novel fuzzy time series forecasting method based on the improved artificial fish swarm optimization algorithm. *Soft Comput.* **2017**, *22*, 3907–3917. [CrossRef]

149. Farmer, J.; Packard, N.H.; Perelson, A.S. The immune system, adaptation, and machine learning. *Phys. D Nonlinear Phenom.* **1986**, *22*, 187–204. [CrossRef]

150. Zhao, W.; Wang, L.; Zhang, Z. Knowledge-Based Systems Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowl.-Based Syst.* **2019**, *163*, 283–304. [CrossRef]

151. Binetti, G.; Davoudi, A.; Naso, D.; Turchiano, B.; Lewis, F.L. A Distributed Auction-Based Algorithm for the Nonconvex Economic Dispatch Problem. *IEEE Trans. Ind. Inform.* **2013**, *10*, 1124–1132. [CrossRef]

152. Passino, K. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control. Syst.* **2002**, *22*, 52–67. [CrossRef]

153. Civicioglu, P. Backtracking Search Optimization Algorithm for numerical optimization problems. *Appl. Math. Comput.* **2013**, *219*, 8121–8144. [CrossRef]

154. Yang, X.-S. A New Metaheuristic Bat-Inspired Algorithm. In *Studies in Computational Intelligence*; Springer: Berlin, Germany, 2010; Volume 284, pp. 65–74.

155. Pelikan, M.; Goldberg, D.E.; Cant-Paz, E. BOA: The Bayesian optimization algorithm. In Proceedings of the Genetic and Evolutionary Computation Conference—GECCO-99, Orlando, FL, USA, 13–17 July 1999; Volume I, pp. 525–532.

156. Sakthivel, S.; Pandiyan, S.A.; Marikani, S.; Selvi, S.K. Application of big-bang big-crunch algorithm for optimal power flow problems. *Int. J. Eng. Sci.* **2013**, *2*, 41–47.

157. Bhattacharya, A.; Chattopadhyay, P. Application of biogeography-based optimisation to solve different optimal power flow problems. *IET Gener. Transm. Distrib.* **2011**, *5*, 70–80. [CrossRef]

158. Yuhui, S. An optimization algorithm based on brainstorming process. *Int. J. Swarm Intell. Res.* **2011**, *2*, 35–62.

159. Chu, S.-C.; Tsai, P.-W.; Pan, J.-S. Cat swarm optimization. In *Trends in Artificial Intelligence (PRICAI 2006)*; Yang, Q., Webb, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4099, pp. 854–858.

160. Qu, G.; Cheng, H.; Yao, L.; Ma, Z.; Zhu, Z. Transmission surplus capacity based power transmission expansion planning. *Electr. Power Syst. Res.* **2010**, *80*, 19–27. [CrossRef]

161. Kaveh, A.; Talatahari, S. A novel heuristic optimization method: Charged system search. *Acta Mech.* **2010**, *213*, 267–289. [CrossRef]

162. Lam, A.Y.S.; Li, V.O.K. Chemical-Reaction-Inspired Metaheuristic for Optimization. *IEEE Trans. Evol. Comput.* **2010**, *14*, 381–399. [CrossRef]

163. Ray, T.; Liew, K. Society and civilization: An optimization algorithm based on the simulation of social behavior. *IEEE Trans. Evol. Comput.* **2003**, *7*, 386–396. [CrossRef]

164. De Castro, L.; Von Zuben, C.J.; De Castro, L.N. Learning and optimization using the clonal selection principle. *IEEE Trans. Evol. Comput.* **2002**, *6*, 239–251. [CrossRef]

165. Kulkarni, A.J.; Durugkar, I.P.; Kumar, M. Cohort Intelligence: A Self Supervised Learning Behavior. In Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, 13–16 October 2013; pp. 1396–1400.

166. Salcedo-Sanz, S.; Del Ser, J.; Landa-Torres, I.; Gil-López, S.; Portilla-Figueras, J.A. The Coral Reefs Optimization Algorithm: A Novel Metaheuristic for Efficiently Solving Optimization Problems. *Sci. World J.* **2014**, *2014*, 1–15. [CrossRef]

167. Hansen, N.; Müller, S.D.; Koumoutsakos, P. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evol. Comput.* **2003**, *11*, 1–18. [CrossRef]

168. Kaveh, A.; Mahdavi, V. Colliding bodies optimization: A novel meta-heuristic method. *Comput. Struct.* **2014**, *139*, 18–27. [CrossRef]

169. Pierezan, J.; Coelho, L.D.S. Coyote Optimization Algorithm: A New Metaheuristic for Global Optimization Problems. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brasil, 8–13 July 2018; pp. 1–8. [CrossRef]

170. Meng, A.-B.; Chen, Y.-C.; Yin, H.; Chen, S.-Z. Crisscross optimization algorithm and its application. *Knowl.-Based Syst.* **2014**, *67*, 218–229. [CrossRef]

171. Askarzadeh, A. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Comput. Struct.* **2016**, *169*, 1–12. [CrossRef]

172. Reynolds, R.G. An introduction to cultural algorithms. In Proceedings of the Third Annual Conference on Evolutionary Programming, San Diego, CA, USA, 24–26 February 1994; pp. 131–139.

173. Greensmith, J.; Aickelin, U.; Cayzer, S. Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection. In *Haptics: Science, Technology, Applications*; Springer Science and Business Media LLC: Berlin, Germany, 2005; Volume 3627, pp. 153–167.

174. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **1997**, *11*, 341–359. [CrossRef]

175. Civicioglu, P. Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. *Comput. Geosci.* **2012**, *46*, 229–247. [CrossRef]

176. Witten, T.A.; Sander, L.M. Diffusion-Limited Aggregation, a Kinetic Critical Phenomenon. *Phys. Rev. Lett.* **1981**, *47*, 1400–1403. [CrossRef]

177. Kaveh, A.; Farhoudi, N. A new optimization method: Dolphin echolocation. *Adv. Eng. Softw.* **2013**, *59*, 53–70. [CrossRef]

178. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2015**, *27*, 1053–1073. [CrossRef]

179. Yang, X.-S.; Deb, S. Eagle Strategy Using Lévy Walk and Firefly Algorithms for Stochastic Optimization. In *Studies in Computational Intelligence*; Springer: Berlin, Germany, 2010; Volume 284, pp. 101–111.

180. Cuevas, E.; Oliva, D.; Zaldivar, D.; Perez, M.A.; Sossa-Azuela, H.; Zaldívar, D. Circle detection using electro-magnetism optimization. *Inf. Sci.* **2012**, *182*, 40–55. [CrossRef]

181. Emami, H.; Derakhshan, F. Election algorithm: A new socio-politically inspired strategy. *AI Commun.* **2015**, *28*, 591–603. [CrossRef]

182. Wang, G.-G.; Deb, S.; Coelho, L.D.S. Elephant Herding Optimization. In Proceedings of the 3rd International Symposium on Computational and Business Intelligence, Bali, Indonesia, 7–9 December 2015; pp. 1–5. [CrossRef]

183. Faramarzi, A.; Heidarinejad, M.; Stephens, B.; Mirjalili, S. Equilibrium optimizer: A novel optimization algorithm. *Knowl.-Based Syst.* **2020**, *191*, 105190. [CrossRef]

184. Mühlenbein, H.; Paas, G. From recombination of genes to the estimation of distributions I. Binary parameters. In *Computer Vision*; Springer: Berlin, Germany, 1996; Volume 1141, pp. 178–187.

185. Fogel, D.B. *Artificial Intelligence through Simulated Evolution*; Wiley: New York, NY, USA, 2009; pp. 227–296.

186. Rechenberg, I. Evolutionsstrategie–Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution (in German). Ph.D. Thesis, Technical University of Berlin, Berlin, Germany, 1971.

187. Shayanfar, H.; Gharehchopogh, F.S. Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems. *Appl. Soft Comput.* **2018**, *71*, 728–746. [CrossRef]

188. Yang, X.-S. Firefly algorithm, stochastic test functions and design optimisation. *Int. J. Bio-Inspired Comput.* **2010**, *2*, 78. [CrossRef]

189. Tan, Y.; Zhu, Y. Fireworks Algorithm for Optimization. In *Computer Vision*; Springer: Berlin, Germany, 2010; Volume 6145, pp. 355–364.

190. Yang, X.-S. Flower Pollination Algorithm for Global Optimization. In *Computer Vision*; Springer: Berlin, Germany, 2012; Volume 7445, pp. 240–249.

191. Punnathanam, V.; Kotecha, P. Yin-Yang-pair Optimization: A novel light weight optimization algorithm. *Eng. Appl. Artif. Intell.* **2016**, *54*, 62–79. [CrossRef]

192. Pan, W.-T. A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example. *Knowl.-Based Syst.* **2012**, *26*, 69–74. [CrossRef]

193. Muthiah-Nakarajan, V.; Noel, M.M. Galactic Swarm Optimization: A new global optimization metaheuristic inspired by galactic motion. *Appl. Soft Comput.* **2016**, *38*, 771–787. [CrossRef]

194. Hosseini, H.S. Principal components analysis by the galaxy-based search al- gorithm: A novel metaheuristic for continuous optimisation. *Int. J. Comput. Sci. Eng.* **2011**, *6*, 132–140.

195. Abdechiri, M.; Meybodi, M.R.; Bahrami, H. Gases Brownian Motion Optimization: An Algorithm for Optimization (GBMO). *Appl. Soft Comput.* **2013**, *13*, 2932–2946. [CrossRef]

196. Krishnanand, K.; Ghose, D. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. In Proceedings of the 2005 IEEE Swarm Intelligence Symposium, Pasadena, CA, USA, 8–10 June 2005; pp. 84–91.

197. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper Optimisation Algorithm: Theory and application. *Adv. Eng. Softw.* **2017**, *105*, 30–47. [CrossRef]

198. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]

199. Feo, T.A.; Resende, M.G. A probabilistic heuristic for a computationally difficult set covering problem. *Oper. Res. Lett.* **1989**, *8*, 67–71. [CrossRef]

200. Ahrari, A.; Atai, A.A. Grenade Explosion Method—A novel tool for optimization of multimodal functions. *Appl. Soft Comput.* **2010**, *10*, 1132–1140. [CrossRef]

201. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Int. J. Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]

202. He, S.; Wu, Q.; Saunders, J. A Novel Group Search Optimizer Inspired by Animal Behavioural Ecology. In Proceedings of the 2006 IEEE International Conference on Evolutionary Computation, Vancouver, BC, Canada, 16–21 July 2006.

203. El-Abd, M. An improved global-best harmony search algorithm. *Appl. Math. Comput.* **2013**, *222*, 94–106. [CrossRef]

204. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Futur. Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]

205. Atashpaz-Gargari, E.; Lucas, C. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; pp. 4661–4667.

206. Hosseini, H.S. Shah Problem solving by intelligent water drops. In Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; pp. 3226–3231.

207. Mehrabian, A.; Lucas, C. A novel numerical optimization algorithm inspired from weed colonization. *Ecol. Inform.* **2006**, *1*, 355–366. [CrossRef]

208. Javidy, B.; Hatamlou, A.; Mirjalili, S. Ions motion algorithm for solving optimization problems. *Appl. Soft Comput.* **2015**, *32*, 72–79. [CrossRef]

209. Rao, R.V. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int. J. Ind. Eng. Comput.* **2016**, *7*, 19–34. [CrossRef]

210. Moein, S.; Logeswaran, R. KGMO: A swarm optimization algorithm based on the kinetic energy of gas molecules. *Inf. Sci.* **2014**, *275*, 127–144. [CrossRef]

211. Gandomi, A.H.; Alavi, A.H. Krill herd: A new bio-inspired optimization algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 4831–4845. [CrossRef]

212. Kashan, A.H. League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships. *Appl. Soft Comput.* **2014**, *16*, 171–200. [CrossRef]

213. Yazdani, M.; Jolai, F. Lion Optimization Algorithm (LOA): A nature-inspired metaheuristic algorithm. *J. Comput. Des. Eng.* **2015**, *3*, 24–36. [CrossRef]

214. Zhao, W.; Zhang, Z.; Wang, L. Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103300. [CrossRef]

215. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.* **2020**, *152*, 113377. [CrossRef]

216. Abbass, H.A. MBO: Marriage in honey bees optimisation: A haplometrosis polygynous swarming approach. In Proceedings of the Congress on Evolutionary Computation—CEC, Seoul, Korea, 27–30 May 2001; pp. 207–214.

217. Erlich, I.; Venayagamoorthy, G.K.; Worawat, N. A Mean-Variance Optimization algorithm. In Proceedings of the 2010 IEEE World Congress on Computational Intelligence, Barcelona, Spain, 18–23 July 2010.

218. Ashrafi, S.M.; Dariane, A.B. Performance evaluation of an improved harmonysearch algorithm for numerical optimization: Melody Search (MS). *Eng. Appl.Artif. Intel.* **2013**, *26*, 1301–1321. [CrossRef]

219. Moscato, P. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. In *Caltech Concurrent Computation Program (Report 826)*; California Institute of Technology: Pasadena, CA, USA, 1989; pp. 158–179.

220. Sadollah, A.; Bahreininejad, A.; Eskandar, H.; Hamdi, M. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Appl. Soft Comput.* **2013**, *13*, 2592–2612. [CrossRef]

221. Wang, G.-G.; Deb, S.; Cui, Z. Monarch butterfly optimization. *Neural Comput. Appl.* **2015**, *31*, 1995–2014. [CrossRef]

222. Zhao, R.; Tang, W. Monkey Algorithm for Global Numerical Optimization. *J. Uncertain Syst.* **2007**, *2*, 165–176.

223. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **2015**, *89*, 228–249. [CrossRef]

224. Kashan, A.H. A new metaheuristic for optimization: Optics inspired optimization (OIO). *Comput. Oper. Res.* **2015**, *55*, 99–125. [CrossRef]

225. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization. *Swarm Intell.* **2007**, *1*, 33–57. [CrossRef]

226. Duan, H.; Qiao, P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *Int. J. Intell. Comput. Cybern.* **2014**, *7*, 24–37. [CrossRef]

227. Boettcher, S.; Percus, A.G. Optimization with Extremal Dynamics. *Phys. Rev. Lett.* **2001**, *86*, 5211–5214. [CrossRef] [PubMed]

228. Li, T.; Wang, C.-F.; Wang, W.-B.; Su, W.-L. A global optimization bionics algorithm for solving integer programming-plant growth simulation algorithm. *Syst. Eng.-Theory Prac.* **2005**, *25*, 76–85.

229. Higashitani, M.; Ishigame, A.; Yasuda, K. Particle Swarm Optimization Considering the Concept of Predator-Prey Behavior. In Proceedings of the 2006 IEEE International Conference on Evolutionary Computation, Vancouver, BC, Canada, 16–21 July 2006; pp. 434–437.

230. Narayanan, A.; Moore, M. Quantum-inspired genetic algorithms. In Proceedings of the IEEE International Conference on Evolutionary Computation ICEC-96, Nagoya, Japan, 20–22 May 1996; pp. 61–66.

231. Guang, Q.; Feng, L.; Lijuan, L.; Lu, J.W.Z.; Leung, A.Y.T.; Iu, V.P.; Mok, K.M. A Quick Group Search Optimizer and Its Application to the Optimal Design of Double Layer Grid Shells. In Proceedings of the AIP Conference Proceedings, Hong Kong-Macau, China, 30 November–3 December 2009; AIP Publishing: College Park, MD, USA, 21 May 2010; Volume 1233, p. 718.

232. Rahmani, R.; Yusof, R. A new simple, fast and efficient algorithm for global optimization over continuous search-space problems. *Appl. Math. Comput.* **2014**, *248*, 287–300.

233. Kaboli, S.H.A.; Selvaraj, J.; Rahim, N. Rain-fall optimization algorithm: A population based algorithm for solving constrained optimization problems. *J. Comput. Sci.* **2017**, *19*, 31–42. [CrossRef]

234. Kaveh, A.; Khayatazad, M. A new meta-heuristic method: Ray Optimization. *Comput. Struct.* **2012**, *112–113*, 283–294. [CrossRef]

235. Rabanal, P.; Rodríguez, I.; Rubio, F. Using River Formation Dynamics to Design Heuristic Algorithms. In *Swarm, Evolutionary, and Memetic Computing*; Springer Science and Business Media LLC: Berlin, Germany, 2007; Volume 4618, pp. 163–177.

236. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]

237. Glover, F. Heuristics for Integer Programming Using Surrogate Constraints. *Decis. Sci.* **1977**, *8*, 156–166. [CrossRef]

238. Dhiman, G.; Kumar, D. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowl.-Based Syst.* **2019**, *165*, 169–196. [CrossRef]

239. Dai, C.; Chen, W.; Zhu, Y. Seeker Optimization Algorithm. In *Computational Intelligence and Security (CIS 2006)*; Wang, Y., Cheung, Y., Liu, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; Volume 1, pp. 225–229.

240. Eusuff, M.; Lansey, K.E.; Pasha, F. Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization. *Eng. Optim.* **2006**, *38*, 129–154. [CrossRef]

241. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [CrossRef]

242. Monismith, D.R.; Mayfield, B.E. Slime Mold as a model for numerical opti-mization. In Proceedings of the IEEE Swarm Intelligence Symposium, St. Louis, MO, USA, 21–23 September 2008.

243. Moosavian, N.; Roodsari, B.K. Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks. *Swarm Evol. Comput.* **2014**, *17*, 14–24. [CrossRef]

244. Satapathy, S.C.; Naik, A. Social group optimization (SGO): A new population evolutionary optimization technique. *Complex Intell. Syst.* **2016**, *2*, 173–203. [CrossRef]

245. Yu, J.J.; Li, V.O. A social spider algorithm for global optimization. *Appl. Soft Comput.* **2015**, *30*, 614–627. [CrossRef]

246. Jain, M.; Singh, V.; Rani, A. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. *Swarm Evol. Comput.* **2019**, *44*, 148–175. [CrossRef]

247. Salimi, H. Stochastic Fractal Search: A powerful metaheuristic algorithm. *Knowl.-Based Syst.* **2015**, *75*, 1–18. [CrossRef]

248. Cheng, M.-Y.; Prayogo, D. Symbiotic Organisms Search: A new metaheuristic optimization algorithm. *Comput. Struct.* **2014**, *139*, 98–112. [CrossRef]

249. Glover, F. Tabu Search—Part I. *ORSA J. Comput.* **1989**, *1*, 190–206. [CrossRef]

250. Rao, R.V.; Savsani, V.J.; Vakharia, D. Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Comput. Des.* **2011**, *43*, 303–315. [CrossRef]

251. Kıran, M.S.; Kiran, M.S. TSA: Tree-seed algorithm for continuous optimization. *Expert Syst. Appl.* **2015**, *42*, 6686–6698. [CrossRef]

252. Mladenović, N.; Hansen, P. Variable neighborhood search. *Comput. Oper. Res.* **1997**, *24*, 1097–1100. [CrossRef]

253. Li, M.D.; Zhao, H.; Weng, X.W.; Han, T. A novel nature-inspired algorithm for optimization: Virus colony search. *Adv. Eng. Softw.* **2016**, *92*, 65–88. [CrossRef]

254. Moghdani, R.; Salimifard, K. Volleyball Premier League Algorithm. *Appl. Soft Comput.* **2018**, *64*, 161–185. [CrossRef]

255. Dogan, B.; Ölmez, T. A new metaheuristic for numerical function optimization: Vortex Search algorithm. *Inf. Sci.* **2015**, *293*, 125–145. [CrossRef]

256. Eskandar, H.; Sadollah, A.; Bahreininejad, A.; Hamdi, M. Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* **2012**, *110*, 151–166. [CrossRef]

257. Zheng, Y.-J. Water wave optimization: A new nature-inspired metaheuristic. *Comput. Oper. Res.* **2015**, *55*, 1–11. [CrossRef]

258. Baykasoglu, A.; Senol, M.E. Combinatorial optimization via weighted superposition attraction. In Proceedings of the International Conference on Operations Research of the German Operation Society (GOR 2016), Hamburg, Germany, 30 August–12 September 2016.

259. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]

260. Bayraktar, Z.; Komurcu, M.; Werner, U.H. Wind Driven Optimization (WDO): A novel nature-inspired optimization algorithm and its application to electromagnetics. In Proceedings of the 2010 IEEE Antennas and Propagation Society International Symposium, Toronto, ON, Canada, 11–17 July 2010; pp. 1–4. [CrossRef]

261. Tang, R.; Fong, S.; Yang, X.-S.; Deb, S. Wolf search algorithm with ephemeral memory. In Proceedings of the Seventh International Conference on Digital Information Management (ICDIM 2012), Macau, China, 22–24 August 2012; pp. 165–172.