energies MDPI

Review

Review and Comparison of Genetic Algorithm and Particle Swarm Optimization in the Optimal Power Flow Problem

Georgios Papazoglou and Pandelis Biskas *

School of Electrical and Computer Engineering, Aristotle University of Thessaloniki, 541 24 Thessaloniki, Greece
* Correspondence: pbiskas@auth.gr; Tel.: +30-6973-841753

Abstract: Metaheuristic optimization techniques have successfully been used to solve the Optimal Power Flow (OPF) problem, addressing the shortcomings of mathematical optimization techniques. Two of the most popular metaheuristics are the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). The literature surrounding GA and PSO OPF is vast and not adequately organized. This work filled this gap by reviewing the most prominent works and analyzing the different traits of GA OPF works along seven axes, and of PSO OPF along four axes. Subsequently, cross-comparison between GA and PSO OPF works was undertaken, using the reported results of the reviewed works that use the IEEE 30-bus network to assess the performance and accuracy of each method. Where possible, the practices used in GA and PSO OPF were compared with literature suggestions from other domains. The cross-comparison aimed to act as a first step towards the standardization of GA and PSO OPF, as it can be used to draw preliminary conclusions regarding the tuning of hyper-parameters of GA and PSO OPF. The analysis of the cross-comparison results indicated that works using both GA and PSO OPF offer remarkable accuracy (with GA OPF having a slight edge) and that PSO OPF involves less computational burden.

Keywords: Optimal Power Flow; Genetic Algorithm; Particle Swarm Optimization; hyper-parameter tuning; metaheuristic optimization

check for updates

Citation: Papazoglou, G.; Biskas, P. Review and Comparison of Genetic Algorithm and Particle Swarm Optimization in the Optimal Power Flow Problem. *Energies* **2023**, *16*, 1152. https://doi.org/10.3390/en16031152

Academic Editors: Ali Mehrizi-Sani and José Gabriel Oliveira Pinto

Received: 19 November 2022
Revised: 15 January 2023
Accepted: 18 January 2023
Published: 20 January 2023

1. Introduction

1.1. Motivation

Since its introduction in 1962 [1], Optimal Power Flow (OPF) has been one of the most well-researched problems in power systems engineering [2]. OPF seeks to optimize a given objective (e.g., cost, reliability, planning, etc.) within a power system, while respecting Power Flow (PF) constraints, the system's physical constraints, and the operating limits of its components [2]. In its most general formulation, OPF is a non-linear (NL), non-convex, large-scale, highly constrained, multimodal optimization problem, which contains both continuous and discrete control variables [3].

OPF is a fundamental tool for efficient power system planning and operation. Prior to the deregulation of the electricity sector, OPF was routinely used in transmission-constrained economic dispatch and reactive power management [4]. Since the deregulation and emergence of competitive electricity markets, OPF has had many use cases, such as market clearing, scheduling, and dispatch, determination of Locational Marginal Prices (LMPs), calculation of available transfer capacity, etc. [2,4,5].

Recent trends in the industry indicate that the size and complexity of the OPF problems practitioners will need to solve in the near future will increase. These developments are crucial, as solving the OPF problem becomes increasingly harder as the size of the network and the complexity of its components increases [2]. In Europe, the trend towards integrating the European day-ahead electricity markets, as well as the shift towards quarter-hourly market time units, drastically increases the size of OPFs that need to be solved for market clearing, scheduling, and dispatch purposes [6]. Moreover, traditionally, OPF has included only the transmission network topology. With the proliferation of distributed and renewable generation, which is typically located in the distribution network, the distribution network topology may also have to be included in the OPF formulation, further increasing its size [7]. At the same time, the emergence of demand response [8] adds extra variables to the demand-side of OPF problems, increasing the complexity [2]. Additionally, due to aging equipment and the increase in demand, OPF is more tightly constrained, and thus a more realistic reflection of the network's physical constraints (i.e., its non-linearities and non-convexities) is desirable. In addition, the presence of distributed energy resources, storage entities, and electric vehicle fleets, along with the adoption of advanced control devices such as Flexible AC Transmission Systems (FACTS), further complicates the OPF formulation [2].

Due to its demanding nature and importance, numerous optimization techniques have been used to solve the OPF problem [2,3,9–12]. Optimization techniques can be broadly split into two categories: mathematical (also known as classical) and metaheuristic [2].

Mathematical techniques take advantage of the analytical properties of a problem to generate a sequence of points that converge to a globally optimal solution [13]. In the context of OPF, most mathematical techniques used in the literature use one of the following: gradient method, Newton's method, Linear Programming (LP), Sequential Linear Programming (SLP), Successive Quadratic Programming (SQP), and the Interior Point Method (IPM). While mathematical techniques can be efficient, accurate, and robust, they feature three main disadvantages: (i) their global optimality cannot be guaranteed for non-convex problems (such as OPF), (ii) they cannot easily handle discrete variables, and (iii) they are unsuitable for multi-objective optimization problems, since they cannot easily handle discontinuous or non-convex Pareto fronts [2,3,5,14].

These shortcomings have inspired researchers to apply metaheuristic techniques to the OPF problem. Metaheuristics are optimization techniques that are inspired by processes observed in physics, biology, or sociology [3]. Metaheuristic methods address the disadvantages of deterministic methods, as they can both escape local optima and converge to the global optimum, and easily handle discrete variables and discontinuous or non-convex Pareto fronts [2,3,5,14]. The main drawbacks of metaheuristics stem from their heavy computational burden and the need for application-specific tuning of various parameters (called hyper-parameters) to ensure adequate performance [2]. It is also important to note that LMPs, which have direct implications in market-based applications, cannot be readily produced by metaheuristics [5].

The first-mentioned drawback (i.e., the computational burden) was a detrimental factor to the adoption of metaheuristics for OPF when they were first introduced in the late 1990s, as computer resources were then scarce. Meanwhile, research conducted on metaheuristics has produced some guidelines on the tuning of the parameters of these algorithms (e.g., [3,15,16]). While a straightforward method for the extraction of LMPs from metaheuristics has not been determined, progress to that end has also been made (e.g., [17]). Thus, the relative abundance of computer resources nowadays and the general advancements in information technologies, as well as the aforementioned research on metaheuristics, lift (or at least ease) the barriers that prohibited their adoption in the past. Having mentioned the added complexity that trends in the industry impose on OPF, the authors of this paper feel that revisiting the use of metaheuristics in OPF can potentially provide a viable solution to address the increasing complexity of OPF in the current landscape.

To this end, this work focuses on the use of two of the most popular metaheuristic methods, namely the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) in the context of OPF. Apart from power engineering, GA and PSO have successfully been applied in various other domains such as image processing, software engineering, speech recognition, healthcare, machine learning, product design, optimal routing, scheduling problems, etc. [18,19]. It should also be noted that GA and PSO are not the only metaheuristic optimization methods that have been used in the context of OPF. Ant colony optimization, artificial neural networks, chaos optimization, differential evolution algorithms, simulated annealing, tabu search, as well as hybrid methods (which combine elements of the aforementioned methods), have been used in the context of OPF [9]. However, none of these methods have received as much research interest as GA and PSO. Therefore, despite their popularity, GA and PSO were chosen as the main focus of this review due to the depth of the existing literature on their application to OPF, which allows the extraction of useful conclusions.

1.2. Similar Works & Contribution

The goal of this work was:

1. To perform a comprehensive literature review of the most prominent GA and PSO OPF works.
2. To pinpoint the best practices from the reviewed works, and, where possible, corroborate them with literature from other domains.
3. To perform a cross-comparison among the reviewed GA and PSO OPF implementations (where possible) by analyzing the reported results.
4. To propose promising research directions for the standardization of GA and PSO OPF algorithms.

There are a few other research papers on this topic in the literature [3,9,12,14,20]. These works have made good contributions to organizing the existing literature, summarizing the existing research, and comparing the two methodologies. However, they leave some knowledge gaps that this work aimed to address. Specifically, the focus of [9,12,14] is not exclusively on GA and PSO, but rather on the whole topic of metaheuristics in the context of OPF. Consequently, the extent to which works using GA or PSO are reviewed is limited, as the authors do not go into detail about their mathematical formulations or the specific characteristics of each method (e.g., formulation-specific operators, fitness functions, tuning of parameters, etc.). On the other hand, [3,20] focus solely on PSO and GA. Yet, both works do not provide a comprehensive literature review on the topic. Instead, they focus on specific implementations of the GA and PSO and apply them to test cases to perform a comparison between these two mathematical programming methods. Moreover, except for [12,14], all the similar works were introduced more than a decade ago, leaving an important knowledge gap that involves works that were published in the meantime. A summary of the characteristics of similar works and a comparison with this work can be found in Table 1.

Table 1. Comparison between this work and similar works.

Reference	Literature Review Extent	Literature Review Focus	Analysis of Reviewed GA and PSO OPF Characteristics	Comparison of GA and PSO OPF	Date Published
[3]	Limited	PSO and GA	Yes (limited)	Yes (limited)	2005
[9]	Extensive	Metaheuristics	No	No	2012
[12]	Extensive	Metaheuristics	No	No	2018
[14]	Extensive	Metaheuristics	No	Yes (limited)	2021
[20]	Limited	PSO and GA	Yes (limited)	Yes (limited)	2007
This work	**Extensive**	**PSO and GA**	**Yes (extensive)**	**Yes (extensive)**	**2023**

This work aimed to fill all these gaps by performing a comprehensive review of relevant works, focusing on a comparison of the characteristics of the different implementations of GA and PSO OPF. This work focused on single-objective OPF, but some works featuring multi-objective OPF are also presented, for the sake of completeness. For the first time, the traits of GA and PSO OPF were standardized along several axes and analyzed. The practices used in GA and PSO OPF were compared to examine the best practices used in other domains. Where possible, the results of the reviewed works were cross-compared with each other, both in terms of accuracy and performance, to pinpoint the best practices of the reviewed works. Finally, the choice of hyper-parameters in the reviewed works is discussed, as a first step towards standardizing GA and PSO OPF.

The remaining sections of this paper are organized as follows: in Section 2 the OPF formulation is introduced. Section 3 presents the GA and a literature review on works using the GA OPF, and analyzes and discusses their characteristics. Section 4 presents the PSO and a literature review on works using PSO OPF, and analyzes and discusses their characteristics. In Section 5, GA OPF and PSO OPF are cross-compared, the relative performance and accuracy of each approach is discussed, and the choice of hyper-parameters of the reviewed works is analyzed. In Section 6, conclusions are drawn and potential directions for further research are identified.

2. The Optimal Power Flow Problem

The OPF problem contains a set of optimization problems that seek to optimize a given objective of the power system subject to its physical constraints. Physical constraints can be imposed either by electrical lows (e.g., power balance in the system) or the engineering limits of its components (e.g., thermal limits of line flows). OPF can be used by stakeholders to make informed decisions at any planning horizon. For example, OPF can be formulated for long-term transmission network capacity planning to minimize investment costs in network reinforcement, or in close to real-time to optimize the active and reactive power dispatch [21]. The goal of OPF is to calculate the values of the control variables (i.e., the variables whose values can be controlled) that result in the optimal value of the objective function. Therefore, in the example of long-term planning, the control variables could include the location of the installation of new lines, and in the example of dispatch, they could include the active and reactive power dispatch of generators. Then, using these values, the stakeholders can make informed decisions to optimize the operation of the power system. For an introduction to OPF, the reader is referred to [22], or academic textbooks such as [21]. More information regarding the solution to the OPF problem using mathematical methods can be found in [2,10,11], while information regarding more recent OPF solution techniques (such as intelligent algorithms and metaheuristic optimization methods) can be found in [9] or [12]. Other recent developments, such as the co-optimization of power-gas coupled systems are discussed in [23].

The OPF problem can be formulated as a mathematical optimization problem as follows:

$$\min f(\boldsymbol{z}) \tag{1}$$

subject to,

$$\boldsymbol{g}(\boldsymbol{z}) = 0, \tag{2}$$

$$\boldsymbol{h}(\boldsymbol{z}) \leq 0, \tag{3}$$

$$\boldsymbol{z} \in \boldsymbol{Z}. \tag{4}$$

Equation (1) defines the objective function of OPF, which represents the power system optimization goal. The objective function f can either be a scalar function (in single-objective OPF) or a vector function (in multi-objective OPF). Equations (2)–(4) define the OPF constraints. The equality constraints (2) represent the system's PF equations. Inequality constraints (3) represent the functional operating constraints, such as branch flow limits, voltage magnitude limits, etc. Constraints (4) define the feasibility region of the problem control variables, such as the active power output limits of the generating units, the transformer tap settings, etc.

The electrical state of the system is represented through the state variables. The state variables are continuous. Usually, they include nodal voltage magnitude and angle, and the nodal active and reactive power injections (although alternative formulations that use current instead of power injections have also been proposed). Control variables are a subset of the state variables, whose values can be controlled by the system operator to achieve the goal of optimization. Typically, in the context of OPF, control variables include the active and reactive power injection of generating (and sometimes also of demand-side) assets. However, depending on the objective function, control variables can also include regulated bus voltage magnitude [24], transformer tap settings [25], loads to shed [26], flexibility activation [27], etc. Control variables may be continuous or discrete.

The objective function (1) is typically the minimization of generation costs or the maximization of social welfare. Other objective functions include loss minimization, investment cost minimization¸ optimal voltage profile, minimization of environmental impact, minimization of load shedding, etc. [2]. The most common objective functions are presented in Table 2.

Table 2. Common OPF objective function categories.

Objective Function	Abbreviation	Example
(Linear/Quadratic) Generation Cost Minimization	(L/Q)GCM	Fuel cost minimization
Welfare Maximization	WM	Market-based applications
Losses Minimization	LM	Minimization of active power losses
Reactive Power Dispatch Optimization	RPDO	Minimization of reactive power dispatch
Investment Cost Minimization	ICM	Minimization of cost of capacitor banks
Post-Contingency Security Maximization	PCSM	Minimization of post-contingency loading
Minimization of Voltage Deviation	MVD	Voltage profile improvement
Minimization of Environmental Impact	MEI	Minimization of emissions
Congestion Minimization	CM	Overload alleviation

The choice of functions f, g, and h plays a pivotal role in determining the nature of the resulting OPF problem. For example, if linear approximations for all three functions are used, the resulting OPF will be a linear or mixed-integer linear problem, whose solution is relatively straightforward using mathematical methods such as simplex, the IPM, or SLP. Simplex exploits the convexity of linear problems to efficiently explore the feasible region and find the global optimum [2]. The most prominent linearization methodology is the DC approximation. SLP iteratively solves a series of LP subproblems [7], taking advantage of the desirable properties of the simplex method during the solution of each LP subproblem (e.g., [7,28]). The robustness, solution speed, simplicity, and efficiency of the simplex method make these methods some of the most popular OPF solution methods, even in real-life applications. However, because of the employed linearizations, some accuracy is sacrificed.

In the general case, f, g, and h are non-linear and non-convex functions, therefore the resulting OPF is a non-linear or non-linear mixed-integer optimization problem, whose solution is challenging. Mathematical methods used to solve the non-linear OPF include gradient methods, SQP, Newton's method, and the IPM. Gradient methods were amongst the first practical approaches (e.g., [29]) used to solve the OPF problem. However, even though they are easy to implement and reliable, they exhibit a slow rate of convergence [3] and can guarantee global optimality only for convex optimization problems [2]. SQP approaches (e.g., [30]) improve the rate of convergence of gradient methods by using second-order derivatives; however, their performance is not satisfactory, especially as the number of control variables increases [3]. Newton's method has been extensively studied in the context of OPF (e.g., [31]), mainly due to its quadratic convergence properties. Its shortcomings include difficulty in the identification of binding inequality constraints and its time-consuming solution process [3]. In recent years, IPMs have emerged as the most promising deterministic method for solving the non-linear (as well as the linear) OPF problem. It has been shown that IPMs provide the fastest and most efficient mathematical algorithms for many OPF applications [2]. Challenges of IPMs include efficient parameter selection and the assurance of convergence [2]. Research on IPMs and their variants (e.g., primal-dual IPMs, trust-region IPMs, etc.) is ongoing.

Since, to date, no single solution approach is deemed suitable for all the different forms of OPF problems [2,12], practitioners often have to perform a trade-off between accuracy, efficiency, and robustness, depending on the OPF problem they are facing. When robustness and solution speed are more important than accuracy and realistic reflection of the physical constraints, the linear OPF is an attractive formulation, with many real-life use-cases. Although, in cases where accuracy is desired, the choice of a solution method is more complex. Therefore, the metaheuristic methods that are presented in the following sections should be considered as alternatives for non-linear mathematical techniques, as the efficiency and convergence properties of linear mathematical methods are well-documented.

3. Optimal Power Flow Using the Genetic Algorithm

3.1. The Genetic Algorithm

GAs are general-purpose, robust optimization algorithms that were invented by Holland in the early 1970s [32]. Their operational principles are based on the Darwinian laws of genetics, natural selection, and evolution. They use chromosomes to represent the control parameters of a given problem. Chromosomes are composed of genes. GAs start from a random initial population of candidate solutions, and rely on the biologically inspired operators to iteratively evolve towards fitter (i.e., better) solutions [33]. The fitness of each solution is evaluated using the Fitness Function (FF), which assigns a quality value to each solution. The process is repeated for each generation g, until the maximum number of generations g_{max} is reached. The process is outlined in Figure 1.

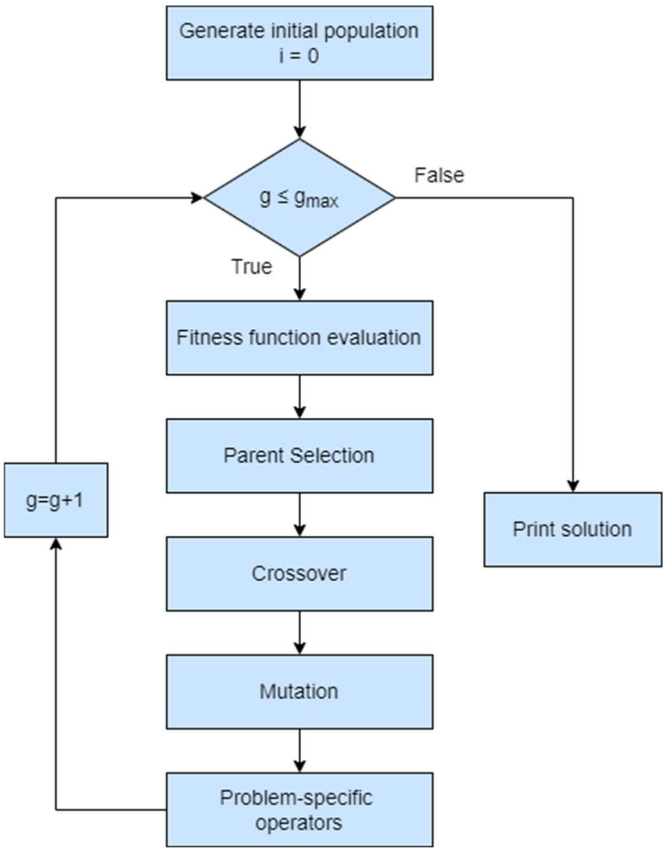

Figure 1. Outline of the GA.

The formulation of the FF is the most crucial step in GA (and PSO). To formulate the FF, the objective function (1) of the original problem is used (if the original problem is a minimization problem, it has to be transformed into a maximization problem). By only using (1) as the FF, the problem is unconstrained as GA and PSO do not have any knowledge of the problem's search space. Therefore, the FF must also penalize violations of the search space's constraints (2)–(4), so that they are respected. Thus, the general form of the FF in GA and PSO OPF is:

$$FF = Objective\ Function - Penalties\ of\ Constraints. \tag{5}$$

There are no straightforward rules for the penalties that must be introduced for the violations of the problem's constraints in the FF. Usually, the specific characteristics of each problem determine the choice of penalties in the FF. The most popular way of penalizing violations of the problem's constraints is through the use of linear or quadratic distance-based penalties. These methods introduce penalties using the distance from the constraint limit as a measure of "severity" of the constraint violation. Linear distance-based penalties attribute the same weight to all violations, while quadratic distance-based penalties penalize more heavily "large" violations of the problem's constraints. Additionally, some practitioners may use coefficients to attribute weights to violations of specific constraints of the problem. These penalties can be static or dynamic. Usually, when penalty coefficients are dynamic, they are small in the first generations (allowing for better exploration of the search space), and they become larger in the later generations (emphasizing the feasibility of the candidate solutions). An introduction to penalty selection can be found in [33].

The biologically inspired operators at the core of GA are *parent selection*, *crossover*, and *mutation*. In *parent selection*, two chromosomes from the population are selected to produce an offspring for the next generation. Bias is introduced so that chromosomes with a high fitness have more chances of being selected to produce offspring [33]. *Crossover* is responsible for the exchange of genetic information between mating chromosomes. During crossover, the chromosomes that were selected during parent selection are recombined from new chromosomes that combine the genetic information of the two parents. The crossover probability is defined as the probability that two chromosomes exchange genetic information. A high crossover probability should be applied to improve the convergence speed [33]. *Mutation* is responsible for introducing new information in each generation by randomly perturbing the values of some genes of some chromosomes. A small mutation probability should be applied, to prevent the GA from converging in local optima [33]. According to [33], application-specific operators that leverage the characteristics of each problem can be applied to enhance the efficiency of the GA. A comprehensive description of the GA can be found in [33].

As discussed earlier, there are three main differences between GAs and mathematical optimization techniques. First, GAs operate on the encoded string of the control variables, and not the actual control variables of the problem. Secondly, GAs are population-based optimization techniques and use several points rather than a single point in their search. Finally, GAs do not require any prior knowledge regarding the objective function, such as smoothness, convexity, linearity, continuity, etc. The only requirement is the calculation of the FF, which assigns the quality value to each solution (and presupposes knowledge of the status of the system's non-controllable elements, which is a given). The latter two advantages also apply to PSO. Research on the use of GAs in the context of the OPF has been extensive, as is analyzed in the following subsection.

3.2. Genetic Algorithm Optimal Power Flow

The seminal work on the use of GA to solve the OPF is [25]. In [25] the authors introduced the so-called Enhanced Genetic Algorithm (EGA), which demonstrates remarkable performance enhancements compared to other GA implementations. In the EGA, parent selection is carried out through roulette-wheel selection and uniform crossover is applied between the mating chromosomes. Chromosomes are encoded using binary digits. The authors of [25] introduced some "advanced" features and some problem-specific operators in their implementation that enhanced its performance. Specifically, the advanced features of fitness scaling, elitism, and hill-climbing were introduced. Fitness scaling aims to avoid encourage healthy competition among equals and avoid the domination of extraordinary solutions during the first few iterations. Elitism and hill climbing aim to enhance the convergence speed of the algorithm. Five problem-specific operators were used to introduce random modifications to the chromosomes of each new generation. Additionally, population statistics were calculated to adaptively change the crossover and mutation probabilities of each generation. To improve the efficiency of the EGA, the Fast-Decoupled Load Flow (FDLF) algorithm was used for the execution of the required PF analyses.

In [25] the EGA was compared with the Improved Genetic Algorithm (IGA) introduced in [34]. IGA also uses binary encoding, roulette-wheel selection, and uniform crossover. The main difference between EGA and IGA (beyond the advanced features and operators introduced in [25]) lies in the choice of the FF. EGA uses fitness scaling and linear penalties for constraint violations, while IGA does not use fitness scaling and uses quadratic penalties for constraint violations. According to [25], the performance of EGA is superior to that of the IGA, as demonstrated through various test-cases.

Ref. [35] proposed a GA consisting of two parts to solve the OPF. The first part aimed at finding feasible solutions, while the second part aimed at accelerating the convergence to the optimal solution. To this end, [35] combined the concepts of co-evolution and solution repairs (for the first part of the algorithm) and elitism (for the second part). The repair of infeasible solutions was carried out through the use of a repair function, which co-evolves the infeasible individuals until they become feasible. The proposed algorithm used binary encoding, roulette wheel selection, and single-point crossover. The proposed algorithm was demonstrated in 6-bus system.

The Adaptive GA with Adaptive Population size (AGAPOP) was introduced in [36]. The main idea of the AGAPOP was to flood the high-dimensional solution space with solutions that cover the entire search space, then decrease the population size when a direction that increases the elite FF was found. To achieve this, mutation, crossover probabilities, and the population size were all adaptively changed based on the FFs of the population. The AGAPOP also used binary encoding and roulette-wheel selection. The choice of crossover methods and penalization of constraint violations in the FF are not discussed. The application of the AGAPOP was demonstrated using the IEEE 30-bus network for three objective functions (fuel cost minimization, minimization of voltage deviations, and fuel cost minimization with quadratic cost functions). According to [36] the simulations with AGAPOP yielded more accurate results and required fewer generations compared with other optimization methods, including the IGA introduced in [34].

In [37] the GA was applied for the minimization of the severity of a post-contingency state in power systems. The proposed GA used real encoding, tournament selection, and blend crossover (for continuous variables), and single-point crossover (for integer variables). Real-encoding was used, as it is argued that the resulting GA is more efficient, as the encoding-decoding procedures are eliminated, and more accurate, as no accuracy is lost due to the discretization of binary variables. To calculate the FF, PF analyses using the Newton-Raphson (NR) method were executed. Violations of the system's constraints were penalized with quadratic penalties. The proposed algorithm was successfully applied in the IEEE 30-bus and 118-bus power systems.

Ref. [38] introduced the Refined Genetic Algorithm (RGA). The RGA uses binary encoding, roulette-wheel selection, and uniform crossover. The main contribution of the RGA lies in dynamically (exponentially) varying the mutation and crossover probabilities, in order to achieve faster convergence and better exploration of the search space. The required PF analyses were solved using the Decoupled PF, to achieve better performance. In [38] cost-minimization was used as the objective function of the OPF, and violations of the power system's constraints were penalized with quadratic penalties. The effectiveness of the proposed method was demonstrated in 6-bus and the IEEE 30-bus power system.

The EGA introduced in [25] was modified to tackle multi-objective OPF in [39]. To this end, different combinations of the following objectives were considered: generation cost minimization, minimization of losses, and maximization of voltage stability. The work used the penalties introduced by [34] to punish the violations of the network's constraints. The strength Pareto Evolutionary Algorithm was introduced, which finds the best compromise solution that satisfies all the objectives, using the EGA. Decoupled Quadratic Load Flow (DQLF) was used to solve the necessary PF analyses. The effectiveness of the proposed algorithm was demonstrated in the IEEE 30-bus power system, where its performance was found to be superior compared to a PSO-fuzzy algorithm.

The authors of [40] argue that the selection of the initial population plays a crucial role in the performance of the GA. Therefore, their work focused on the selection of an initial population, which resulted in increased performance of the GA. To this end, a methodology to select the active power generation and voltage magnitudes in generator buses was developed. The proposed initialization methodology was then used in a real encoded GA, which used roulette wheel selection (based on relative fitness), and single-point crossover. A FF of cost minimization was chosen, where violations were penalized with quadratic penalties. Penalty coefficients were initially small but were dynamically increased in each generation, to better control the exploration of the search space. The FDLF and the methodology introduced in [41] were used to solve the required PFs. The proposed GA was applied in 4 test cases, where it was proven that the proposed initialization methodology drastically reduced the required solution time and improved the performance of the proposed GA-OPF.

Ref. [42] used GA to choose optimal locations and sizes for both shunt capacitors and series voltage regulators in three-phase unbalanced distribution systems. The proposed GA used a binary encoded GA, roulette wheel selection, and dispersed crossing. The OPF was formulated as a multi-objective problem with the objectives of active power loss minimization, minimization of voltage limit violations, minimization of voltage drop violations, minimization of cost of capacitor banks, and minimization of cost of voltage regulators. Different penalties were introduced in each objective function, and the problem was converted from a multi-objective to a single-objective function using the global criterion method. The effectiveness of the method was demonstrated via its application in a 70-bus system.

In [17] the GA with Generating Scaling Factors (GA-GSF) was introduced, which aimed to address a major shortcoming of GA-OPFs: to determine a framework for the extraction of LMPs from GA-OPFs. In the case of uncongested lines, the proposed framework used the price of the marginal unit as the LMP. In case congested lines existed, the inverse Jacobian matrix was used to calculate the sensitivity of active power injection with regards to voltage magnitude and angles, and of active power injection with regards to line loading (a detailed analysis of this methodology can also be found in [43], albeit in a different context). This sensitivity, in essence, expresses the per megawatt increase of the cost of production, while respecting (the linearized) network constraints. The GA-GSF was then demonstrated in a 14-bus network and was capable of successfully calculating the LMPs of each node.

The authors of [44] explored the capabilities of GA in fuzzy goal programming. Specifically, they developed a GA algorithm in order to solve the congestion management problem, in the presence of priority-based fuzzy goals. The problem was multi-objective, with the following objective functions considered: minimization of overloading, minimization of system losses, and minimization of operational costs. To solve this problem, a GA algorithm with binary encoding, roulette wheel selection, and single-point crossover was used. The multi-objective optimization problem was converted to a single-objective problem by associating numerical weights relative to the importance of each objective. The proposed GA is applied in the IEEE 30-bus system to demonstrate its effectiveness.

One of the most recently introduced works to explore GA in the context of OPF is [45]. This work aimed to formulate a multi-objective OPF, which considered the stochastic nature of wind and solar power. The objectives that were considered by this work were: minimization of generation cost, minimization of real power losses, minimization of voltage deviations, and minimization of emissions. To solve the problem, the authors used an adapted version of the well-established NSGA-II algorithm [46]. Specifically, they used adaptive crossover, mutation, and selection based on differential evolution to ensure better exploration of the search space. The proposed method was compared with NSGA-II in several test cases, and was found to perform better.

3.3. Discussion on GA OPF

In Table 3 a summary of the reviewed works is presented. The analysis of the reviewed works focused on seven axes:

1. *Encoding*: The choice of encoding is an important factor in GA applications. Most early works use binary encoding, as it lowers the memory requirements. More recent works usually adopt real encoding, as it circumvents the encoding-decoding steps of the GA, increasing the overall efficiency.
2. *Parent selection*: Almost all works use roulette wheel selection, as it is easy to implement and leads to good results. A few exceptions also exist, e.g., [37] which uses tournament selection. According to [16], tournament selection allows for better exploration of the search space, but it leads to slower convergence. It should be noted that using roulette wheel selection requires special transformations, in case particles with negative fitness values exist amongst the population.
3. *Crossover*: Single-point crossover is the most popular choice amongst the reviewed papers, as it is one of the first and easiest ways to implement crossover methods. The uniform crossover has also been used, which is considered suitable for large-scale problems [16]. Other crossover methods, such as blend and dispersed crossing, have also been explored in the context of GA OPF.
4. *Adaptiveness*: In this work, adaptiveness is defined as the dynamic tuning of coefficients over the generations. Many works adaptively change mutation and crossover probabilities, using each generation's population statistics, to keep the diversity within the population and allow for better exploration. Other works dynamically change coefficients for the penalization of constraint violations. These coefficients are usually small during the first generation (i.e., violations are not heavily penalized), which allows for better explorations of the edges of the search space. In later generations, violations are heavily penalized, to keep the population within the search space. Some works omit the use of adaptive coefficients and rely on static values.
5. *Power flow formulation*: To calculate the FF for each member of the population, a PF analysis must be executed. Due to the volume of the required calculations, this is the most time-consuming step of the GA OPF. Most works use a full PF, i.e., no approximations are used and the PF is solved, e.g., with the Newton-Raphson method. Other methods used to solve the required PFs include FDLF, DQLF, and the method introduced in [41]. While these methods can ease the computational burden, they can lead to some loss of accuracy and pose some limitations to the applicability of each algorithm (e.g., FDLF is not accurate in the distribution network due to its line characteristics).
6. *Objective functions*: As presented in Table 2, many different objectives can be applied to OPF. While cost minimization is the most popular amongst the reviewed works, a wide array of other objectives has also been used, which is a testament to the versatility of the GA OPF.
7. *Constraint violation penalties*: As earlier discussed, to include the constraints in the GA OPF, penalties for the violation of each constraint are added to the FF. Most works use quadratic penalties to penalize the violations. Quadratic penalties penalize heavily large deviations from the search space limits and lightly smaller deviations. This way, they allow for better exploration of the search space, as points close to the bounds of the search space are not penalized too much, but points far off are. On the contrary, linear penalties place the same weight on smaller and larger violations of constraints.

Table 3. Summary of the reviewed GA OPF works.

Ref.	Encoding	Parent Selection	Crossover	Adaptiveness	Power Flow Formulation	Objective Functions *	Constraint Violation Penalties
[25]	Binary	Roulette wheel	Single-point	Mutation, crossover probabilities	FDLF	QGCM	Linear
[34]	Binary	Roulette wheel	Uniform	Static	Full PF	QGCM	Quadratic
[35]	Binary	Roulette wheel	Single-point	Constraint violation penalties	Full PF	QGCM	Quadratic
[36]	Binary	Roulette wheel	N/A	Mutation, crossover probabilities, population size	Full PF	LGCM, QGCM, MVD	N/A
[37]	Real	Tournament	Blend	Static	Full PF	PCSM	Quadratic
[38]	Binary	Roulette wheel	Uniform	Mutation, crossover probabilities (exponential variation)	FDLF	QGCM	Quadratic
[39]	Binary	Roulette wheel	Single-point	Mutation, crossover probabilities	DQLF	QGCM, LM, MVD	Quadratic
[40]	Real	Roulette wheel	Single-point	Constraint violation penalties increase	FDFL, [41]	QGCM	Quadratic
[42]	Binary	Roulette wheel	Dispersed crossing	Static	Full PF	LM, MVD, ICM	Quadratic
[17]	Real	N/A	N/A	Static	Full PF	QGCM	Linear
[44]	Real	Roulette wheel	Single-point	Static	Full PF	*CM, QGCM, LM*	N/A
[45]	Real	N/A	N/A	Crossover, mutation, probabilities	Full PF	*QGCM, LM, MVD, MEI*	N/A

* Italics indicate multi-objective OPF.

Other common traits found in most works are elitism, hill climbing, and fitness scaling. Elitism makes sure that the solution with the best FF value is passed to the next generation. Hill climbing is closely related to elitism, as it involves randomly perturbing the value of a randomly selected gene in the elite chromosome. Its FF is then re-calculated, and if the disturbance results in an increased FF value, the modified chromosome is accepted; otherwise, the change is reversed. Hill climbing and elitism improve the convergence rate of the GA. Fitness scaling is used to avoid the early domination of extraordinary solutions in the early stages and involves scaling the FF values of the entire population to foster competition.

Surprisingly, the only work that features problem-specific operators is [25]. The use of problem-specific operators is recommended to increase the performance of the GA, regardless of the domain [33]. These operators can be based on simple heuristics or observations based on the knowledge of the problem at hand. For example, in [25] five operators were introduced, each aiming to exploit the specific characteristics of OPF, and successfully increased the proposed method's performance.

The important issue of the initial population choice for the GA OPF was addressed in [40]. Literature suggests that initial populations may have a significant effect on the best objective function value over several generations [47]. According to [40], this is also the case in the context of OPF. Importantly, the authors of [40] also suggest a framework for creating an initial population for the GA OPF that leads to an increase in performance.

The main gap in GA OPF is the lack of a systematic way to choose the hyperparameters (e.g., population size, number of generations, crossover probability, etc.) of the GA OPF. All the reviewed works relied on rules of thumb or repeated simulations and observations to fine-tune the GA OPF.

Regarding the population size and the number of generations, [48] suggests that a larger population size should be preferred over a greater number of generations (if the available memory suffices). Ref. [49] was one of the first to give guidelines on the optimal population size of the GA. Adjusting the population size can lead to a reduction of the computational burden, without loss of accuracy [50]. This was also demonstrated in the context of OPF in [36].

The choice of mutation and crossover probabilities also play a crucial role in the performance of the GA. The consensus is that crossover probability should be high (i.e., greater than 0.8), and mutation probability should be low (i.e., about 0.0001 for every bit) [33]. Research suggests that these probabilities should change as the number of generations increases, for better exploration of the search space and better overall performance of the GA [19]. This has also been demonstrated in a number of the reviewed works on OPF.

In our opinion, for applications with critical performance requirements, practitioners should look into the more "advanced" GA features, such as problem-specific operators, population initialization strategies and dynamic population size, mutation, and crossover probabilities adjustment. The literature suggests that these features can increase the performance and accuracy of the GA OPF. For more trivial GA OPF applications, simpler GA OPF implementations can lead to satisfactory results and are easier to implement.

4. Optimal Power Flow Using Particle Swarm Optimization

4.1. The Particle Swarm Optimization Algorithm

PSO is an iterative, general-purpose, population-based, robust optimization algorithm, which was introduced by Kennedy and Eberhart in 1995 [51]. The method was broadly inspired by the swarm behavior of animals (e.g., schools of fish or birds flocking), as stated by its inventors [51]. In PSO, the search is conducted by using a population (or swarm) of particles (candidate solutions) to look for the optimal solution. Each particle moves in the multi-dimensional search space. The behavior of each particle is described by two parameters: its position and velocity (stored in the $\boldsymbol{X}$ and $\boldsymbol{V}$ matrices, respectively). According to [51], the movement of the animals that inspired PSO is inspired by two components: the cognitive (individual) and the social components. In the context of animal behavior, the social component suggests that individuals disregard their own behaviors and adjust to the behavior of other individuals in their proximity. On the other hand, the cognitive component suggests that individuals are isolated beings and disregard the behavior of individuals in their proximity, and adjust their behavior based on their own experiences. PSO combines the social and the cognitive components to adjust the behavior (i.e., change the position in the search space) of each individual particle. Even though PSO is more geared towards using continuous variables, a discrete binary version of PSO has also been introduced by Kennedy and Eberhart [52]. The overall performance of PSO in mixed-integer programming has been thoroughly verified (e.g., [53]).

The basic elements of PSO according to [54] include particles, time, the population, particle velocity, inertia weights, fitness, individual best, and global best. A *particle* j is a candidate solution, presented by the m-dimensional vector $\boldsymbol{X_j}(t) = \left[x_{j,1}(t), \ldots, x_{j,m}(t)\right]$, where m is the number of optimization variables, and $x_{j,k}(t)$ is the position of the j-th element with respect to the k-th dimension ($k \in \{1, \ldots, m\}$) at time t. The parameter t is used as a counter to count the elapsed *time* (or *epochs*) of the PSO and is incremented in each generation (similar to the generation counter in GA). The *population* $\boldsymbol{P_t}$ includes all n particles (n is the number of particles) at time t: $\boldsymbol{P_t} = [X_1(t), \ldots, X_n(t)]^T$. *Particle velocity* is represented for each particle j with the m-dimensional vector $\boldsymbol{V_j}(t) = \left[v_{j,1}(t), \ldots, v_{j,m}(t)\right]^T$, where $v_{j,k}(t)$ is the velocity of the j-th particle with respect to the k-th dimension at time t. *Inertia weights* are used to control the effect of previous velocities on the current velocity. This is an important parameter in PSO, as it balances the trade-off between the local and global exploration of the process. The concept of *fitness* in PSO is the same as that in GA, i.e., a FF is used to assign a fitness value to each particle, which is used as a measure of the quality of each solution. The *individual best* position $\boldsymbol{X_j^*}$ of a particle is defined as its position where the individual encounters its best fitness score. The global best position $\boldsymbol{X}_j^{**}$ is the position amongst all individuals where the best fitness score was encountered. The individual and global bests are stored in each iteration, and along with the inertia weights, they are used to update each particle's velocity in each iteration. An outline of PSO can be seen in Figure 2.

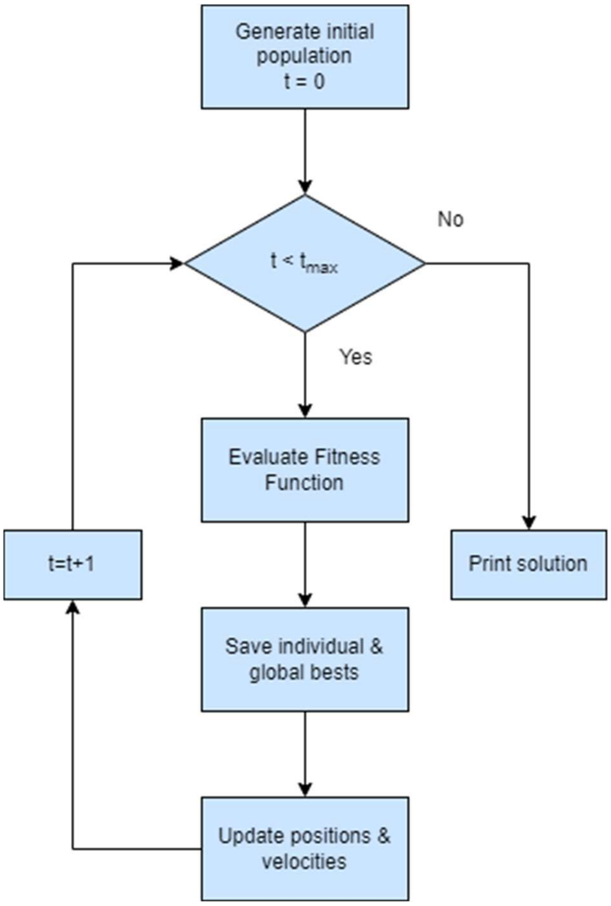

Figure 2. Outline of PSO.

4.2. Particle Swarm Optimization Optimal Power Flow

The seminal work on PSO in the context of OPF is [54]. In [54] PSO was used to solve a single-objective OPF. The effectiveness of the method in four different objective functions was examined: fuel cost minimization, voltage profile improvement, voltage stability enhancement, and cost minimization with quadratic cost curves. An annealing procedure was introduced to fine-tune the search procedure. The annealing procedure decreased the value of inertia weights as time progressed. This allowed for a more uniform search in the initial stages and a more local search in the final stages. Infeasibilities in the FF were punished with the use of quadratic penalties. Each particle position was checked for feasibility. If infeasibilities were detected, the corresponding limit on each direction was imposed. Finally, a maximum particle velocity was enforced, to enhance local exploration. The proposed PSO was successfully demonstrated in the IEEE 30-bus test system, for all four objective functions.

Ref. [55] introduced a Modified PSO (MPSO) for the OPF. According to the authors, the fact that particles in PSO update their position only based on the individual and personal bests can lead to early domination of the swarm by the personal best. To mitigate this, in MPSO, the individual best position and the position of another random particle are used to update each particle's position. MPSO was demonstrated in a 5-bus test case, with the objective of cost minimization.

The authors of [56] introduced an Improved PSO (IPSO) to solve the OPF. The main contribution of [56] lies in introducing the so-called non-stationary multi-stage assignment penalty function to their formulation. According to the authors, the use of proper values of static penalties to punish constraint violations is challenging and can cause poor performance. To mitigate this, they introduced a non-stationary multi-stage assignment penalty function that dynamically altered the penalties in each iteration, depending on the iteration counter and the penalty factor. The maximum velocity limit and feasibility restoration used in [54] were also used in this work. The proposed IPSO was successfully demonstrated in the IEEE 30-bus test system, using cost minimization, reactive power optimization, and optimal active and reactive power dispatch as the objective functions. The same authors revisited and further improved the IPSO OPF in [57]. The required modifications in the IPSO for the accommodation of integer variables were introduced, and the IPSO was applied in two test cases with the objective of reactive power dispatch optimization.

In [58] the concepts of aggregation and congregation were used. Aggregation refers to the swarming of particles by external forces, and congregation to the swarming of particles by social forces. Both concepts can be expressed actively or passively. In [58] active aggregation and passive congregation were considered, and to this end, the Local Passive Congregation (LPAC) and the General Passive Congregation (GPAC) PSO algorithms were developed. Moreover, the concept of Coordinated Aggregation (CA) was explored, according to which particles consider moving towards particles with better achievements than their own. The CA PSO algorithm was developed to examine its effectiveness. All three proposed PSO algorithms were compared to an IPM solver and a simple PSO algorithm for three test cases and two objective functions. The proposed algorithms were found to outperform their counterparts in terms of accuracy but required more solution time. While the performance of LPAC was the best, the authors argue that the use of CA should be preferred, as it requires the tuning of fewer parameters. The CA PSO algorithm is also discussed by the same authors in [59].

The concept of passive congregation was also used in [60], where PSO with Passive Congregation (PSOPC) was introduced. The authors argue that without the extra information passive congregation provides, the population is likely to lose diversity and be confined to local minima. The maximum velocity in this work was set to half the length of the search space. PSOPC was subsequently used to optimize fuel consumption, voltage profile, and voltage stability in the IEEE 30-bus test system.

The focus of [61] lies in using PSO to solve the OPF subject to security constraints. The main contribution of [61] was the introduction of the Reconstruction Operators (RO) that allowed particles to satisfy the units' operating constraints, while only searching the feasible space, reducing the computing time. Moreover, a penalty dynamic coefficient for the punishment of constraint violations was introduced. The proposed RO PSO was applied in two test cases.

Ref. [62] focused on the integration of integer variables in the PSO, and introduced the Mixed-Integer PSO (MIPSO) algorithm. According to the MIPSO algorithm, for discrete variables, trajectories can be interpreted as probabilities, not as the values of the variables themselves. Thus, the trajectories are changes in the probability that a coordinate will take on a discrete value. The effectiveness of the MIPSO algorithm was demonstrated in two test cases, for three different objective functions.

Another work that focused on the integration of integer variables is [63]. This work encoded integer variables using 8-bit encoding. Moreover, Ref. [63] explored the use of the constriction factor to ensure the convergence of the proposed method and control its convergence speed. The proposed method was demonstrated in the cost minimization of the IEEE 30-bus and 96-bus networks and was compared with a simple GA implementation, which it outperformed.

The authors of [64] focused on the impact of time-varying inertia weights on the performance of PSO. To this end, they examined three algorithms: a PSO with static weights, a Time-Varying Inertia Weight (TVIW) PSO where weights linearly decreased with respect to time, and their proposed algorithm, GLbest Inertia Weight (GLbestIW) PSO. According to the GLbestIW PSO, inertia weight is defined as a function of the local best and global best values of the particles in each generation. GLbestIW PSO outperformed the other two examined PSO implementations in the cost-minimization of the IEEE 30-bus network.

The choice of a time-varying inertia weight strategy was the focus of [65]. To this end, the Weight-Improvement (WI) PSO algorithm was introduced. The WI PSO combined the linear reduction of the weights with respect to time with a stochastic element it introduced. The WI PSO led to more accuracy and faster convergence than traditional PSO, as demonstrated in the cost-minimization of the IEEE 30-bus network.

Ref. [66] introduced PSO with Aging Leader and Challengers (PSO ALC). PSO ALC was inspired by the law of nature that every organism on earth has a limited lifespan. Therefore, in the context of PSO, the leader of the swarm becomes older with the passage of time, and gradually losses its ability to lead the swarm by being challenged by new and younger challenges. A framework for dynamically tuning the lifespan of the leader is introduced, based on the leading power of the leader. If the leader shows strong leading power, its lifespan is extended; otherwise, if the leader fails to improve, challengers claim the leadership, bringing diversity to the swarm. The PSO ALC was demonstrated in the IEEE 30-bus and IEEE 118-bus networks, with objectives of fuel cost minimization, active power loss minimization, and voltage deviation minimization. The same authors used PSO ALC to tackle OPF with FACTS devices in [67].

The Fuzzy-Based Improved Comprehensive-Learning PSO (FBICL PSO) algorithm was introduced in [68]. Firstly, the authors of [68] were concerned with introducing some less common practical generator constraints in the OPF problem, such as the valve-point effect, multi-fuel option, prohibited operating zones, and the modeling of FACTS in the formulation. Regarding the FBICL PSO, its first novelty lies in using the Comprehensive Learning (CL) framework in the context of OPF. CL was used as a counter-measure to the dominance of the global best position in the updated positions of the particles. So, instead of using the global best position, in the CL framework, the personal best positions of other particles were used, which according to the authors, improved the diversity of the swarm. In [68], the CL framework was improved by introducing an iterative mutation strategy, which randomly selected three mutant vectors from the initial population and applied them to the current population. Another contribution of [68] lies in introducing fuzzy logic using Mamdani-type fuzzy rules to adapt the inertia weights in each iteration. The fuzzification and de-fuzzification processes were thoroughly described. The authors argue that the introduction of fuzzy logic improves the search ability (and hence the performance) of the proposed algorithm. The effectiveness of the FBICL PSO was demonstrated in various test cases on the IEEE 30-bus network and was compared against other techniques.

In [69], PSO was used in a relatively new problem, the estimation of the Feasible Operating Region (FOR) of a distribution network. As the exploration of points close to the operational limits of the network is important in this problem, emphasis was given to the penalty coefficients used in the FF. Linear penalties were used, that were scaled dynamically depending on the elapsed time (i.e., initially penalty coefficients were small, to allow for exploration of the search space, and towards the end, penalty coefficients were high to encourage feasible solutions). Two operators to balance the trade-off between the exploration of feasible and infeasible points were also introduced. The proposed algorithm was successfully applied in a benchmark CIGRE distribution network.

Finally, Refs. [70,71] are two of the seminal works that use PSO to tackle multi-objective OPF. In [70], the Multi-Objective PSO (MO PSO) algorithm was introduced to solve an OPF that co-optimizes fuel and wheeling costs. The authors extended the single-objective PSO by proposing new definitions of the local and global best individuals for multi-objective optimization problems. Emphasis was also given on the use of clustering techniques to reduce the size of the Pareto Set, to improve the algorithm's performance. The effectiveness of MO PSO was demonstrated for two cases in the IEEE 30-bus network. Ref. [71] introduced a fuzzy decision-based multi-objective PSO to solve the OPF problem. The combination of cost, losses, voltage stability, and emission reduction was considered as the objective function. A self-adaptive method for the factors that determine the influence of personal and global bests was used, which according to the authors, had a significant effect on the rate of convergence. Moreover, inertia weights were tuned dynamically using a chaotic formula to balance global and local exploration. Similar to [68], a mutation factor was also used to avoid trapping the proposed algorithm in local optima. The proposed algorithm was applied in the IEEE 30-bus network for two test cases.

4.3. Discussion on PSO OPF

In Table 4 a summary of the reviewed works is presented. The analysis of the reviewed works focused on four axes:

1. *Adaptiveness*: Similar to the GA OPF analysis, adaptiveness is defined as the dynamic tuning of parameters as time elapses. Most reviewed inertia weights adapt dynamically, as time elapses. The simplest form of adaptation is to linearly reduce the weights as time elapses. However, more sophisticated techniques have also been proposed, such as the GLbest and WI algorithms (in [64,65] respectively) or fuzzy rules [68]. According to all researchers, the use of adaptive weights allows for a more uniform search of the search space. Apart from inertia weights, penalty coefficients are also commonly adapted dynamically. These works use small penalty coefficients at the beginning of the process, to allow for better exploration of the search space. Penalty coefficients are subsequently increased in the latter stages, to confine the swarm within the feasible search space. An interesting concept is presented in [66], where the leader's leading power was dynamically adapted.
2. *Maximum velocity*: Many works use an upper limit on the velocity of each particle for each dimension. The use of such a limit can enhance the local character of the search, and better resembles the incremental process of the human learning process [54].
3. *Objective functions*: The PSO OPF has been applied for many different objective functions, showcasing its versatility.
4. *Constraint violation penalties:* Similar to the GA OPF analysis, quadratic penalties penalize heavily large deviations from the search space limits and lightly smaller deviations. This way, they allow for better exploration of the search space, as points close to the bounds of the search space are not penalized too much, but points far off are. On the contrary, linear penalties place the same weight on smaller and larger violations of constraints.

First of all, it is worth mentioning that only four axes were used for the analysis of PSO OPF, while seven were necessary for the analysis of GA OPF. This is indicative of the relatively simpler implementation of PSO compared to GA, as also noted by other researchers (e.g., [14]). This is also evident by the smaller number of hyperparameters that need to be tuned in the case of PSO, as is discussed later in this paper. In addition, PSO OPF generally requires less computational burden, thus all works used a full PF formulation.

Another typical method that was used throughout the various works that were reviewed (e.g., [63,69]) is the so-called constriction factor method. The constriction factor was first introduced in [72], where it was demonstrated that its use may be necessary to insure the convergence of PSO. The constriction factor is used as a coefficient that controls the rate of change of speed between time intervals. Typically, the constriction factor K is expressed as:

$$K = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|},\ where\ \varphi = c_1 + c_2,\ \varphi > 4 \tag{6}$$

and c_1, c_2 are the cognitive and social accelerator coefficients, respectively. In [72], it is demonstrated that the convergence characteristics of the system can be controlled by φ. A φ value greater than 4 is required for convergence. The smaller the constriction factor, the slower the convergence. A thorough analysis of the stability of PSO can be found in [73].

Table 4. Summary of the reviewed PSO OPF works.

Ref.	Adaptiveness of Inertia Weights	Maximum Velocity	Objective Functions *	Constraint Violation Penalties
[54]	Annealing to reduce inertia weights	Yes	LGCM, QGCM, MVD	Quadratic
[55]	Dynamic inertia weights adjustment	N/A	LGCM	Linear
[56,57]	Dynamic penalty coefficients	Yes	LCGM, RPDO	Linear
[58,59]	Dynamic inertia weights adjustment	Yes	LM, MVD	Linear
[60]	No	Yes	GCM	Quadratic
[61]	Dynamic penalty coefficients, inertia weights adjustment	N/A	GCM	Quadratic
[62]	Dynamic inertia weights adjustment	Yes	GCM, MVD	Linear
[63]	Dynamic inertia weights adjustment	Yes	GCM	Quadratic
[64]	Dynamic inertia weights adjustment (GLbest algorithm)	N/A	GCM	N/A
[65]	Dynamic inertia weights adjustment (WI algorithm)	N/A	GCM	N/A
[66,67]	Dynamic adjustment of leader's leading power	N/A	GCM, LM, MEI	Linear
[68]	Dynamic inertia weights adjustment (Mamdani-type fuzzy rules)	Yes	GCM	Quadratic
[69]	Dynamic penalty coefficients	Yes	FOR estimation	Linear
[70]	No	N/A	*GCM, wheeling cost minimization*	N/A
[71]	Dynamic inertia weights adjustment (chaotic formula), influence of the personal and global bests (self-adaptive)	N/A	*GCM, LM, MEI*	N/A

* Italics indicate multi-objective OPF.

The simplicity of PSO stems largely from the fact that a particle uses only two reference points to update its position: the individual and global bests. Some of the reviewed works (e.g., [58–60]) argue that this causes the premature convergence of PSO. Hence, methodologies based on congregation are used to mimic the social interactions of animals [74] to enhance the diversity of the swarm. An analysis of the available congregation methods and their usefulness in the context of OPF is presented in [58]. According to the works that use congregation, the use of congregation increases overall PSO performance but comes at the expense of requiring tuning of more parameters. These findings are in accordance with relevant work in other fields (e.g., [75]).

As previously discussed, PSO has been introduced as an optimizer of continuous variables [51], but later a version of PSO that also supported discrete variables emerged [52]. This led to the introduction of several different methodologies for the incorporation of discrete variables in the context of PSO OPF. For example, Ref. [57] simply truncated discrete variables to transform them into integers, and [62,63] used binary encoding for discrete variables, and interpreted the trajectories of particles as probabilities, not as the values of the discrete variables themselves (which is the approach suggested by [52]). In both cases, the PSO handled the mixed-integer problem with remarkable accuracy, but generally, the second approach is more broadly accepted. The effectiveness of PSO in mixed-integer problems has been extensively verified (e.g., [53,76]).

Surprisingly, none of the reviewed works tackled the effect of a population initialization strategy on the performance of PSO OPF, as the topic has been studied in GA OPF. The literature suggests (e.g., [77]) that the choice of a good initial population can increase the performance of PSO. Therefore, this is a gap in the literature of PSO OPF that could be the focus of future research. Moreover, the discussion in the reviewed works on rules of thumb regarding swarm size, adaptive population size, or the number of iterations, is very limited. These are also gaps in the PSO OPF literature. The literature on PSO with adaptive population size gives promising results regarding the increase of the efficiency of PSO (e.g., [78]). Regarding the swarm size, the traditional rule of thumb suggests choosing a swarm size consisting of 20 to 50 particles. Recent research, however, suggests that this number is too conservative, and a swarm size of 70 to 500 particles should be preferred [79].

For practitioners looking to develop their own PSO OPF algorithms, we offer similar advice to that offered for practitioners looking to develop their own GA OPF algorithms. For applications with critical performance requirements, practitioners should look into the more "advanced" PSO features, such as dynamic penalty coefficients, congregation, or coordinated aggregation. For more trivial PSO OPF applications, simpler PSO OPF implementations can lead to satisfactory results and are easier to implement.

5. Cross-Comparison of GA OPF and PSO OPF

Cross-comparison amongst the different GA and PSO OPF implementations is a daunting task. Due to the volume of published works, the omission of information regarding some details of the exact implementation of each algorithm (see the N/A values in Tables 3 and 4), and the lack of standardized benchmark tests, the exact replication of the published works is, in some cases, impossible. Instead of trying to replicate the results, this work focuses on using the reported results of each work and comparing them with other works that have been applied in a similar setting. While a definitive benchmark for assessing the performance of a metaheuristic algorithm does not exist, the most popular test bed amongst the reviewed works is the IEEE 30-bus network. Indeed, 10 of the 12 reviewed GA OPF works and 14 of the 18 PSO OPF works were demonstrated on the IEEE 30-bus network.

In this work, we compile the reported results of the algorithms that were applied in that network and cluster them by the same objective function. The most popular objective function for both PSO and GA OPF is quadratic fuel cost minimization. Thus, we focused on GA and PSO OPF applications that were applied in the 30-bus IEEE network and used quadratic fuel cost minimization as their objective function, as this was the only combination of topology and objective function with an adequate number of works to allow for the extraction of useful conclusions regarding the accuracy of the proposed methods. Filtering the reviewed works for this objective function and this topology yielded five GA OPF works and eight PSO OPF works, which are presented in Tables 5 and 6, respectively.

Table 5. Reported results of the reviewed GA OPF works for the IEEE 30-bus network and cost minimization objective function.

Ref.	Number of Generations	Population Size	NFFEs	Crossover Probability	Mutation Probability	Cost [$]
[25]	200	80	16,000	0.9 (initial)	0.001/bit (initial)	802.06
[36]	50	400 (initial)	~17,000	dynamic	dynamic	799.84
[34]	200	50	10,000	0.9	1 (initial), 0.005 (final)	801.49
[40]	500	200	12,000	0.9	0.01	801.05
[17]	-	-	-	0.85	-	796.22

Table 6. Reported results of the reviewed PSO OPF works for the IEEE 30-bus network and cost minimization objective function.

<table>
<tr><th>Ref.</th><th>Number of Epochs</th><th>Population Size</th><th>NFFEs</th><th>Cognitive Coef.</th><th>Social Coef.</th><th>Inertia Weight</th><th>Cost [$]</th></tr>
<tr><td>[54]</td><td>500</td><td>50</td><td>25,000</td><td>2</td><td>2</td><td>-</td><td>800.41</td></tr>
<tr><td>[60]</td><td>500</td><td>50</td><td>25,000</td><td>0.5</td><td>0.5</td><td>0.9 (initial), 0.4 (final)</td><td>802.04</td></tr>
<tr><td>[56]</td><td>100</td><td>50</td><td>5000</td><td>2.05</td><td>2.05</td><td>-</td><td>4389 *</td></tr>
<tr><td>[63]</td><td>150</td><td>20</td><td>3000</td><td>2.1</td><td>2.1</td><td>1 (initial)</td><td>800.74</td></tr>
<tr><td>[66]</td><td rowspan="2">-</td><td rowspan="2">60</td><td rowspan="2">5000</td><td rowspan="2">2.05</td><td rowspan="2">2.05</td><td rowspan="2">0.9 (initial), 0.4 (final)</td><td>825.89 *</td></tr>
<tr><td>[67]</td><td>797.14</td></tr>
<tr><td>[68]</td><td>50</td><td>200</td><td>10,000</td><td>-</td><td>-</td><td>0.9 (initial), 0.4 (final)</td><td>800.4</td></tr>
<tr><td>[64]</td><td>20</td><td>200</td><td>4000</td><td>2</td><td>2</td><td>-</td><td>801.84</td></tr>
</table>

* Use modified versions of the IEEE 30-bus network.

The comparison of the efficiency of the various algorithms was not as straightforward. Most reviewed works focused on the elapsed CPU time when reporting the results. This, however, is not a representative performance metric, as the release dates of the reviewed works span almost three decades. So, due to the vast technological advancements that have occurred in this period, this metric could not be used. Thus, as a proxy metric for the computational efficiency of each of the reviewed works, the Number of Fitness Function Evaluations (NFFE) was used. As a PF was run in each FF evaluation, most of the execution time of a GA or PSO OPF was consumed in this step; therefore, the choice of this metric was justified. Additionally, the only other work [14] that attempted a similar cross-comparison of metaheuristics used the same metric.

The results of the cross-comparison of works that used the IEEE 30-bus network and QFCM objective function can be seen in Tables 5 and 6 for GA OFP and PSO OPF, respectively. It should be noted that [56,66] used modified versions of the IEEE 30-bus network, but were included in the cross-comparison for the sake of extracting conclusions regarding the efficiency and choice of PSO OPF hyper-parameters.

Ref. [17] yielded the best results overall, and [67] was the best-performing formulation amongst the PSO OPFs. Unfortunately, Ref. [17] omitted to report several key GA OPF parameters, such as the number of generations, population size, etc. The overall performance of both GA and PSO OPF was excellent, as the worst reported solution (802.06 $ of [25]) was only 0.733% worse than the best-reported solution. Therefore, the general accuracy of GA and PSO in the context of OPF was reaffirmed by this cross-analysis.

A large variation in the choice of hyper-parameters was observed in GA OPF. The number of generations ranged from 50 to 500, and the population size from 50 to 400. This was not in accordance with the guidelines presented in [48] which suggests that a larger population size should be preferred over a large number of generations. Regarding the choice of crossover and mutation probabilities, the general guidelines suggested by [33] were followed: the reported crossover probability was greater than 0.8 in all works, and the mutation probability stayed sufficiently low. Definitive conclusions cannot be drawn regarding the relative performance of GA OPFs that dynamically alter their crossover and mutation probabilities versus GA OPFs that use static values, as only [40] belonged in the latter category. The literature suggests that the performance of GA OPF with dynamic probabilities outperforms GA OPF with static probabilities [19].

The choice of hyper-parameters in PSO OPF was more consistent compared to GA OPF, a well-known advantage of PSO over the GA [9,14]. Rules of thumb about PSO suggest that the PSO swarm size should consist of 20 to 50 particles, and most reviewed works adhered to this rule of thumb. However, some works chose larger swarm sizes, and research suggests that this might lead to better performance [79]. Some variability, on the other hand, could be spotted in the number of epochs in the PSO OPF, which spanned from as low as 20 to as high as 500. It should be mentioned that works that used many epochs (e.g., [54]) usually included a stopping criterion to limit the NFFEs once a certain level of accuracy was obtained. Refs. [66,67] adopted a slightly different approach, and instead of defining the number of epochs, they defined the maximum number of NFFEs, which was set to 5000. The values of the cognitive and social coefficients were within the range suggested by rules of thumb (i.e., around 2), with the exception of [60], which claimed to have achieved better performance by setting the values to 0.5. Moreover, the choice of inertia weights was also relatively consistent, with most works using an initial inertia weight of 0.9, and a final of 0.4.

The comparison of the computational efficiency of GA and PSO OPF seemed to be slightly in favor of PSO OPF. PSO OPF generally required fewer NFFEs and achieved on-par (if not better) results in terms of accuracy, and, remarkably, several PSO OPF works required fewer than 5000 NFFEs to yield very accurate results. This observation was in accordance with the findings of [9,14]. The relationship between the objective function and NFFEs for the works included in Tables 5 and 6 is illustrated in Figure 3. As shown in Figure 3, a greater NFFE does not necessarily lead to more accuracy (especially in PSO OPF), suggesting that the proper implementation of a method (e.g., proper choice of hyper-parameters, etc.) cannot be offset by increasing its number of generations or epochs.

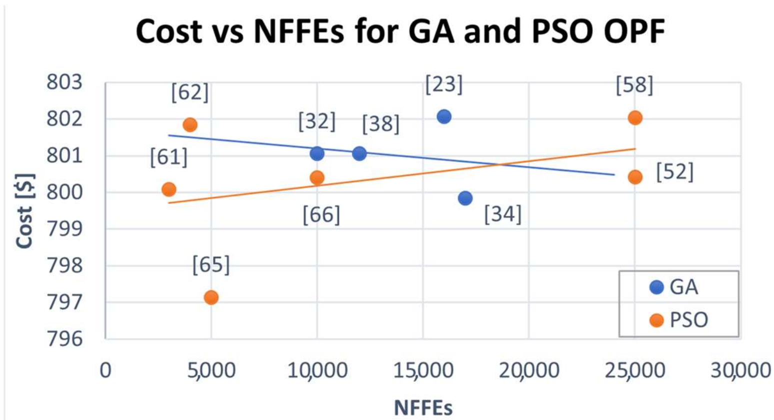

Figure 3. Cost vs. NFFES and trend lines of the GA OPF works in Table 5 (blue dots and line) and the PSO OPF works in Table 6 (orange dots and line).

Our findings can also be compared with the findings of [3,20]. These works directly compared specific GA and PSO OPF implementations, finding that GA OPF implementations were generally more accurate. This can also be argued from the findings of this work, but the gap is much smaller than the reported gap described in [3,20]. A plausible explanation for this is that these works were published in the mid-2000s, when research on PSO was still in its early stages and GA was more established (for example, [67], which yielded the best results amongst PSO OPFs, was published much later, in 2015).

6. Conclusions

This work focused on a literature review and cross-comparison of GA and PSO OPF implementations. In the first part of this work, the OPF problem and its sources of complexity were described. The deterministic optimization methods that can be used to solve the OPF were briefly discussed, and their shortcomings were highlighted. The shortcoming of deterministic optimization methods can be mitigated through the use of metaheuristic optimization methods. Two of the most popular metaheuristic optimization methods are GA and PSO.

Many works that use GA or PSO to solve the OPF problem exist in the literature. This work performed a thorough literature review of the most prominent of these works, presenting their contributions and main characteristics. To the best of our knowledge, this is the first time a thorough literature review that focuses exclusively on GA and PSO OPF has been carried out.

Both GA and PSO OPF can handle OPF for all the objective functions shown in Table 2. A significant advantage of GA and PSO OPF is that in order to handle a different objective function, the only necessary modification is the change of the FF. As already mentioned, these methods do not require any prior knowledge of the characteristics of the objective function or the search space, as they only calculate the FF and assign a quality value to each solution. Importantly, as shown in works that use GA or PSO OPF to solve the OPF problem for different objective functions (e.g., [36,39,42,54,66,67]), the hyper-parameters of these methods do not change as a result of a change in the FF. It should also be noted that GA and PSO OPF methods are able to handle both deterministic and stochastic (e.g., [45,65]) OPFs, which is crucial in the current landscape, where the uncertainty caused by stochastic generation should be considered.

The GA OPF works that were reviewed here were analyzed along seven axes: encoding, parent selection, crossover, adaptiveness, power flow formulation, objective functions, and constraint violation penalties, as can be seen in Table 3. The implications of choosing different values along these axes were discussed, and where possible, were corroborated with the best practices from other domains.

Similarly, the PSO OPF works that were reviewed were analyzed along four axes: adaptiveness of inertia weights, maximum velocity, objective functions, and constraint violation penalties, as can be seen in Table 4. Again, the implications of choosing different values along these axes were discussed, and where possible, were corroborated with the best practices from other domains.

Additionally, a cross-comparison between the reviewed GA and PSO OPF works was carried out. Since replication of some of the reviewed works is sometimes impossible (due to the lack of some implementation details), the reported results were used for the cross-comparison. Most of the reviewed works were demonstrated on the IEEE 30-bus network and used fuel cost minimization as their objective function. Thus, these works were clustered, and their reported results are presented in Tables 5 and 6 for GA OPF and PSO OPF, respectively. The goal of the cross-comparison was to compare GA and PSO OPF both in terms of accuracy and computational performance. To compare accuracy, the reported objective function values were used, and to compare computational performance, the NFFEs metric was used. The results of the cross-comparison suggest that GA OPF is slightly more accurate; however, PSO imposes much less computational burden. It should be noted, however, that the accuracy of all reported works was very satisfactory.

Finally, clustering the reported results also allowed for the comparison of the chosen hyper-parameters in GA and PSO OPFs. According to the reported results, it was observed that GA OPF approaches have more variability in the choice of hyper-parameters compared to PSO OPF approaches. Moreover, PSO OPFs adhere more to the rules of thumb that exist in the literature, potentially making their tuning process easier.

Our hope is that the clustering of the hyper-parameters of GA and PSO OPF can act as a first step towards the development of standardized GA and PSO OPF algorithms, which, in our opinion, is still the major barrier to the widespread adoption of GA or PSO OPF. Even though the emergence of competitive mixed-integer non-linear solvers has somewhat decreased the interest in metaheuristics, they can still offer many advantages. Apart from the benefits that are discussed in Section 1 (e.g., the accurate capture of non-convexities of the system, etc.), two more benefits are relevant. Firstly, GA or PSO OPF algorithms can be easily created for free by practitioners using open-source tools that are available online, while most mixed-integer non-linear solvers are commercial and come with a hefty cost. Secondly, as suggested by [80], deterministic optimization techniques require the full topology of the network to be available. With the proliferation of distributed generation and the need to run OPF analyses in the distribution network, where the observability of the network is more often than not limited, practitioners may need to rely on measurements to infer the topologies (e.g., use "black-box" neural network approaches). Metaheuristics are readily compatible with these approaches, while mathematical approaches will require special transformation to be compatible.

Therefore, since the performance of all reviewed works is adequate, further research should concentrate on the standardization of GA and PSO OPF methodologies, focusing on the following three directions:

1. The development of frameworks for the systematic tuning of the hyperparameters of GA and PSO OPF. This could entail providing generic rules for the tuning of the hyper-parameters depending on the parameters of each specific OPF (e.g., size of the problem, number of constraints, desired accuracy, etc.).
2. Future works should provide more transparent details regarding their implementation (e.g., reporting of the values of all hyper-parameters should be done explicitly). This would allow easier comparison of different methodologies and the replication of the work by other researchers.
3. The results reported in each work should also be standardized. Apart from the values of objective functions in each case, a useful metric that should be reported is the NFFEs. As discussed in this work, the NFFE metric allows the extraction of useful conclusions regarding the computational efficiency of each method.
4. Finally, standardization is also desired in the topologies used for benchmarking the proposed methods. The IEEE 30-bus network is a good benchmark for proof-of-concept experimentation. However, since practical networks are much larger, and as the scalability of GA and PSO OPF has not been extensively researched, larger benchmark networks should also be introduced and used. Additionally, future works should use distribution network topologies to benchmark the proposed methodology, as the application of GA or PSO OPF in the distribution network has not yet been adequately researched.

Author Contributions: Conceptualization, G.P. and P.B.; methodology, G.P. and P.B.; resources, G.P. and P.B.; writing—original draft preparation, G.P. and P.B.; writing—review and editing, G.P. and P.B.; visualization, G.P.; supervision, P.B.; project administration, P.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Hellenic Petroleum SA, via the Aristotle University of Thessaloniki Research Committee, Grant Number 624304.

Data Availability Statement: Not applicable.

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

References

1. Carpentier, J. Contribution to the economic dispatch problem. *Bull. De La Soc. Fr. Des Electr.* **1962**, *3*, 431–447.
2. Frank, S.; Steponavice, I.; Rebennack, S. Optimal power flow: A bibliographic survey I: Formulations and deterministic methods. *Energy Syst.* **2012**, *3*, 221–258. [CrossRef]
3. Biskas, P.N.; Ziogos, N.; Tellidou, A.; Zoumas, C.; Bakirtzis, A.; Petridis, V.; Tsakoumis, A.L. Comparison of Two Metaheuristics with Mathematical Programming Methods for the Solution of OPF. In Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems, Arlington, VA, USA, 6–10 November 2005; pp. 510–515. [CrossRef]
4. Tong, X.; Wu, F.F.; Zhang, Y.; Yan, Z.; Ni, Y. A semismooth Newton method for solving optimal power flow. *J. Ind. Manag. Optim.* **2007**, *3*, 553–567. [CrossRef]
5. Qiu, Z.; Deconinck, G.; Belmans, R. A literature survey of Optimal Power Flow problems in the electricity market context. In Proceedings of the 2009 IEEE/PES Power Systems Conference and Exposition, Seattle, WA, USA, 15–18 March 2009; pp. 1–6. [CrossRef]
6. Chatzigiannis, D.I.; Dourbois, G.; Biskas, P.; Bakirtzis, A.G. European day-ahead electricity market clearing model. *Electr. Power Syst. Res.* **2016**, *140*, 225–239. [CrossRef]
7. Papazoglou, G.K.; Forouli, A.; Bakirtzis, E.; Biskas, P.; Bakirtzis, A.G. Day-ahead local flexibility market for active and reactive power with linearized network constraints. *Electr. Power Syst. Res.* **2022**, *212*, 108317. [CrossRef]
8. Forouli, A.; Bakirtzis, E.; Papazoglou, G.; Oureilidis, K.; Gkountis, V.; Candido, L.; Ferrer, E.; Biskas, P. Assessment of Demand Side Flexibility in European Electricity Markets: A Country Level Review. *Energies* **2021**, *14*, 2324. [CrossRef]
9. Frank, S.; Steponavice, I.; Rebennack, S. Optimal power flow: A bibliographic survey II: Non-deterministic and hybrid methods. *Energy Syst.* **2012**, *3*, 259–289. [CrossRef]

10. Momoh, J.A.; Adapa, R.; El-Hawary, M.E. A review of selected optimal power flow literature to 1993. I. Nonlinear and quadratic programming approaches. *IEEE Trans. Power Syst.* **1999**, *14*, 96–104. [CrossRef]
11. Momoh, J.A.; Adapa, R.; El-Hawary, M.E. A review of selected optimal power flow literature to 1993. II. Newton, linear programming and interior point methods. *IEEE Trans. Power Syst.* **1999**, *14*, 105–111. [CrossRef]
12. Ebeed, M.; Kamel, S.; Jurado, F. Optimal Power Flow Using Recent Optimization Techniques. In *Classical and Recent Aspects of Power System Optimization*; Elsevier: Amsterdam, The Netherlands, 2018; pp. 157–183. [CrossRef]
13. Lin, M.-H.; Tsai, J.-F.; Yu, C.-S. A Review of Deterministic Optimization Methods in Engineering and Management. *Math. Probl. Eng.* **2012**, *2012*, 756023. [CrossRef]
14. Papadimitrakis, M.; Giamarelos, N.; Stogiannos, M.; Zois, E.; Livanos, N.-I.; Alexandridis, A. Metaheuristic search in smart grid: A review with emphasis on planning, scheduling and power flow optimization applications. *Renew. Sustain. Energy Rev.* **2021**, *145*, 111072. [CrossRef]
15. Banks, A.; Vincent, J.; Anyakoha, C. A review of particle swarm optimization. Part II: Hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. *Nat. Comput.* **2008**, *7*, 109–124. [CrossRef]
16. Katoch, S.; Chauhan, S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* **2021**, *80*, 8091–8126. [CrossRef] [PubMed]
17. Dashtdar, M.; Najafi, M.; Esmaeilbeig, M. Calculating the locational marginal price and solving optimal power flow problem based on congestion management using GA-GSF algorithm. *Electr. Eng.* **2020**, *102*, 1549–1566. [CrossRef]
18. Shabir, S. A Comparative Study of Genetic Algorithm and the Particle Swarm Optimization. *Int. J. Electr. Eng.* **2016**, *9*, 215–223.
19. Hassanat, A.; Almohammadi, K.; Alkafaween, E.; Abunawas, E.; Hammouri, A.; Prasath, V.B.S. Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach. *Information* **2019**, *10*, 390. [CrossRef]
20. Kumari, M.S.; Priyanka, G.; Sydulu, M. Comparison of Genetic Algorithms and Particle Swarm Optimization for Optimal Power Flow Including FACTS devices. In Proceedings of the 2007 IEEE Lausanne Power Tech, Lausanne, Switzerland, 1–5 July 2007; pp. 1105–1110. [CrossRef]
21. Wood, A.J.; Wollenberg, B.; Sheblé, G.B. *Power Generation, Operation, and Control*; John Wiley & Sons: New York, NY, USA, 2013.
22. Frank, S.; Rebennack, S. An introduction to optimal power flow: Theory, formulation, and examples. *IIE Trans.* **2016**, *48*, 1172–1197. [CrossRef]
23. Zheng, X.; Xu, Y.; LI, Z.; Chen, H. Co-optimisation and settlement of power-gas coupled system in day-ahead market under multiple uncertainties. *IET Renew. Power Gener.* **2021**, *15*, 1632–1647. [CrossRef]
24. Gomez, T.; Perez-Arriaga, I.; Lumbreras, J.; Parra, V.M. A security-constrained decomposition approach to optimal reactive power planning. *IEEE Trans. Power Syst.* **1991**, *6*, 1069–1076. [CrossRef]
25. Bakirtzis, A.G.; Biskas, P.; Zoumas, C.; Petridis, V. Optimal Power Flow by Enhanced Genetic Algorithm. *IEEE Trans. Power Syst.* **2002**, *17*, 8. [CrossRef]
26. Jiang, Q.; Han, Z. Solvability identification and feasibility restoring of divergent optimal power flow problems. *Sci. China Ser. E-Technol. Sci.* **2009**, *52*, 944–954. [CrossRef]
27. Papazoglou, G.K.; Bakirtzis, E.; Forouli, A.; Biskas, P.; Bakirtzis, A.G. A two-stage market-based TSO-DSO coordination framework. In Proceedings of the 2022 2nd International Conference on Energy Transition in the Mediterranean Area (SyNERGY MED), Thessaloniki, Greece, 17–19 October 2022; pp. 1–6. [CrossRef]
28. Yang, Z.; Zhong, H.; Xia, Q.; Bose, A.; Kang, C. Optimal power flow based on successive linear approximation of power flow equations. *IET Gener. Transm. Amp. Distrib.* **2016**, *10*, 3654–3662. [CrossRef]
29. Dommel, H.W.; Tinney, W.F. Optimal Power Flow Solutions. *IEEE Trans. Power Appar. Syst.* **1968**, *PAS-87*, 1866–1876. [CrossRef]
30. Burchett, R.C.; Happ, H.; Vierath, D.R. Quadratically Convergent Optimal Power Flow. *IEEE Trans. Power Appar. Syst.* **1984**, *PAS-103*, 3267–3275. [CrossRef]
31. Sun, D.I.; Ashley, B.; Brewer, B.; Hughes, A.; Tinney, W.F. Optimal Power Flow By Newton Approach. *IEEE Trans. Power Appar. Syst.* **1984**, *PAS–103*, 2864–2880. [CrossRef]
32. Holland, J.H. Genetic Algorithms-Computer programs that 'evolve' in ways that resemble natural selection can solve complex problems even their creators do not fully understand. *Sci. Am.* **2005**, *267*, 1992.
33. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*, 13th ed.; Addison-Wesley Professional: Boston, MA, USA, 1989.
34. Lai, L.L.; Ma, J.T. Improved genetic algorithms for optimal power flow under both normal and contigent operation states. *Electr. PowerEnergy Syst.* **1996**, *19*, 287–292. [CrossRef]
35. Osman, M.S.; Abo-Sinna, M.; Mousa, A.A. A solution to the optimal power flow using genetic algorithm. *Appl. Math. Comput.* **2004**, *155*, 391–405. [CrossRef]
36. Attia, A.-F.; Al-Turki, Y.; Abusorrah, A.M. Optimal Power Flow Using Adapted Genetic Algorithm with Adjusting Population Size. *Electr. Power Compon. Syst.* **2012**, *40*, 1285–1299. [CrossRef]
37. Devaraj, D.; Yegnanarayana, B. Genetic-algorithm-based optimal power flow for security enhancement. *IEE Proc. Gener. Transm. Distrib.* **2005**, *152*, 899. [CrossRef]
38. Paranjothi, S.R.; Anburaja, K. Optimal Power Flow Using Refined Genetic Algorithm. *Electr. Power Compon. Syst.* **2002**, *30*, 1055–1063. [CrossRef]

39. Kumari, M.S.; Maheswarapu, S. Enhanced Genetic Algorithm based computation technique for multi-objective Optimal Power Flow solution. *Int. J. Electr. Power Energy Syst.* **2010**, *32*, 736–742. [CrossRef]
40. Todorovski, M.; Rajicic, D. An Initialization Procedure in Solving Optimal Power Flow by Genetic Algorithm. *IEEE Trans. Power Syst.* **2006**, *21*, 480–487. [CrossRef]
41. Todorovski, M.; RajiCiC, D. A Power Flow Method Suitable for Solving OPF Problems Using Genetic Algorithms. In Proceedings of the IEEE Region 8 EUROCON 2003. Computer as a Tool, Ljubljana, Slovenia, 22–24 September; 2003; p. 1248186.
42. Szuvovivski, I.; Fernandes, T.; Aoki, A.R. Simultaneous allocation of capacitors and voltage regulators at distribution networks using Genetic Algorithms and Optimal Power Flow. *Int. J. Electr. Power Energy Syst.* **2012**, *40*, 62–69. [CrossRef]
43. Forouli, A.A.; Papazoglou, G.; Bakirtzis, E.; Biskas, P.; Bakirtzis, A.G. AC-feasible Local Flexibility Market with Continuous Trading. In Proceedings of the 11th Bulk Power Systems Dynamics and Control Symposium-IREP 2022, Banff, AB, Canada, 25–30 July 2022; p. 9. [CrossRef]
44. Biswas, P.; Pal, B. A fuzzy goal programming method to solve congestion management problem using genetic algorithm. *Decis. Mak. Appl. Manag. Eng.* **2019**, *2*, 36–53. [CrossRef]
45. Li, S.; Gong, W.; Wang, L.; Gu, Q. Multi-objective optimal power flow with stochastic wind and solar power. *Appl. Soft Comput.* **2022**, *114*, 108045. [CrossRef]
46. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Computat.* **2002**, *6*, 182–197. [CrossRef]
47. Maaranen, H.; Miettinen, K.; Penttinen, A. On initial populations of a genetic algorithm for continuous optimization problems. *J. Glob. Optim.* **2007**, *37*, 405–436. [CrossRef]
48. Vrajitoru, D. Large Population or Many Generations for Genetic Algorithms? Implications in Information Retrieval. In *Soft Computing in Information Retrieval*; Crestani, F., Pasi, G., Eds.; Physica-Verlag Heidelberg: Heidelber, Germany, 2000; Volume 50, pp. 199–222. [CrossRef]
49. Alander, J.T. On optimal population size of genetic algorithms. In Proceedings of the CompEuro 1992 Proceedings Computer Systems and Software Engineering, Hague, The Netherlands, 4–8 May 1992. [CrossRef]
50. Eiben, A.E.; Marchiori, E.; Valkó, V.A. Evolutionary Algorithms with On-the-Fly Population Size Adjustment. In *Parallel Problem Solving from Nature-PPSN VIII*; Yao, X., Burke, E., Lozano, J., Smith, J., Merelo-Guervós, J., Bullinaria, J., Rowe, J., Tiňo, P., Kabán, A., Schwefel, H.-P., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3242, pp. 41–50. [CrossRef]
51. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; p. 7. [CrossRef]
52. Kennedy, J.; Eberhart, R.C. A discrete binary version of the particle swarm algorithm. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; Volume 5, pp. 4104–4108. [CrossRef]
53. Laskari, E.C.; Parsopoulos, K.; Vrahatis, M.N. Particle swarm optimization for integer programming. In Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), Honolulu, HI, USA, 12–17 May 2002; Volume 2, pp. 1582–1587. [CrossRef]
54. Abido, M.A. Optimal power Flow using particle swarm optimization. *Int. J. Electr. Power Energy Syst.* **2002**, *24*, 563–571. [CrossRef]
55. Wang, C.-R.; Yuan, H.-J.; Huang, Z.-Q.; Zhang, J.-W.; Sun, C.-J. A modified particle swarm optimization algorithm and its application in optimal power flow problem. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005; Volume 5, pp. 2885–2889. [CrossRef]
56. Zhao, B.; Guo, C.; Cao, Y.J. Improved particle swam optimization algorithm for OPF problems. In Proceedings of the IEEE PES Power Systems Conference and Exposition, New York, NY, USA, 10–13 October 2004; pp. 933–938. [CrossRef]
57. Zhao, B.; Guo, C.; Cao, Y.J. An improved particle swarm optimization algorithm for optimal reactive power dispatch. In Proceedings of the IEEE Power Engineering Society General Meeting, San Francisco, CA, USA, 16 June 2005; pp. 403–410. [CrossRef]
58. Vlachogiannis, J.G.; Lee, K.Y. A Comparative Study on Particle Swarm Optimization for Optimal Steady-State Performance of Power Systems. *IEEE Trans. Power Syst.* **2006**, *21*, 1718–1728. [CrossRef]
59. Vlachogiannis, J.G.; Lee, K.Y. Coordinated aggregation particle swarm optimization applied in reactive power and voltage control. In Proceedings of the 2006 IEEE Power Engineering Society General Meeting, Montreal, QC, Canada, 18–22 June 2006; p. 6. [CrossRef]
60. He, S.; Wen, J.; Prempain, E.; Wu, Q.; Fitch, J.; Mann, S. An improved particle swarm optimization for optimal power flow. In Proceedings of the 2004 International Conference on Power System Technology, Singapore, 21–24 November 2004. [CrossRef]
61. Yumbla, P.E.O.; Ramirez, J.; Coello, C.A.C. Optimal Power Flow Subject to Security Constraints Solved With a Particle Swarm Optimizer. *IEEE Trans. Power Syst.* **2008**, *23*, 33–40. [CrossRef]
62. Gaing, Z.-L. Constrained optimal power flow by mixed-integer particle swarm optimization. In Proceedings of the IEEE Power Engineering Society General Meeting, San Francisco, CA, USA, 16 June 2005; pp. 290–297. [CrossRef]
63. Swarup, K.S. Swarm intelligence approach to the solution of optimal power flow. *J. Indian Inst. Sci.* **2006**, *86*, 439.
64. Umapathy, P.; Venkataseshaiah, C.; Arumugam, M.S. Particle Swarm Optimization with Various Inertia Weight Variants for Optimal Power Flow Solution. *Discret. Dyn. Nat. Soc.* **2010**, *2010*, 462145. [CrossRef]

65. Vu, P.; Le, D.; Vo, N.; Tlusty, J. A novel weight-improved particle swarm optimization algorithm for optimal power flow and economic load dispatch problems. In Proceedings of the IEEE PES T&D 2010, New Orleans, LA, USA, 19–22 April 2010; pp. 1–7. [CrossRef]
66. Singh, R.P.; Mukherjee, V.; Ghoshal, S.P. Particle swarm optimization with an aging leader and challengers algorithm for the solution of optimal power flow problem. *Appl. Soft Comput.* **2016**, *40*, 161–177. [CrossRef]
67. Singh, R.P.; Mukherjee, V.; Ghoshal, S.P. Particle swarm optimization with an aging leader and challengers algorithm for optimal power flow problem with FACTS devices. *Int. J. Electr. Power Energy Syst.* **2015**, *64*, 1185–1196. [CrossRef]
68. Naderi, E.; Pourakbari-Kasmaei, M.; Abdi, H. An efficient particle swarm optimization algorithm to solve optimal power flow problem integrated with FACTS devices. *Appl. Soft Comput.* **2019**, *80*, 243–262. [CrossRef]
69. Sarstedt, M.; Kluß, L.; Gerster, J.; Meldau, T.; Hofmann, L. Survey and Comparison of Optimization-Based Aggregation Methods for the Determination of the Flexibility Potentials at Vertical System Interconnections. *Energies* **2021**, *14*, 687. [CrossRef]
70. Zaro, F.R.; Abido, M.A. Multi-objective particle swarm optimization for optimal power flow in a deregulated environment of power systems. In Proceedings of the 2011 11th International Conference on Intelligent Systems Design and Applications, Cordoba, Spain, 22–24 November 2011; pp. 1122–1127. [CrossRef]
71. Niknam, T.; Narimani, M.; Aghaei, J.; Azizipanah-Abarghooee, R. Improved particle swarm optimisation for multi-objective optimal power flow considering the cost, loss, emission and voltage stability index. *IET Gener. Transm. Distrib.* **2012**, *6*, 515. [CrossRef]
72. Clerc, M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; pp. 1951–1957. [CrossRef]
73. Bonyadi, M.R.; Michalewicz, Z. Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review. *Evol. Comput.* **2017**, *25*, 1–54. [CrossRef]
74. Parrish, J.K.; Hamner, W.M. *Animal Groups in Three Dimensions: How Species Aggregate*, 1st ed.; Cambridge University Press: Cambridge, UK, 1997.
75. He, S.; Wu, Q.; Wen, J.; Saunders, J.; Paton, R.C. A particle swarm optimizer with passive congregation. *Biosystems* **2004**, *78*, 135–147. [CrossRef]
76. Rosendo, M.; Pozo, A. Applying a Discrete Particle Swarm Optimization Algorithm to Combinatorial Problems. In Proceedings of the 2010 11th Brazilian Symposium on Neural Networks, Sao Paulo, Brazil, 23–28 October 2010; pp. 235–240. [CrossRef]
77. ECampana, F.; Fasano, G.; Pinto, A. Dynamic analysis for the selection of parameters and initial population, in particle swarm optimization. *J. Glob. Optim.* **2010**, *48*, 347–397. [CrossRef]
78. Chen, D.; Zhao, C. Particle swarm optimization with adaptive population size and its application. *Appl. Soft Comput.* **2009**, *9*, 39–48. [CrossRef]
79. Piotrowski, A.P.; Napiorkowski, J.; Piotrowska, A.E. Population size in Particle Swarm Optimization. *Swarm Evol. Comput.* **2020**, *58*, 100718. [CrossRef]
80. Papazoglou, G.; Biskas, P. Review of Methodologies for the Assessment of Feasible Operating Regions at the TSO–DSO Interface. *Energies* **2022**, *15*, 5147. [CrossRef]